

Государственное бюджетное профессиональное образовательное учреждение
Республики Хакасия
«Техникум коммунального хозяйства и сервиса»

Комплект контрольно-оценочных средств

по ПМ.04 Разработка веб приложений на стороне клиента

для подготовки специалистов среднего звена/квалифицированных рабочих, служащих по
специальности/профессии

09.02.09 Веб-разработка

Абакан, 2025

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования, по профессии/специальности *09.02.09 Веб-разработка* и программы **ПМ.04 Разработка веб-приложений на стороне клиента**

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Формы контроля и оценки освоения учебной дисциплины по темам (разделам)
3. Контрольно-оценочные средства для проведения текущего контроля (контроль усвоения знаний и освоения умений)
4. Контрольно-оценочные средства для контроля по разделу (рубежный контроль)
5. Контрольно-оценочные средства для проведения промежуточной аттестации

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения контрольно-оценочных средств (далее – КОС)

Контрольно-оценочные средства предназначены для оценки освоения основного вида деятельности и уровня сформированности соответствующих ему общих и профессиональных компетенций в процессе текущего и рубежного контроля, промежуточной аттестации.

1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 06.	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 08.	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ВД 3	Разработка веб приложения на стороне клиента
ПК 3.1.	Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя.
ПК 3.2.	Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки.
ПК 3.3.	Создавать структуру кода веб-страницы информационных ресурсов в соответствии с дизайн-макетом.
ПК 3.4.	Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.

1.1.3. В результате освоения профессионального модуля обучающийся должен¹:

Владеть навыками	– разработки эскизов, схем, прототипов интерфейса пользователя информационного ресурса; – разработки дизайна компонентов интерфейса пользователя в соответствии со стандартами и требованиями заказчика; – разработки программного кода веб-страниц информационного ресурса, в том числе с использованием готовых технических решений; – разработки кроссбраузерной верстки веб-страниц информационного ресурса; – разработки клиентской части web-приложения в соответствии с техническим заданием (спецификацией).
Уметь	– применять программные средства для разработки интерфейса; – применять инструменты для оценки эффективности и удобства созданного интерфейса; – применять полученные данные для оптимизации интерфейса; – применять нормативные документы, определяющие требования к оформлению интерфейсов информационных ресурсов; – создавать адаптивный интерфейс web-ресурса; – применять специализированное программное обеспечение для

¹ Берутся сведения, указанные по данному виду деятельности в п. 4.2.

	верстки страниц информационных ресурсов; – использовать язык разметки страниц информационных ресурсов; – применять выбранные языки программирования для написания программного кода; – использовать выбранную среду программирования и средства системы управления базами данных; – использовать возможности имеющейся программной архитектуры информационного ресурса.
Знать	– современные принципы построения интерфейсов пользователя; – основные требования, предъявляемые к дизайну графических интерфейсов; – способы представления информации с учетом особенностей пользователя: возрастных, особенностей ограниченных возможностей здоровья и др.; – особенности отображения элементов интерфейсов web-ресурсов в различных браузерах; – правила реализации адаптивного интерфейса web-ресурса; – методы повышения читаемости программного кода; – синтаксис выбранного языка программирования, особенностей программирования на этом языке, стандартных библиотек языка программирования; – отраслевую нормативную техническую документацию; – особенности выбранной среды программирования; – компоненты программно-технических архитектур информационных ресурсов, существующих приложений и интерфейсов взаимодействия с ними; – сетевые протоколы и основы web-технологий; – современные стандарты взаимодействия компонентов распределенных приложений; – программные средства и платформы для разработки web-ресурсов; – основы информационной безопасности web-ресурсов.

2.

3.

4. ФОРМЫ КОНТРОЛЯ И ОЦЕНКИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ПО ТЕМАМ (РАЗДЕЛАМ)

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине (*название дисциплины*), направленные на формирование общих и профессиональных компетенций.

(технология оценки З и У по дисциплине прописывается в соответствии со спецификой дисциплины. Если экзамен проводится поэтапно или предусмотрена рейтинговая система оценки, то это подробно описывается)

Контроль и оценка освоения учебной дисциплины по темам (разделам)

Элемент учебной дисциплины	Формы и методы контроля					
	Текущий контроль		Рубежный контроль (контроль по разделу)		Промежуточная аттестация	
	Форма контроля	Проверяемые ОК; У, З	Форма контроля	Проверяемые ОК; У, З	Форма контроля	Проверяемые ОК; У, З
Раздел 1 (наименование)						
Тема 1.1 (название)	<i>Устный опрос №1 (или Практическая работа 1,...)</i>	<i>У1, 32, ОК3, ОК6.,ПК1.1, ПК1.2</i>	–	–	–	–
Тема 1.2	<i>Устный опрос №2 (или Тест №1, Лабораторная работа 1...)</i>	<i>У2, У3, 31, 33, ОК2, ОК3., ПК1.1, ПК1.2</i>	–	–	–	–

Тема 1.п	<i>Устный опрос №3 (или Самостоятельна я работа 1, Практическая работа 2....)</i>	<i>У., З., ОК...</i>	–	–	–	–
Контроль по разделу 1	–	–	<i>Контро льная работа №1</i>	<i>У1..., У2.... 3.1.. 32... ОК..., ОК..., ОК... ПК1.1, ПК1.2</i>	–	–
Раздел 2 (наименование)						
Тема 2.1	<i>Устный опрос (или Самостоятельна я работа 2 Практическая работа 3...)</i>	<i>У1, У2, 3 1, 32, 33, ОК 3, ОК 7 ПК1.1- ПК1.3</i>	–	–	–	–
Тема 2.2	<i>Самостоятельна я работа 2 (или...)</i>	<i>У., З., З., ОК., ОК..., ОК.. ПК1.1-ПК1.3</i>	–	–	–	–
Тема 2.п	<i>Форма контроля</i>	<i>У., З., З., ОК., ОК..., ПК...</i>	–	–	–	–
Контроль по разделу 2	–	–	<i>Контроль на я работа № 2</i>	<i>ОК..., ОК..., ОК... У., У2.... З... З...ПК</i>	–	–
Раздел N(наименование)						
Тема N.1	<i>Форма контроля</i>	<i>У., З., ОК.,ПК</i>	–	–	–	–
Тема N.2	<i>Форма контроля</i>	<i>У., З., ОК.,ПК</i>	–	–	–	–
Контроль по разделу N	–	–	<i>Форма контр оля</i>	<i>З,У, ОК, ПК</i>	–	–
Промежуточ на я аттестация	–	–	–	–	<i>Форма контр оля</i>	<i>У, З, ОК, ПК</i>

5. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ (КОНТРОЛЬ УСВОЕНИЯ ЗНАНИЙ И ОСВОЕНИЯ УМЕНИЙ)

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

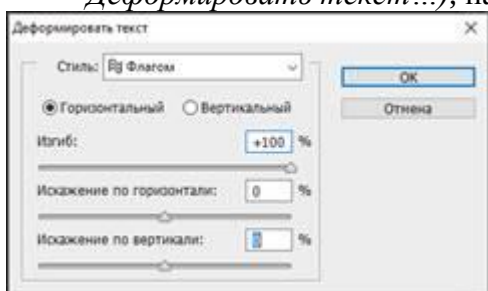
Раздел 1 Проектирование и дизайн интерфейсов

Лабораторная работа №1 Adobe Photoshop

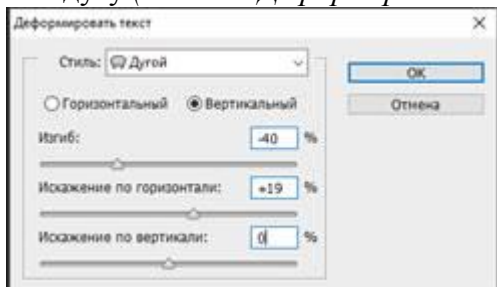
Тема: Текстовые эффекты

Задание 1. Изгиб и текстура

1. Создаем новую заготовку (например, 300*240, разрешением 150, RGB с белым фоном).
2. Пишем COMPUTER DESIGN горизонтальным текстом , искажаем как флаг (*Текст – Деформировать текст...*), настраиваем параметры.



3. Пишем LIKE I вертикальным текстом , искажаем как дугу (*Текст – Деформировать текст...*), настраиваем параметры.

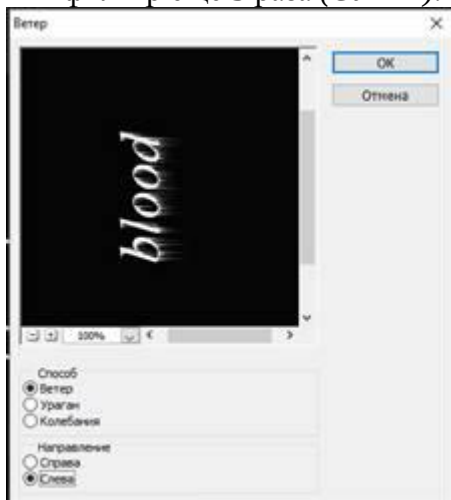


4. Объединяем оба текстовых слоя (но не задний план!) (Выделите слои, используя *Ctrl+щелчок по слою левой клавишей мыши*, затем нажмите комбинацию клавиш *Ctrl+E*).
5. Открываем fern.jpg. Переносим картинку в нашу заготовку (*Ctrl+A* (выделить всё), *Ctrl+C* (скопировать), *Ctrl+V* (вставить)). При необходимости растягиваем картинку, чтобы она закрывала все надписи.
6. Слой / Создать обтравочную маску.



Задание 2. Кровь

1. Создаем новую заготовку 400*400, разрешением 150, RGB, с черным фоном (Alt+Backspace).
2. Используя белый цвет, пишем «blood» шрифтом Times New Roman, Italic 25 пунктов по центру.
3. Растрируем текстовый слой (*Текст – Растрировать текстовый слой*).
4. Сводим слои (Ctrl+E).
5. Изображение / Вращение изображения / 90 против часовой
6. Фильтр / Стилизация / Ветер, Способ - Ветер, Направление - Слева. Повторить этот фильтр еще 3 раза (Ctrl+F).



7. Разворачиваем холст обратно (*Изображение / Вращение изображения / 90 по часовой*).
8. Фильтр / Галерея фильтров ... / Эскиз / Линогравюра. Тоновый баланс: 5-6, Смягчение: 1-3.
9. Изображение / Коррекция / Инверсия.
10. **Выделить текст.**
11. Выбрать основной цвет для заливки текста. Залить текст.
12. Фильтр / Галерея фильтров / Bas Relief. Detail: 12, Smoothness: 5.



Задание 3. Лед

1. Создаем новую заготовку 300*200, разрешением 150, черно-белую с белым фоном.
2. Используя черный цвет, пишем «ice» шрифтом Times New Roman, Bold Italic 25 пунктов по центру.
3. Растрируем текстовый слой.
4. Фильтр / Шум / Добавить шум, 70%, одноцветный.
5. Сводим слои.
6. Фильтр / Pixelate / Crystallize, Cell Size = 3
7. Изображение / Повернуть холст / 90 против часовой
8. Проинвертировать цвета: Изображение / Регулировки / Инвертировать
9. Для создания сосулек: Фильтр / Stylize / Wind, Method - Wind, Direction - From the left.
10. Разворачиваем холст обратно
11. Изображение / Режим / RGB.
12. Изображение / Регулировки / Тон/Насыщенность. Включить тонирование (190, 70, +3).



Задание 4. Пламя

1. Создаем новую заготовку (например, 300*100, разрешением 150, Черно-белое с белым фоном).
2. Пишем DESIGNER горизонтальным текстом.
3. Фильтр / Смазывание / Смазывание Гаусса 1 пкс.
4. Изображение / Повернуть холст / 90 против часовой.
5. Свести слои.
6. Проинвертировать цвета: Изображение / Регулировки / Инвертировать.
7. Для создания языков пламени: Фильтр / Stylize / Wind, Method - Wind, Direction - From the left.
8. Для большей реалистичности используем Фильтр / Distort / Ripple amount 50, size medium).
9. Разворачиваем холст обратно.
10. Изображение / Режим / Индексированные цвета.
11. Изображение / Режим / Палитра Black Body



Задание 5. Лава

1. Создайте новый документ 600x600 с черным фоном (Alt+Backspace). Напишите крупный белый текст, продублируйте текстовый слой (Слои – Создать дубликат слоя ...). Видимость копии отключите, объедините текстовый слой с фоном (Слои – Объединить слои, Ctrl+E).



2. Filter > Stylize > Wind (Method: Wind, Direction: From the left). Повторите несколько раз (CTRL+F).
Filter > Stylize > Wind (Method: Wind, Direction: From the right). Повторите несколько раз (CTRL+F).



3. Установите цвета по умолчанию (D). Затем Filter->Render->Difference Clouds. Включите видимость верхнего слоя и установите его прозрачность на 60%. Сведите слои.



4. Filter > Stylize > Extrude



5. Image > Adjustment > Invert



6. Filter > Sharpen > Unsharp Mask (Amount 500%, Radius: 2, Threshold: 0). Затем Filter >

Sharpen > Sharpen. Повторить пару раз.



7.	Image	>	Adjustments	>	Color	Balance:	
-	Midtones:	[	+100	]	[	-40	]
-	Shadows:	[	+70	]	[	-70	]
-	Highlights:	[	+80	]	[	0	]



Задание 8. Трёхмерный текст

1. Создаем новую заготовку (например, 400*300, разрешением 150, черно-белое).
2. Заливаем фон черным цветом.
3. Используя белый цвет, пишем «3D-Text» шрифтом Garamond, Bold, 15 пунктов.
4. Редактирование / Трансформация / Искривление. Тянем вниз левый нижний угол.
5. Создаем копию слоя.
6. Отключаем слой-копию, объединяем текстовый слой с фоном.
7. Фильтр / Смазывание / Смазывание движением. Искажение: -20, Дистанция: 20.
8. Фильтр / Стилизация / Вычисление краев.
9. Изображение / Регулировки / Инвертировать
10. Включаем копию слоя, задаем непрозрачность 70% и совмещаем слой с текстом с лицевой стороной трёхмерной надписи.
11. Создаем еще одну копию слоя, задаем непрозрачность 30% и совмещаем слой с текстом с задней стороной трёхмерной надписи.



Задание 9. Обведенные буквы

1. Создаем новую заготовку (например, 300*100, разрешением 150, RGB с белым фоном).
2. Пишем DESIGNER горизонтальным текстом.

3. Фильтр / Стилизация / Вычисление краев.
4. Создать копию слоя.
5. Размыть копию: Фильтр / Смазывание / Смазывание Гаусса 2 пкс.
6. Копию сдвинуть.

DESIGNER

Задание 10. Текст, заполненный линиями

1. Создайте новый документ 300x300. Жесткой кистью 3 px закрасьте линиями



2. Filter > Artistic > Cutout: Number of Levels - 7, Edge simplicity - 2, Edge Fidelity - 2
3. Сохраните изображение как .psd файл, он будет использован как карта смещения для другого файла
4. Создайте новый документ, напишите в нем крупный жирный текст
5. Filter > Distort > Displace, растрируйте текстовый слой



6. В качестве карты смещения выберите сохраненный .psd файл



Задание 11. Стекланный текст

1. Создайте документ произвольных размеров с белым фоном и черным напишите что-нибудь на свое усмотрение



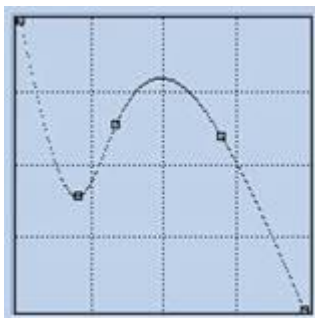
2. Примените к текстовому слою стиль слоя Drop Shadow со следующими настройками:



3. Примените к текстовому слою стиль слоя Bevel and Emboss со следующими настройками:



Здесь кликните по изображению контура (Gloss Contour) и вручную сделайте график примерно таким:



4. Примените к текстовому слою стиль слоя Color Overlay со следующими настройками:



Вот, что должно получиться:



Задание 7. Листок в линейку

1. Создайте новый документ размерами 400 на 400 пикселей. После этого создайте новый слой и выделите на нем прямоугольную область, залейте ее белым цветом. Затем, не снимая выделения, кликните в нем правой кнопкой мыши и выберите пункт Stroke.

Задайте настройки: Width: 1px, location: outside, opacity: 50%, color: #adc7e8

2. Примените к этому слою стиль Drop Shadow:



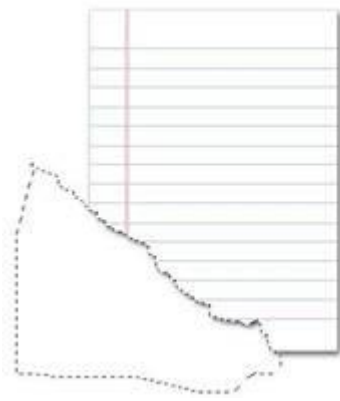
3. На новом слое инструментом Line Tool цветом #5a5858 делаем горизонтальные линии через равные расстояния. Для удобства отобразите сетку (Ctrl+'), настройте ее (Edit > Preferences > Guides, Grid & Slices) и включите привязку к сетке (View > Snap To > Grid)



4. Аналогично нарисуйте две вертикальные красные линии



5. Инструментом Lasso нарисуйте очертания фигуры, одна сторона которой и будет линией отрыва, и нажмите клавишу DELETE



6. Если хотите, можете залить слой Background любым цветом



7. Для большей реалистичности можно применить к слою с линиями фильтр Sketch > Water Paper



Задание 8. Папирус

1. Создадим новое изображение размером 200x200 пикселей с белым фоном. На новом слое с помощью инструмента Прямоугольное выделение (Rectangular marquee tool) нарисуем выделение произвольной формы и зальем его черным



2. Установим цвета по умолчанию (D). Снимем выделение и перейдем к команде Фильтр > Штрихи > Брызги (Filter > Brush Strokes > Spatter). Введем следующие значения: Радиус распыления (Spray radius): 10, Сглаживание (Smoothness): 5



3. Повторим тот же фильтр с параметрами 20, 15. Теперь избавимся от «отвалившихся» кусочков черного. Активируем инструмент Волшебная палочка (Magic wand tool), в опциях установим Отклонение (Tolerance) равным 5 и отметим Смежность (Contiguous). Теперь щелкнем в любом месте участка черного цвета и инвертируем выделение: Выделение > Инверсия (Select > Inverse) или Ctrl+Shift+I. Нажмем Del и инвертируем выделение обратно для перехода к черному



4. Зальем выделение на активном слое светло-желтым или бежевым цветом



5. Выберем в качестве основного коричневатый цвет. Возьмем мягкую кисть диаметром в 100 пикселей с непрозрачностью (Opacity) в размере 10%, активируем на панели параметров Аэрограф. Проведем кистью вдоль границ нашего папируса; легкий мазок в центре тоже не повредит. Снимите выделение



6. Для большей реалистичности обработайте ластиком края. Примените Layer Style > Drop Shadow...



Задание 3. Рамки. Фигурная деревянная рамка



1. Создать свою заготовку.
2. Сделать контур рамки: выделить прямоугольником; преобразовать выделение в контур; изменять его, добавляя и двигая точки.
3. Скопировать выделение на новый слой. Загрузить выделение.
4. Открыть палитру стилей. Загрузить набор Textures. Запустить Oak.
5. Слой / Эффекты слоя / Рельефность.
6. Вставить фото (face), для него тоже применить Рельефность.

Задание 4. Рамка из листьев

1. Создаем новое изображение с белым фоном. На новом слое создаем контур листика и заполняем его произвольным цветом, например, щелкнув правой кнопкой мыши по контуру и выбрав пункт Fill Path



2. Для того чтобы процесс наложения можно было контролировать и динамически изменять, применим соответствующий стиль к слою с лепестком. Перед этим определимся с цветами. К примеру, в панели инструментов установите цвет переднего плана как #77cc00, а фоновый цвет #3fa000. Теперь Layer > Layer style > Gradient Overlay и установите параметры, как на скриншоте. Градиент выбран от основного к фоновому



3. Щелкните правой кнопкой мыши в районе контура и выберите пункт Make Selection для того, чтобы избавиться от контура и выделить созданную им область. Теперь выполните

команду Edit > Define Brush Preset и введите имя создаваемой нами кисти

4. Создаем новый слой. Выбираем из панели инструментов Brush, в панели параметров созданную нами кисть и жмем F5 для открытия вкладки Brushes. Изменим параметры кисти. Для начала поставим флажок возле строки Shape Dynamics и изменим здесь Size Jitter, установив значение 100% и Angle Jitter, установив 70%. Далее ставим флажок Scattering, устанавливаем Scatter 261%, Count 1, Count Jitter 86
5. Теперь можно применять созданную и настроенную кисть для рисования рамки произвольными цветами. Для рамки, показанной ниже, использовался цвет переднего плана #a83e0a и фоновый цвет #c99565

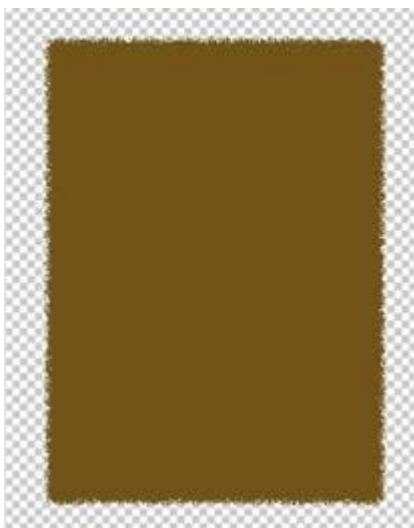


Задание 5. Двойная рамка

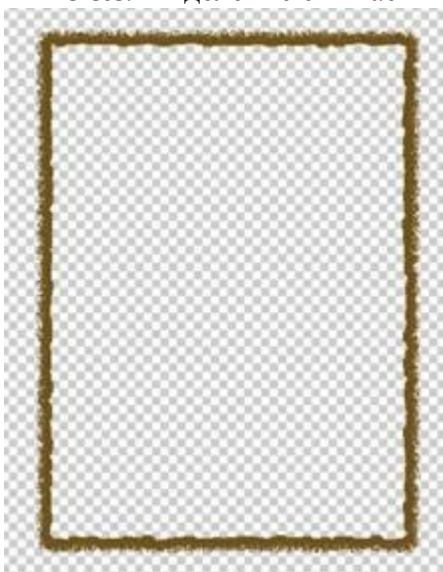
1. Создаем новое изображение 600 x 600 с прозрачным фоном. Устанавливаем цвет переднего плана #725418, заднего плана - белый
2. На новом слое выделяем прямоугольник и заливаем цветом переднего плана. Выделение снимаем



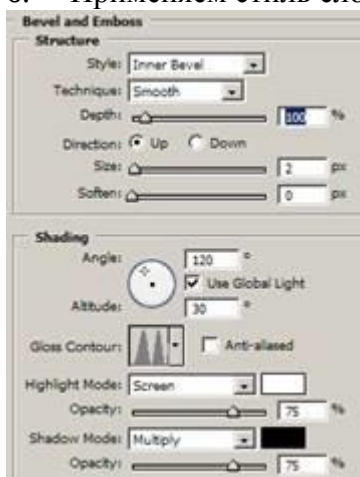
3. Filter > Sketch > Torn Edges (50, 1, 17)
4. Для удаления белой бахромы по краям рамки щелкаем волшебной палочкой с допуском 100 внутри коричневого прямоугольника, инвертируем выделение и нажимаем Delete. Выделение снимаем



5. Выделяем содержимое слоя. Select > Modify > Contract примерно на 8 - 10 px. Нажимаем Delete. Выделение снимаем



6. Применяем стиль слоя



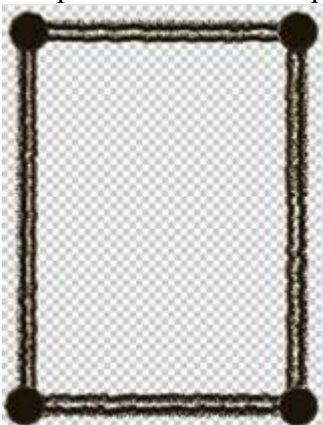
7. Дублируем слой и применяем свободную трансформацию (Ctrl+T), уменьшая рамку на втором слое. Подгоняем её размер, чтобы она как бы вошла внутрь первой



8. Создаем 4 коричневых круга на углах рамки. В конечном итоге все кружки должны оказаться в одном слое

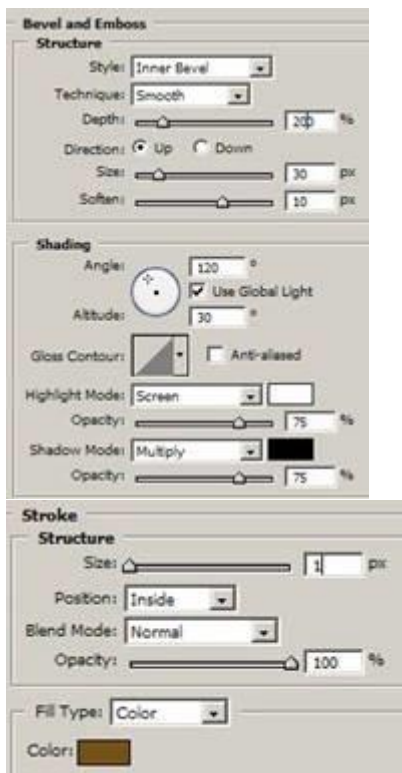


9. Применяем к слою с кружками настройки стиля BEVEL AND EMBOSS такие же, как мы применяли к самой рамке

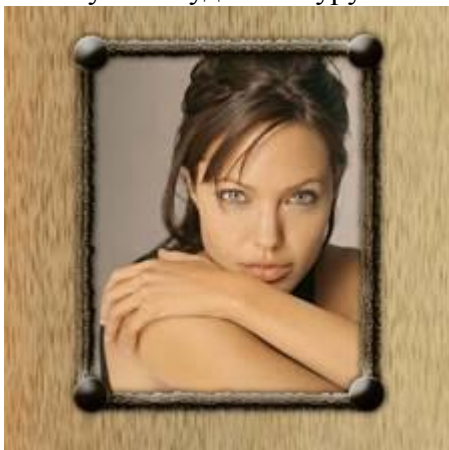


10. Объединяем слой с кружками и рамками и применяем стили слоя





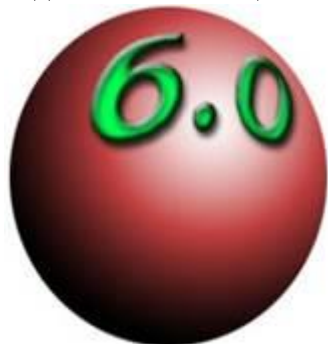
11. Берем подходящую фотографию и вставляем ее в рамку. Нижним слоем используем какую-нибудь текстуру



Лабораторная работа №5

Тема: Имитация объема

Задание 1. Имитация объема. Сфера



1. Заготовка.
2. Цветной текст 6.0, отрастрировать.
3. Фильтр Render / 3D Transform, сфера вокруг надписи, исказить.
4. Создать новый слой, перенести его под слой с надписью. Создать эллиптическое выделение, совпадающее с областью, образованной фильтром.
5. Ластиком стереть лишние серые участки сферы на слое с надписью.
6. Залить градиентом сферу.

7. Для надписи – эффекты слоя Падающая тень, Внутренняя тень, Рельефность.
8. При необходимости трансформировать надпись.

Задание 2. Имитация объема. Болт



1. Заготовка.
2. Создать круглое выделение, залить градиентом.
3. С помощью контуров создать полукруглые выделения для левой и правой части болта, скопировать их на отдельные слои.
4. К новым слоям применить эффекты Внутренняя тень, Атлас.
5. Свести слои.
6. Яркость / Контраст.
7. Фильтр Пикселизация / Мозаика.

Задание 3. Имитация объема. Пуговица



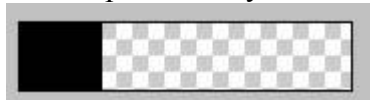
1. Заготовка.
2. Создать круглое выделение для всей пуговицы, залить градиентом от центра: белый – черный – белый – черный.
3. Залить градиентом центр пуговицы: белый – черный от левого верхнего угла.
4. Фильтр Texture / Texturizer.
5. Применить к слою эффект Рельефности с текстурой.
6. Выделить две дырочки, залить белым, скопировать на новый слой.
7. Применить к слою эффекты Внутренние тени, Внешний отблеск.
8. Свести слои, раскрасить.

Задание 4. Сферическая 3D-клетка

1. Создайте новый документ со следующими настройками:

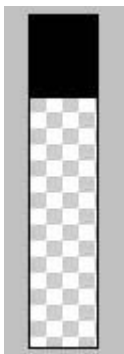


2. Максимально увеличьте масштаб. Черным квадратным карандашом размером 5 закрасьте левую часть



Edit > Define Pattern. Назовите созданный узор «Горизонталь»

3. Image > Rotate Canvas > 90CW.

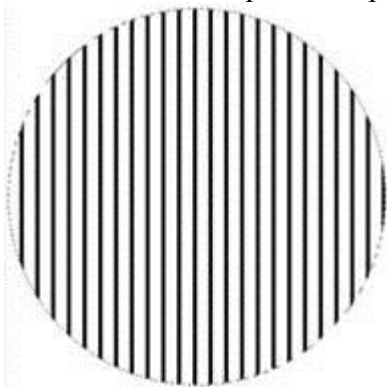


Edit > Define Pattern. Назовите созданный узор «Вертикаль». Теперь можно закрыть документ, не сохраняя

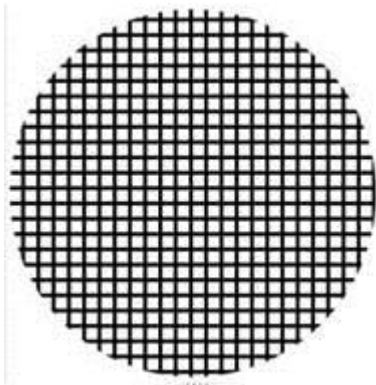
4. Создайте новый документ 500x500 с белым фоном. На новом слое сделайте круглую выделенную область



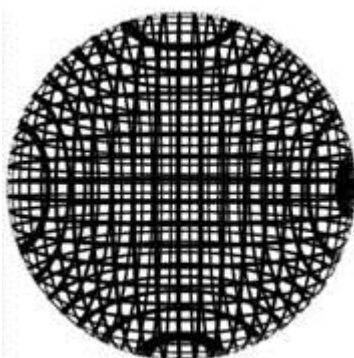
5. Edit > Fill. В разделе «Use» выбираем «Pattern», в разделе «Custom Pattern» выбираем «Горизонталь». Теперь выделение заполнено вертикальными линиями



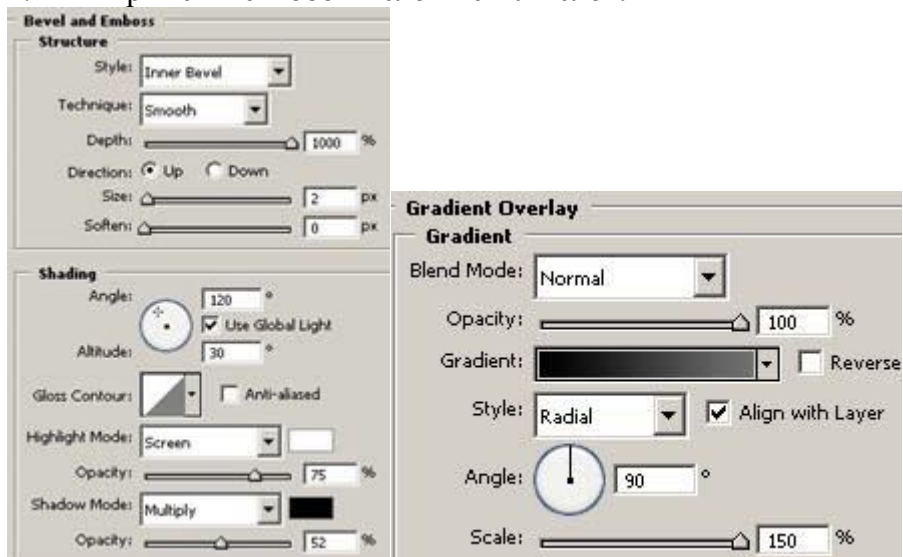
Аналогично заполните выделенную область горизонтальными линиями



6. Создайте дубликат слоя. К верхнему слою примените Filter > Distort > Spherize с параметрами Amount: 100%, Mode: Normal, к нижнему - с параметрами Amount: -100%, Mode: Normal. Снимите выделение



7. Примените к обоим слоям стили слоя:



Уменьшите Opacity второго слоя до 30%



8. Теперь можно разместить кого-нибудь между этими двумя слоями, подобрать фон и т.д.



Задание 5. Кубическое яблоко

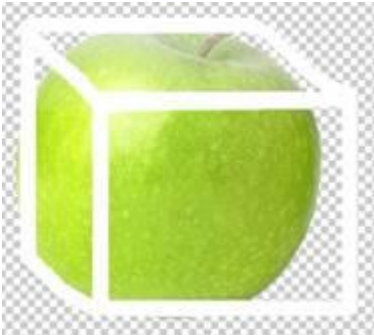
1. Откройте изображение



2. Удалите серый фон и увеличьте размер холста до 800 x 800 px



3. На новом слое белой кистью 25 px нарисуйте каркас



4. Перейдите на слой с яблоком. Edit > Transform > Warp. Растягивайте сетку так, чтобы яблоко вписалось в каркас

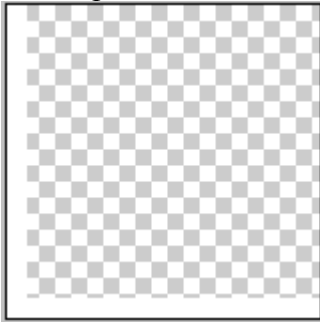


5. Уменьшите прозрачность слоя с каркасом до 40%, размойте его примерно на 25 px: Filter > Blur > Box Blur



Задание 6. Глобус

1. Создайте новый документ 60 x 60 px с прозрачным фоном. Максимально увеличьте масштаб. Белым квадратным карандашом размером 4 px закрасьте вертикальную линию слева и горизонтальную линию внизу. Сохраните созданный узор: Edit > Define Pattern



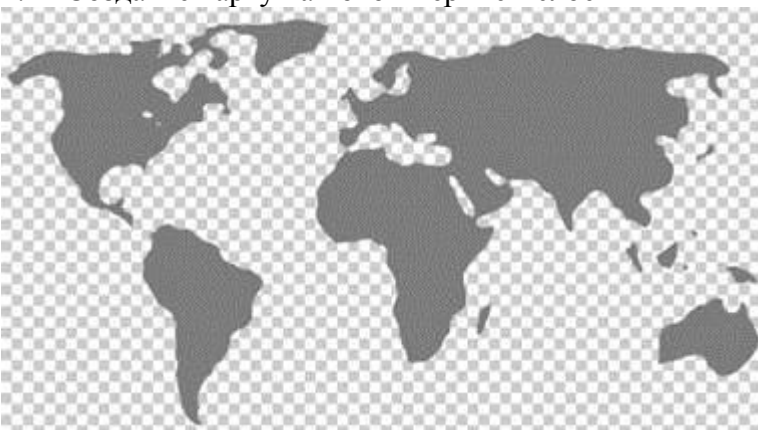
2. Создайте новый документ 500x500 с белым фоном. На новом слое выделите окружность и залейте ее градиентом



3. Не снимая выделения, создайте новый слой «Grid» и залейте его созданным узором: Edit > Fill. Use: Pattern...



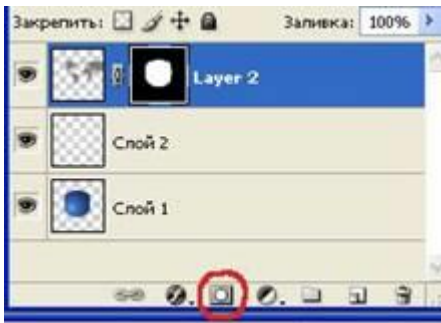
4. Создайте карту на новом верхнем слое



и наложите ее на будущий глобус



5. При выделенном слое с окружностью, перейдите на слой с картой и добавьте слой-маску



6. При выделенном слое с окружностью, примените Filter > Distort > Spherize - Amount: 100%, mode: Normal последовательно для всех трех слоев

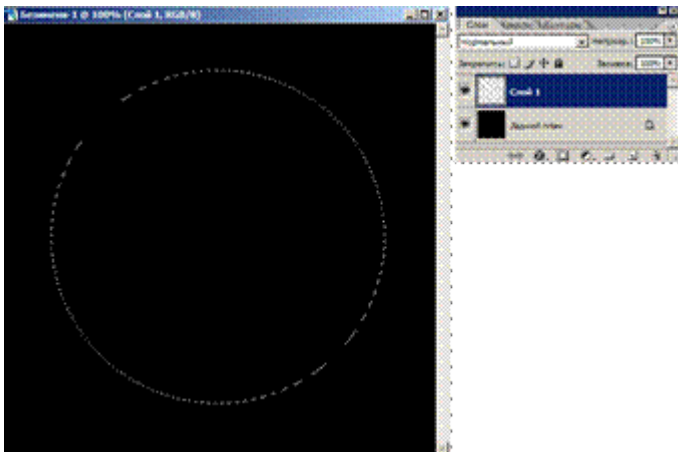


7. Для слоя «Grid» примените режим наложения слоев «Overlay», а для слоя с картой – «Color Dodge»



Задание 7. Планета

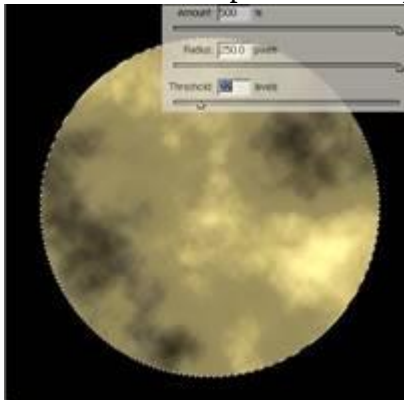
1. Создайте новый документ 500x500 с черным фоном. На новом слое выделите окружность



2. Установите цвет переднего плана #BEB074, заднего плана #736942. На новом слое выполните Filter > Render > Clouds



3. Filter > Sharpen > Unsharp Mask

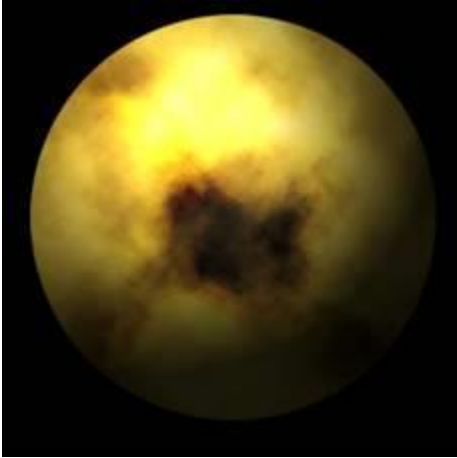


4. Продублируйте текущий слой (CTRL+J). Edit > Transform > Rotate 90CW. Установите режим слоя "Overlay".
Еще раз продублируйте слой и выполните те же действия



5. Сведите три верхних слоя с планетой. Добавьте эффект освещения Filter > Render > 3D

Lighting Effects и настройте его



6. Filter > Distort > Spherize, amount 100%. Потом тот же фильтр, но amount 50%



7. Теперь нарисуем звездное небо. Создайте новый слой и залейте его черным. Затем Filter > Noise > Add Noise. Потом скопируйте этот слой и установите режим слоя «Overlay»



8. Поэкспериментируйте со световыми эффектами, цветами и стилями слоя для планеты



Задание 8. Трехмерная книга

1. Откройте изображение, используемое для обложки



2. Добавьте фоновый слой, по размеру больший оригинала, чтобы в дальнейшем было удобно работать при создании трехмерного объекта. Добавьте нужный текст и картинки



3. Свяжите все слои и трансформируйте в режиме наклона



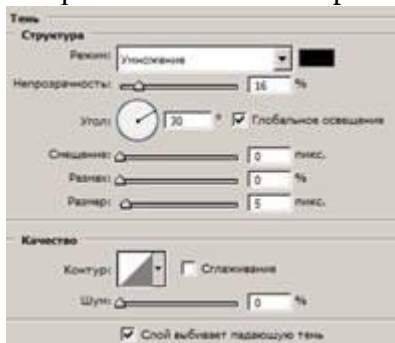
4. На новом слое создайте выделение для внешней части обложки и залейте черным. Также на новом слое создайте прямоугольное выделение вдоль обложки, трансформируйте его и залейте. Можно создать надпись небольшого размера. Заполните фоновый слой подходящим градиентом



5. Перейдем к созданию страниц для книги. Создайте выделение на новом слое и заполните его белым



6. Чтобы будущие страницы не сливались в единую белую массу, к слою с белой заливкой примените стиль Drop Shadow (Тень)



Добавьте страницы, копируя слой. Необязательно размещать эти страницы максимально ровно, должны быть небольшие искажения



7. Объедините слои со страницами, сотрите ненужные части, подкорректируйте

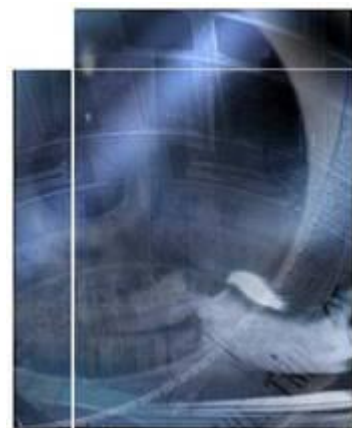


Задание 9. Трехмерная коробка

1. Откройте изображение, используемое для коробки



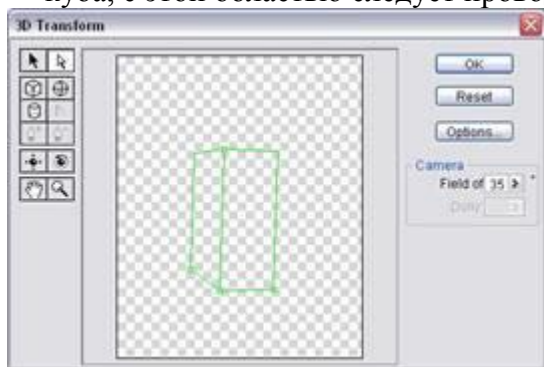
2. Прямоугольником выделите левую часть изображения, вырежьте на новый слой и сместите немного влево. Прodelайте аналогичную операцию с верхней частью изображения. Удалите лишнюю часть угла



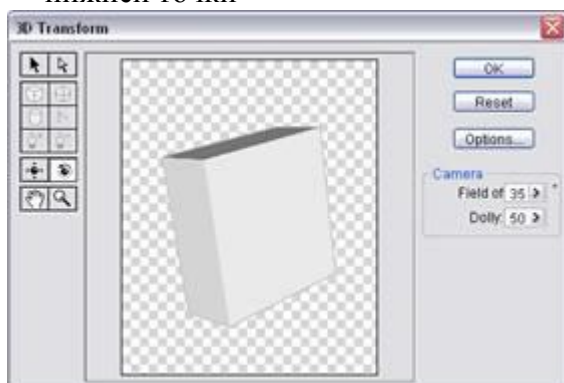
3. Добавьте нужный текст и картинки



4. Теперь с помощью фильтра Render > 3D-Transform создадим куб, на который потом наклеим нашу обложку. Выберите инструмент Cube Tool (Куб) и нарисуйте куб, затем выберите инструмент Selection Tool (Стрелка), с его помощью «схватитесь» и потяните за левую нижнюю точку куба, с этой областью следует проводить дальнейшие манипуляции



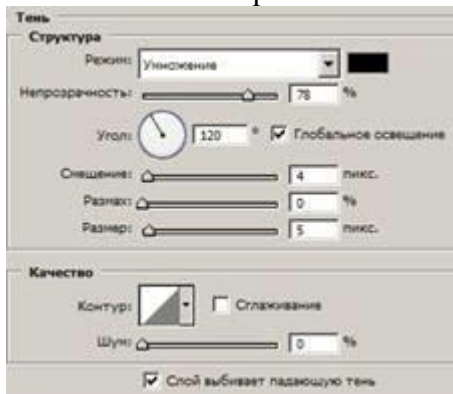
5. Инструментом Trackball Tool (Ориентация) разверните куб таким образом, чтобы каждая из трех сторон имела свой оттенок серого. Если какая-то часть окажется прозрачной, вернитесь к использованию Selection Tool и измените положение левой нижней точки



6. Теперь переместите все слои с обложками на изображение с кубом и с помощью трансформации наложите их поверх

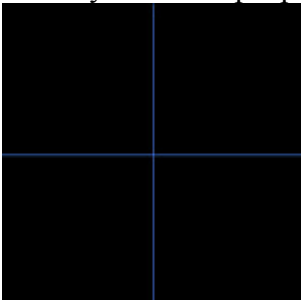


7. Для завершения композиции, выделите половину верхней части коробки, вырежьте на новый слой и примените стиль Drop Shadow (Тень)



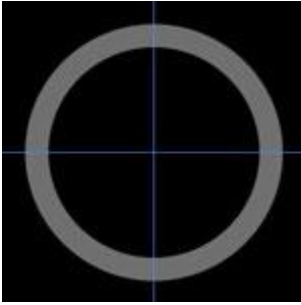
Задание 10. Имитация объема. Металлическое кольцо

1. Создайте новое изображение RGB. Заполните фон черным цветом. Теперь возьмите направляющие сверху и слева и установите их в центре изображения. У вас должно получиться перекрестие как на рисунке. Создайте новый слой.



2. Возьмите инструмент «выделение круга» и установите перекрестие в точке пересечения направляющих линий. Удерживая ALT+SHIFT нажмите и создайте круг. Залейте круг

серым. Повторите этот пункт, но сделайте второй круг немного меньшего диаметра. Теперь нажмите DEL для вырезания меньшего диаметра. Отмените выделение. У вас должно получиться серое кольцо.



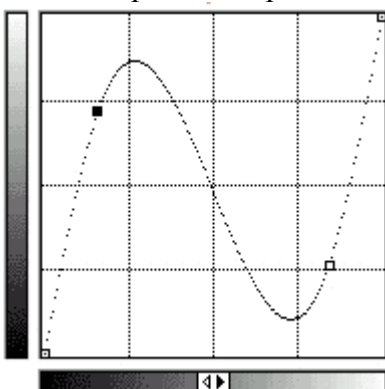
3. Выделите кольцо и создайте новый канал на основе выделения. Залейте выбранную область белым цветом и отмените его выбор. В канале производим Смазывание Гаусса: первым проходом используем фильтр 4, затем 3, а потом 2.



4. На слое с кольцом Фильтр > Render > Lighting effects. В "texture channel" выберите новый канал. Подобрать параметры.



5. Разглаживаем края кольца. Выделяем его, сглаживаем границы на 1 пкс, инвертируем выбор, нажимаем DEL.
6. Настраиваем кривые.



7. Получаем:



Задание 11. Имитация объема. Металлический провод

1. Создаем новое изображение, 250x250 пикселей, RGB. В качестве цвета основного фона выбираем темно-серый. Теперь создаем новый Слой и с помощью кисти рисуем линию требуемой формы шириной 15 пикселей. Назовем слой Wire.



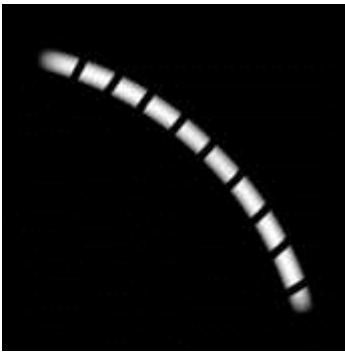
2. Сделаем копию слоя Wire путем его перетаскивания на пиктограмму кнопки Новый Слой. Переименуем новый слой в Cut Wire.
3. Теперь спрячем слой Wire, щелкнув на пиктограмме глаза, на панели Слои. Этот слой нам пригодится немного позже.
4. Сейчас выберем инструмент Ластик, определим размер кисти в пределах 5 пикселей и с его помощью придадим кривой вид, показанный на рисунке.



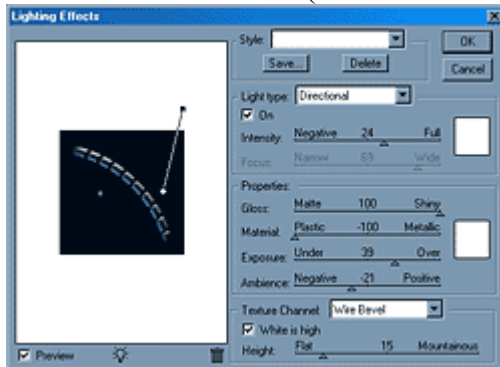
5. Удерживая Ctrl, кликаем по невидимому слою Wire в панели Слои. Таким образом создадим выделение, не принимая во внимание прозрачность слоя Wire.
6. Перейдем затем к панели Каналы и там сохраним созданное выделение. Переходим на этот канал.



7. Применим к каналу фильтр Смазывание Гаусса с опцией Радиус сначала 9 пикселей, затем 6, затем 3 пикселя, и, наконец, 1 пиксел.
8. Применим теперь к выделению АвтоУровни.
9. Вернемся на панель Слои (Layers), и, удерживая Ctrl кликнем на слое Cut Wire, чтобы выбрать его прозрачность. Затем опять переходим к панели Каналы, кликаем на новом канале, инвертируем выделение, и в конце концов очистим выделение, нажав Del. Уберем выделение.



10. Вернемся к панели Слои и сделаем активным слой Cut Wire.
11. Откроем диалоговое окно Lightning Effects (Фильтр > Render > Lighting Effects)... Создадим белый направленный свет (Directional light) сверху вниз - из правого верхнего угла, - как показано на рисунке. Затем создадим направленный свет снизу вверх - из левого нижнего (источник света копируется с Alt).



12. Определим

Глянец	(Gloss):	100	Яркий	параметры:
Материал	(Material):	-100	Пластик	(Shiny)
Экспозиция		(Exposure):		(Plastic)
Среда		(Ambience):		39
				-21

 Установим Текстурный Канал (Texture Channel) в позицию Альфа1 и определим его высоту в размере 15. Теперь установим направление освещения таким образом, чтобы достичь наиболее желаемого эффекта.
13. Подкрашиваем с помощью Цветового баланса или Тона/Насыщенности.



14. Сделаем видимым слой Wire. Еще раз кликнем на нем, удерживая Ctrl, чтобы сделать выделение.
15. Чтобы сделать линию на 1 пиксель тоньше, инвертируем выделение, затем сжимаем выделение на 1 пиксель, очищаем выделение – Del, убираем выделение.
16. И последний шаг: Фильтр > Render > Lighting Effects... Установим Текстурный Канал (Texture Channel) на Wire Прозрачность.
17. Добавляем по усмотрению эффект Тени.

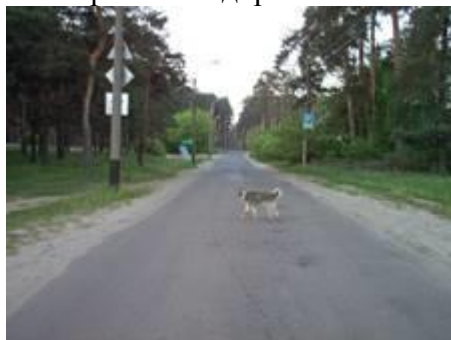


Лабораторная работа №6

Тема: Обработка фотографий

Задание 1. Клонирование в перспективе

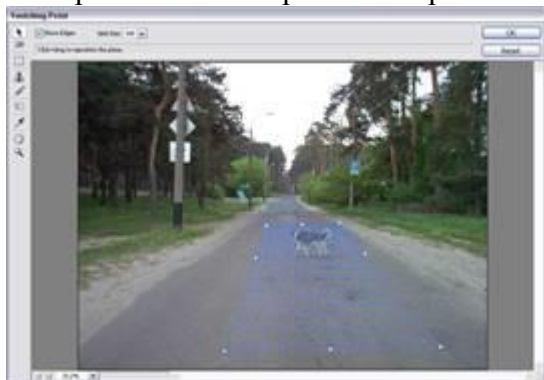
1. Открываем фотографию. Наша задача - удалить собаку таким образом, чтобы сохранилась дорога в ее истинном виде



2. Создайте новый слой.

Filter > Vanishing Point (Alt+Ctrl+V). Прежде всего нужно задать угол перспективы с помощью инструмента Create Plane tool. Щелкните там, где хотите начать и продолжайте щелчки под нужными вам углами, аналогично использованию инструмента Polygonal Lasso.

Если перспектива задана правильно, появится сетка голубых оттенков, в противном случае сетка будет желтого цвета. По завершении расстановки точек, может оказаться так, что вы не угадали изначально с шириной. Инструментом Edit Plane Tool можно редактировать получившуюся сетку. Если сетка поменяет свой цвет на красный, значит перспектива построена неверно



3. Теперь выберите Stamp Tool. Сетка в центре сразу пропадет. Как и при работе с Clone Stamp, сначала нужно задать образец, т.е. источник, который будет использоваться при клонировании. Для этого щелкните в нужном месте, удерживая клавишу Alt. Потом зарисовывайте удаляемый нами объект. При клонировании будет сохраняться эффект перспективы. Для реалистичности лучше почаще задавать новые образцы. По завершении нажмите кнопку Ok. Отклонированная часть будет располагаться на отдельном слое



Задание 2. Колоризация ч/б фото

1. Откроем нужную фотографию



2. Перейдите в цветовой режим RGB. Создайте новый слой, залейте его белым. Режим наложения слоев измените на Multiply (Умножение)



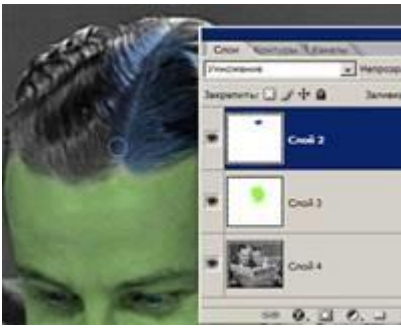
3. Выберите зеленый цвет и мягкой кистью закрасьте кожу



4. Ластиком сотрите краску с губ, глаз и т.д.



5. Создайте новый слой, установите режим наложения слоев Multiply и закрасьте волосы



6. Добавьте цвета левому мужчине

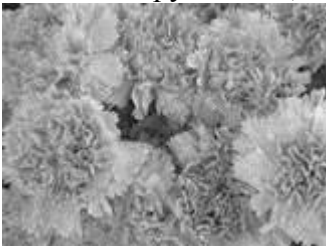


Задание 3. Повышение резкости

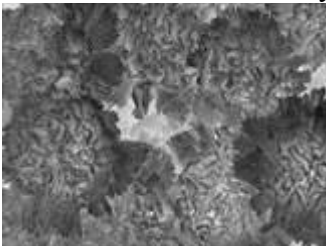
1. Откроем нужную фотографию



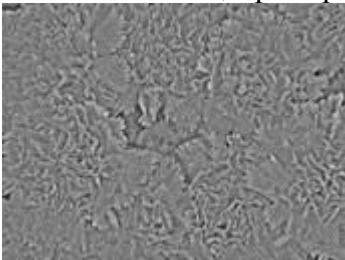
2. Скопируем слой, обесцветим его (Ctrl+Shift+U)



3. Создадим еще одну копию слоя и проинвертируем на нем цвета (Ctrl+I)



4. К верхнему слою примените режим смешивания Vivid Light, потом Filter > Blur > Gaussian Blur, примерно на 3



5. Объедините два верхних слоя, примените режим смешивания Vivid Light



Задание 4. Смягчение фокуса

1. Откроем нужную фотографию



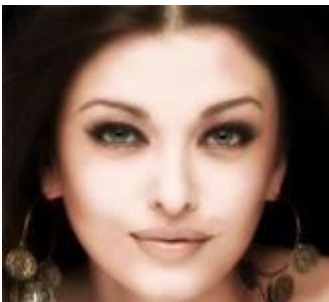
2. Создадим копию слоя и обесцветим его (Ctrl+Shift+U)



3. Image > Adjustment > Posterize, примерно на 7.
Примените режим смешивания Overlay



4. Filter > Blur > Gaussian Blur, примерно на 5



Задание 7. Винтажный снимок

1. Откройте нужную фотографию



2. Откройте палитру слоев, два раза щелкните по слою Background, после чего в появившемся диалоговом окне нажмите ОК. Теперь Background превратился в обычный слой
3. Добавьте немного шума: Filter > Noise > Add noise (Amount: 6%, Distribution: Gaussian, Monochromatic)
4. Наступило время убрать цвета с фото и добавить оттенок сепии. Внизу палитры слоев нажмите кнопку Add Adjustment Layers



и выберите Hue/Saturation (Hue: 23, Saturation: 13, Lightness: 0, Colorize отмечено)



5. Наше фото слишком чистое и неповреждённое. Давайте сделаем его более грубым. Создайте новый слой (Ctrl+Shift+N), прямоугольным выделением (M) выделите верхнюю половину изображения и залейте цветом #A99A90. Теперь Filter > Texture > Grain (Intensity: 40, Contrast: 50, Grain Type: Vertical). Далее Filter > Brush Strokes > Spatter (Spray Radius: 10, Smoothness: 5)



Теперь трансформируйте новый слой так, чтобы он закрыл картинку полностью. Перейдите в палитру слоев, установите режим смешивания Soft Light и прозрачность примерно 60%

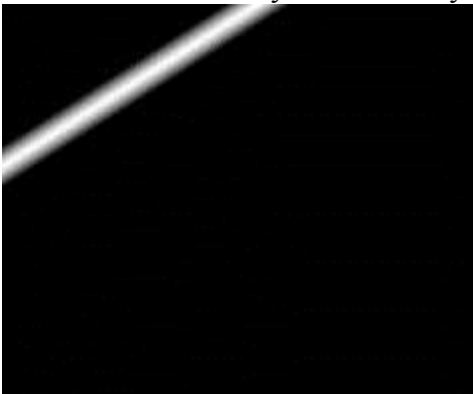


6. Теперь добавим эффект перегиба.

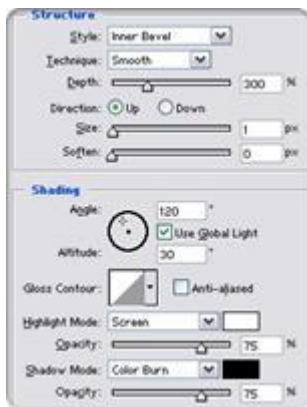
Переключитесь в палитру каналов и создайте новый канал. Выберите Gradient tool (G). Переместитесь в панель опций и установите градиент Foreground to background, Reflective, Lighten mode.



Сделайте полосу в одном из углов.



Ctrl+левая кнопка мыши по каналу Alpha1, после чего выберите RGB канал. Идём в Edit > Copy Merged (Shift+Ctrl+C). Потом - Edit > Paste (Ctrl+V). Теперь добавим к новому слою стиль. Для этого нужно дважды щёлкнуть по слою в палитре слоёв или нажать соответствующую кнопку внизу. Выберите Bevel&Emboss и примените следующие настройки:



Уменьшим прозрачность слоя с перегибом примерно до 50%. Поднастроим параметры других слоев по вкусу



Задание 8. Эффект старого фото

1. Откройте любую фотографию



2. Создайте слой с радиальным градиентом от прозрачного до полупрозрачного черного.

Продублируйте этот слой

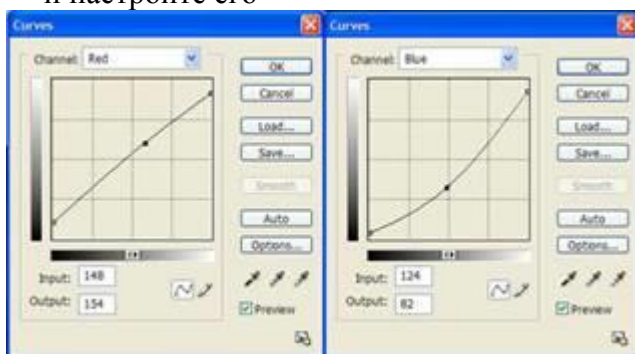


3. Измените режим смешивания обоих градиентов на Soft light

4. Добавьте слой настроек с кривыми Curves



и настройте его



5. Подберите старящую текстуру, перетяните в наше фото верхним слоем



6. Обесцветьте этот слой: Image > Adjustments > Desaturate и поставьте ему режим наложения Hard light, подберите прозрачность слоя



7. Создайте новый слой. Поставьте цвет переднего плана #f5a700 и кистью сделайте

несколько линий



8. Размойте этот слой: Filter>Blur>Motion Blur (Angle -90, Distance 999)



9. Поставьте режим наложения Vivid light



Задание 9. Из цветного в триколор

1. Откроем нужную фотографию



2. Выделим нужный фрагмент и скопируем на новый слой. Исходный слой можно удалить



3. Создаем новый слой и на нем выделяем фрагмент серым



4. Отключаем видимость серого слоя. На слое с нашим фрагментом выделяем темные участки: Select>Color Range, Shadows



5. Создаем новый слой и на нем заливаем выделение черным



6. Отключаем также видимость слоя с темными участками. На слое с нашим фрагментом выделяем светлые участки: Select>Color Range, Lights. Создаем новый слой и на нем заливаем выделение белым



7. Теперь включаем видимость чёрного и серого слоёв



8. Теперь можно по желанию добавить фон, тонировать изображение и т.п.



Задание 10. Имитация живописи. Подсолнухи а'ля Ван Гог

1. Откройте исходную фотографию



2. Замена холодного темного фона на более теплые оттенки. Выделите фон Волшебной Палочкой. Для удобства можно использовать только один канал, а также работать с маской. Не снимая выделения создайте новый слой, добавьте к нему маску, снимите значок цепочки в палитре слоев между пиктограммой слоя и маски. Этот слой будет использоваться для рисования фона, а маска должна защитить сам букет. Залейте верхний слой цветом #64ACA1
10. Добавьте рамку, подберите к ней стили



Задание 11. Преобразуем фото в картину

1. Откроем нужную фотографию



2. Создаем 3 копии фонового слоя, отключаем видимость двух верхних, встаем на второй снизу слой

3. Filter > Artistic > Palette Knife (6, 3, 0).
Потом Image > Adjust > Hue/Saturation (0, 70, 5. Без Colorize) и Filter > Blur > Gaussian Blur (радиус 4,0)



4. Сделайте активным второй слой, который мы дублировали. Filter > Artistic > Dry Brush (2, 8, 1)



5. Сделайте активным третий слой, который мы дублировали. Filter > Blur > Smart Blur (15, 50, Low, Edge Only). Сейчас изображение выглядит как белые линии по черному фону. Теперь, необходимо сделать инверсию (Ctrl+I) для получения эффекта «черным по белому»



6. Для усиления эффекта рисунка можно применить Filter > Artistic > Poster Edges (2, 1, 2)



7. Изменяем режим смешивания всех сдублированных слоев на Soft Light
1. Откроем нужную фотографию



2. Обесцветим ее (Ctrl+Shift+U)



3. Filter > Custom > High Pass, радиус примерно 4



4. Скопируйте слой, примените режим смешивания Color Dodge. Увеличьте контраст: Image > Adjustment > Brightness/Contrast



5. Filter > Blur > Smart Blur (примерные параметры: Radius 20, Threshold 30)



Задание 13. Плакат в стиле шестидесятых

1. Создайте новый документ для постера любого размера. На новом слое кистью создаем 53

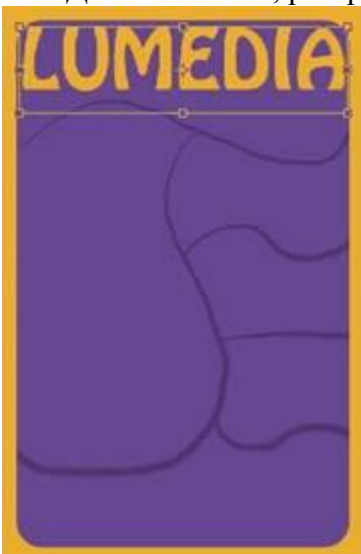
кривые сектора для текстов и изображения



2. Залейте фон желто-оранжевым. Создайте на новом слое контур с помощью Rounded Rectangle Tool. Загрузите его как выделение и залейте фиолетовым



3. Добавьте текст, растрируйте его и подгоните размеры



4. Деформируем текст так, что бы он соответствовал сектору: Edit > Transform > Warp.

Подгоняем весь текст постера под отведенные ему секторы



5. Меняйте цвет и другие параметры текста. Вместо некоторых надписей оставьте их обводку



6. Добавим эффект от старой печати.
Скопируйте слой с текстом, залейте его вторичным цветом и сдвиньте вниз и вправо
8. Добавим в наш плакат фотографию



9. Превратим фотографию в постер.

Обесцветьте, потом Image > Adjustment > Posterize. Получив несколько цветов, залейте их

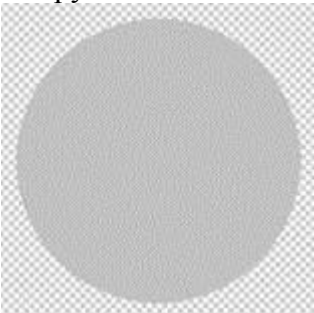


10. Перенесите постер в наш плакат под слой с текстурой, трансформируйте, при необходимости доработайте ластиком

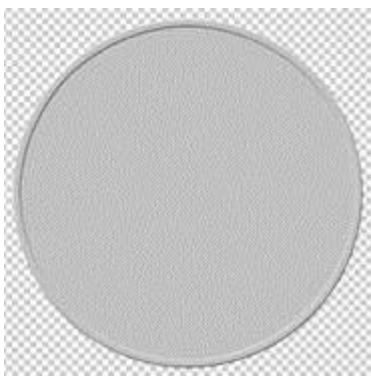


Задание 14. Монета

1. Создаем новое изображение 500 x 500 с прозрачным фоном. На новом слое создаем круглое выделение по центру и заливаем его серым (#C2C2C2)



2. Для создания кромки монеты уменьшаем выделение на 10 пкс (Select > Modify > Contract), инвертируем выделение (Ctrl+Shift+I) и вырезаем получившийся край на новый слой. Применяем к новому слою стиль Bevel and Emboss, настраиваем



3. Откройте изображение, которое вы будете использовать в качестве портрета для вашей монеты



и превратите его в эскиз по алгоритму, приведенному в задании Преобразуем фото в эскиз (Задание 12)



4. Перенесите нужный фрагмент эскиза в монету, выделите на нем оттенки белого (Select > Color Range) и удалите их. К слою с оставшимися контурами примените стиль Bevel and Emboss и настройте



5. Добавьте текст. Для его искажения используйте кнопку Create warped text на панели настроек. Примените стиль Bevel and Emboss и настройте



6. Добавляем по вкусу цвета, тени, текстуры и т.п.

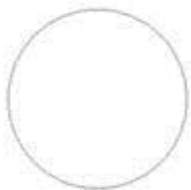


Лабораторная работа №7

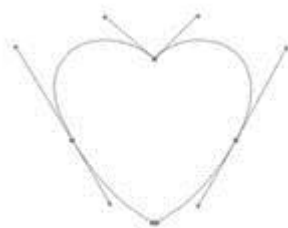
Тема: Рисование

Задание 1. Сердце

1. Создать новый документ 800x600 с белым фоном
2. Делаем круглую выделенную область, на вкладке Paths кнопкой Make work path from selection преобразуем выделение в векторный контур



3. С помощью инструментов для работы векторной графикой Convert Point Tool, Direct Selection Tool, Path Selection Tool и т.д. превращаем круглый контур в сердце

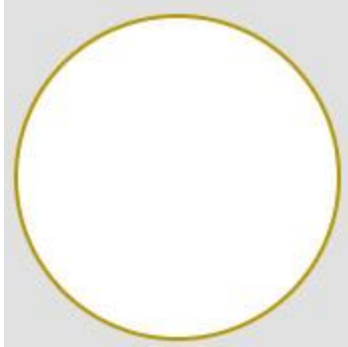


4. На вкладке Paths кнопкой Load path as a selection преобразуем векторный контур обратно в выделение
5. Создаем для сердца новый слой, заливаем его красным
6. Из палитры Color-Swatches-Styles используйте встроенный стиль Голубое стекло (Кнопка) для объема
7. В палитре слоев меняем стили: Color Overlay установить красным; в Inner Glow понизить Opacity до 30%; в разделе Bevel and Emboss в настройке Shading поставить Angle=150 Altitude=64. Добавить стиль Outer Glow для размытия



Задание 2. 3D - Смайлик

1. Создайте новый документ 500x500 с серым фоном. На новом слое создайте белый круг с обводкой шириной 2 px и цветом #B59605



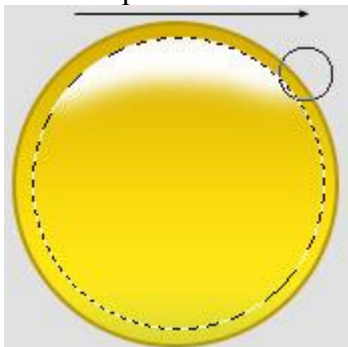
2. Примените к окружности нормальный линейный градиент снизу вверх



Цвета слева направо: #D8D635 -> #FFE617 -> D6AD00



3. На новом слое создайте выделение, на 3 px меньше основной окружности, и закрасьте его верхнюю часть белой мягкой кистью



4. Уменьшите прозрачность верхнего слоя



5. Для глаза создайте новый слой с белой заливкой



6. Залейте глаз серо-белым градиентом, примените обводку шириной 2 px и цветом #B59605, добавьте белый штрих. Скопируйте глаз и отразите его по горизонтали

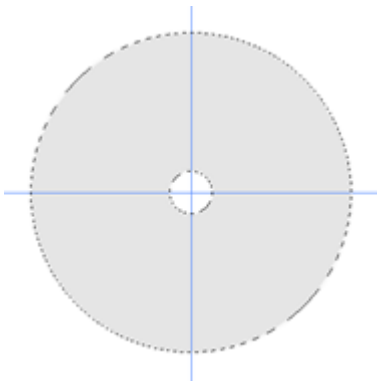


7. На разных слоях нарисуйте брови, глаза и рот. Используйте такие эффекты слоя, как Тени, Свечения, Рельефность и т.п. Можно добавить несколько штрихов

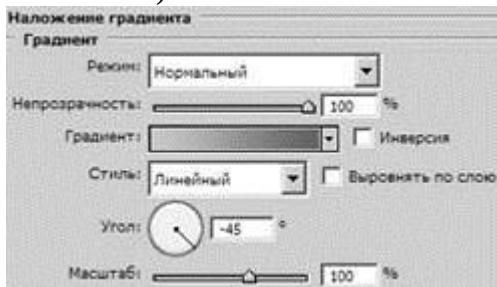


Задание 3. CD

1. Создайте новое изображение размером 400 x 400 с белым фоном. Установите вспомогательные линии ровно по середине (включите линейку Ctrl+R). Создайте новый слой
2. Выделите из центра (удерживая Shift и Alt) окружность 340 x 340. Установите режим «Вычитание из выделенной области» и выделите окружность 44 x 44. Получившуюся фигуру залейте цветом #E5E5E5



3. Из центра диска вычтите окружность 106 x 106, полученную область вырежьте на новый слой. К слою примените стиль Gradient Overlay (цвета градиента: от #CCCCCC к #526577)



4. Выделите правую половину изображения и на новом слое примените градиент Spectrum



5. Примените к этому слою трансформацию в режиме перспективы, передвиньте верхний левый узелок к центру



6. Продублируйте этот слой, копию отразите по горизонтали. Объедините эти два слоя, продублируйте, копию поверните на 90 градусов. Объедините эти два слоя, к получившемуся слою примените стиль наложения Overlay, прозрачность слоя 70%



7. Создайте новый корректирующий слой Hue/Saturation (Цветовой тон/Насыщенность) с прозрачностью 30% и следующими настройками: Colorize, Hue: 227, Saturation: 44, Lightness: -8



8. Выделите в центре слоя со спектральным градиентом окружность диаметром 132 пикселя и вырежете ее на новый слой. Примените к нему режим наложения Screen, прозрачность 60%



9. Примените к некоторым слоям стиль Обводка, настройте параметры



Задание 4. Печать

1. Создайте новый документ 300x300 с белым фоном. Напишите недлинную строчку текста

THE DESIGN WORLD . COM *

2. Выделите слой с текстом, инструмент Text и щелкните на панели сверху по кнопке с буквой T и дугой под ней. Установите следующие значения



3. Скопируйте слой и поверните копию на 180 градусов (Edit > Transform > Rotate 180). Переместите копию так, чтобы она располагалась под оригиналом



4. Создайте новый слой. Выделите область с текстом инструментом Elliptical Marquee. Затем Edit > Stroke > 5px черного цвета



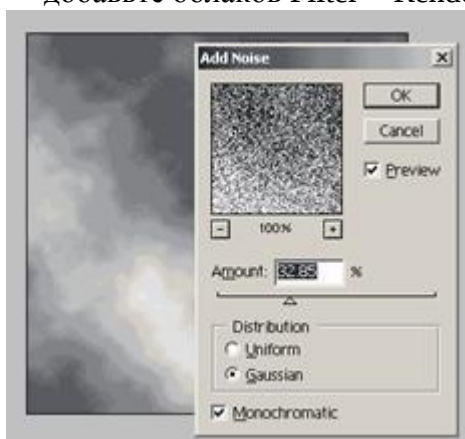
5. Повторите действие, только уже с окружностью поменьше



6. На новом слое напишите нужный текст или даже какое-нибудь изображение для центра печати



7. Создайте новый слой поверх остальных. Поставьте цвета по умолчанию (кнопка D), добавьте облаков Filter > Render > Clouds, а затем Filter > Noise > Add Noise



8. Выделите полученный слой с зашумленными облаками и смените его режим смешивания на Screen



9. При желании поверните немного всю печать: сведите слои и Image > Rotate Canvas > Arbitrary...

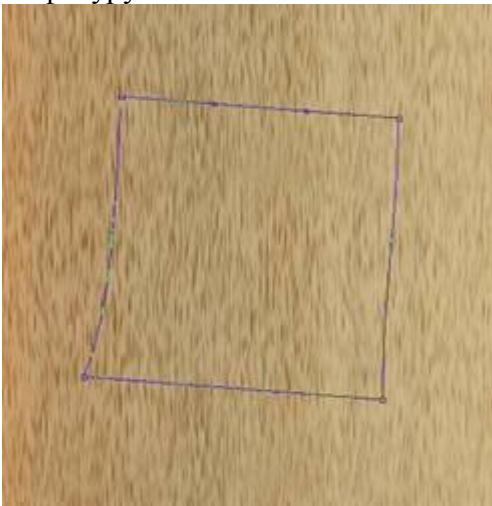


Задание 6. Клейкий листок для заметок

1. Создаем новый документ, заливаем фон любым цветом или картинкой. Можно использовать Layer Styles



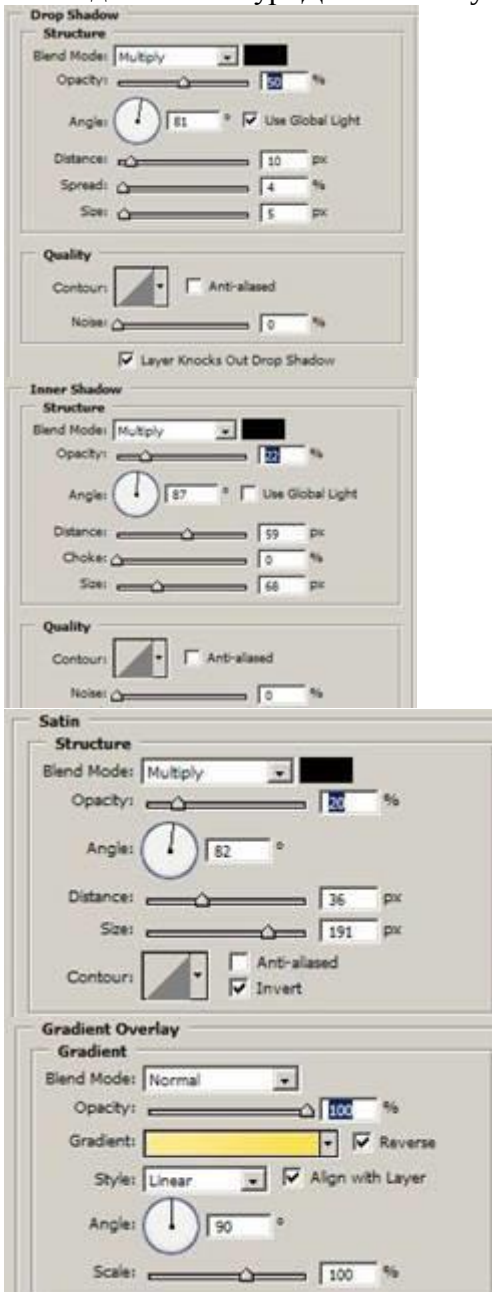
2. Выберите инструмент Pen в режиме Paths. Создайте новый слой, а на нем подобную фигуру



3. Заполните контур белым, кликнув на нем правой кнопкой мыши и выбрав пункт «Fill Path»



4. Удалите контур. Добавьте Layer Styles со следующими настройками:



Ваш клейкий листок сейчас должен выглядеть примерно так:



5. Добавьте немного текста на наш листок, поверните текст вправо на несколько градусов, чтобы его угол совпадал с углом наклона листка



Лабораторная работа 1: Основы работы с Figma и создание простого макета сайта

Цель: Ознакомиться с интерфейсом Figma и научиться создавать простые макеты веб-страниц для понимания процесса дизайна интерфейсов.

1. Создание нового проекта и подготовка рабочего пространства

1. Регистрация и вход в Figma:

- Перейдите на сайт [Figma](https://figma.com).
- Зарегистрируйтесь, если у вас нет аккаунта, или войдите в существующий аккаунт.

2. Создание нового файла:

- После входа в Figma выберите кнопку **New File** в правом верхнем углу.
- Это откроет новый рабочий документ, где вы будете создавать свой макет.

3. Выбор фрейма (Frame):

- В Figma вместо обычных "артбордов" используется **Frame**. В панели инструментов выберите инструмент **Frame** или нажмите клавишу F.
- В Figma есть предустановленные размеры для различных устройств (например, для десктопа, планшетов, мобильных устройств). Для этой лабораторной работы выберите **Desktop (1440x1024)**, так как мы создаем макет для десктопной версии сайта.

2. Создание элементов макета

1. Добавление логотипа:

- Нарисуйте простую фигуру в виде круга с помощью инструмента **Ellipse (O)**, чтобы сделать логотип.
- Залейте его цветом, например, синим или зеленым.
- Добавьте текст в логотип с помощью инструмента **Text (T)** и напишите название вашего сайта или компании, например, "My Website".

2. Создание навигационного меню:

- Нарисуйте **прямоугольник** с помощью инструмента **Rectangle (R)**, который будет фоном для меню. Задайте высоту 60px и ширину 1440px.
- Разместите этот прямоугольник в верхней части фрейма.
- Используя инструмент **Text (T)**, добавьте элементы навигации: "Главная", "О нас", "Контакты". Расположите их горизонтально в меню.

3. Добавление главного баннера:

- Нарисуйте прямоугольник с размерами 1440px на 400px. Это будет ваш главный баннер на странице.
- Залейте его изображением (можно взять картинку с вашего компьютера и загрузить её в Figma с помощью **Place Image**). Например, используйте изображение с пейзажем или фотографию, соответствующую теме сайта.

4. Создание блоков с контентом:

- Нарисуйте несколько прямоугольников, которые будут представлять собой карточки с контентом (например, для блоков с товарами, услугами или информацией).
- Для этого используйте инструмент **Rectangle (R)** с размерами 400x300px. Эти блоки будут располагаться под баннером.

5. Добавление кнопок:

- Нарисуйте небольшие прямоугольники с закругленными углами для кнопок (например, "Купить", "Подробнее").
- С помощью инструмента **Text (T)** добавьте текст на кнопку, например, "Подробнее".

3. Применение стилей и выравнивание

1. Настройка цветов и стилей:

- Для фона, кнопок и текста используйте основные цвета, соответствующие вашему дизайну. Выберите 2-3 основных цвета для сайта.
- Задайте стиль для текста: выберите шрифт (например, Arial, Roboto) и настройте его размер, жирность, межстрочное расстояние.
- Чтобы не делать это вручную для каждого элемента, создайте **стили** для текста и цветов. Для этого выберите текстовый элемент, нажмите на иконку стилистического меню и создайте новый стиль.

2. Выравнивание элементов:

- Для выравнивания элементов можно использовать инструменты выравнивания в верхней панели Figma. Убедитесь, что все элементы (текст, кнопки, блоки) выровнены по сетке или относительно друг друга.
- Например, вы можете выровнять меню по центру, а баннер растянуть на всю ширину страницы.

3. Использование направляющих:

- Чтобы точно выровнять элементы на странице, вы можете использовать **направляющие**. Для этого перетащите их с верхней и левой панели, чтобы выровнять элементы по вертикали и горизонтали.

4. Создание прототипа страницы (интерактивность)

1. Настройка переходов:

- Для того, чтобы ваш макет был интерактивным, вам нужно настроить прототип.
- Переключитесь на вкладку **Prototype** в правом верхнем углу.
- Создайте взаимодействие, например, переход на другую страницу при клике на кнопку "Подробнее". Для этого выделите кнопку, нажмите на маленький круг в правом верхнем углу и протяните его к следующей странице или фрейму.

2. Тестирование прототипа:

- Нажмите на кнопку **Present** в правом верхнем углу Figma, чтобы протестировать свой прототип.
- Убедитесь, что кнопки и переходы работают как ожидалось.

5. Экспорт макета

1. Подготовка к экспортированию:

- Когда вы завершите макет, его можно экспортировать.
- Выберите фрейм или отдельные элементы, которые хотите экспортировать, и нажмите правой кнопкой мыши. Затем выберите **Export**.
- Укажите формат файла, например, PNG или PDF, и нажмите **Export**.

2. Сохранение проекта:

- Не забывайте сохранять проект, чтобы он был доступен для дальнейшей работы. В Figma проект сохраняется автоматически, но вы можете также экспортировать файл, чтобы передать его коллегам или преподавателю.

6. Завершающий этап

- Проверьте, что все элементы размещены правильно, а текст читаем. Убедитесь, что страницы не перегружены информацией и выглядят аккуратно.
- Пересмотрите макет на предмет выравнивания и правильного использования цветов и шрифтов.
- Подготовьте макет для сдачи, экспортируя его в необходимом формате (например, PNG или PDF), если нужно.

Ресурсы для углубленного изучения

- **Онлайн-курс:** [Основы дизайна интерфейсов в Figma на Coursera](#)
- **Книга:** "Дизайн как искусство" Дэвида Кэди (представление принципов дизайна и композиции)
- **Сайт:** Color Theory — теоретические основы использования цвета в дизайне

Лабораторная работа 2: Создание макета мобильного приложения в Figma

Цель: Научиться создавать мобильный макет для приложения с использованием Figma, освоить принципы адаптивного дизайна и разработку интерфейсов для мобильных устройств.

1. Создание нового проекта и подготовка рабочего пространства

1. Открытие Figma и создание нового файла:

- Перейдите на сайт [Figma](#), если не сделали этого ранее.
- Войдите в свой аккаунт и создайте новый проект, кликнув на кнопку **New File** в правом верхнем углу.
- Это откроет пустую рабочую область, где вы будете создавать макет мобильного приложения.

2. Выбор фрейма для мобильного устройства:

- В Figma используйте **Frame** для создания рабочего пространства. Выберите инструмент **Frame** (или нажмите клавишу F).
- В панели справа выберите шаблон для мобильного устройства (например, **iPhone 11/12/13** или **Pixel 5**). Эти размеры предустановлены в Figma и подходят для создания макетов мобильных приложений.
- Размеры фрейма для **iPhone 11** составляют **375 x 812 px** (ширина x высота), что соответствует стандартному размеру экрана.

2. Создание структуры приложения

1. Добавление верхней панели (Header):

- Нарисуйте прямоугольник для верхней панели с помощью инструмента **Rectangle (R)**. Установите размеры 375x100px для ширины и высоты.
- Расположите панель в верхней части экрана.
- Задайте ей темный фон (например, #333333) и добавьте на нее элементы, такие как логотип и иконку меню.
 - Используйте инструмент **Text (T)**, чтобы написать название вашего приложения (например, "AppName").
 - Нарисуйте иконку меню в виде трех горизонтальных линий (гамбургер-меню) с помощью инструмента **Line** или используйте готовую иконку.

2. Добавление основного контента:

- Разделите экран на несколько блоков. Для этого нарисуйте несколько прямоугольников, которые будут служить контейнерами для контента.
 - Например, добавьте блок с изображением или картинкой, которая будет служить баннером.
 - Размер изображения может быть **375 x 200 px** (ширина x высота).

3. Создание кнопок (CTA - Call to Action):

- Нарисуйте прямоугольники для кнопок с закругленными углами. Эти кнопки могут быть использованы для переходов в другие части приложения.
- Для этого используйте инструмент **Rectangle (R)** и установите радиус углов для кнопки (например, 20px), чтобы она выглядела как современная кнопка.
- Залейте кнопку цветом (например, #007BFF для синего цвета) и добавьте текст с помощью инструмента **Text (T)**. Например, кнопка "Подробнее", "Перейти" или "Регистрация".

4. Добавление списка элементов:

- Создайте список карточек, которые будут отображаться в приложении. Например, это могут быть карточки с товарами или услугами.
- Для этого используйте прямоугольники размером 375x150px. Внутри каждого прямоугольника добавьте текст, изображение и цену, чтобы имитировать карточку товара.
 - Используйте инструмент **Text (T)** для добавления названия и описания товара.
 - Для изображения используйте **Place Image** и загрузите картинку с вашего устройства.

3. Применение стилей и выравнивание

1. Настройка цветов и шрифтов:

- Убедитесь, что все элементы соответствуют цветовой палитре вашего приложения. Используйте **2-3 основных цвета** для фона, кнопок и текста.
- Настройте шрифты: выберите подходящий шрифт (например, **Roboto, Helvetica Neue**) и установите размер шрифта для заголовков, основного текста и кнопок.

2. Выравнивание элементов:

- Для правильного выравнивания используйте инструменты выравнивания в верхней панели Figma (например, выравнивание по центру, по левому краю и т.д.).
- Все элементы должны быть выровнены по сетке или относительно друг друга, чтобы макет выглядел аккуратно и гармонично.

3. Использование направляющих и сетки:

- Включите сетку, чтобы помочь выравнивать элементы на экране. Для этого откройте **Layout Grid** в правой панели Figma и выберите тип сетки (например, **Column Grid** или **Rows Grid**).
- Направляющие и сетки помогут вам точно размещать элементы на странице, обеспечивая симметричность и соответствие дизайнерским стандартам.

4. Создание прототипа и интерактивности

1. Добавление переходов между экранами:

- Переключитесь на вкладку **Prototype** в правой панели.
- Добавьте интерактивность: например, к кнопке "Подробнее" настройте переход на другой экран с подробной информацией о товаре или услуге.
 - Для этого выделите кнопку, затем в правом верхнем углу нажмите на маленький кружок и протяните его на другой экран. В настройках перехода выберите **On Click** и тип анимации (например, **Instant** или **Dissolve**).

2. Тестирование прототипа:

- Чтобы протестировать макет, нажмите на кнопку **Present** в правом верхнем углу. Это откроет режим презентации, в котором вы можете взаимодействовать с макетом, как с реальным приложением.
- Убедитесь, что все кнопки работают корректно и переходы между экранами происходят без ошибок.

5. Экспорт макета

1. Экспорт изображения или макета:

- Когда макет готов, его можно экспортировать.
- Выберите элементы, которые хотите экспортировать, или весь фрейм, и нажмите **Export** в правом нижнем углу Figma.
- Выберите формат файла (например, PNG, JPG) и нажмите **Export**.

2. Подготовка к сдаче:

- Перед сдачей проекта убедитесь, что все элементы правильно выровнены и отображаются.
- Экпортируйте макет в формате **PNG** или **PDF**, чтобы передать его преподавателю или коллегам.

6. Завершающий этап

- Проверьте, что все элементы макета расположены правильно, кнопки работают корректно, а весь текст читаем.
- Убедитесь, что макет выглядит гармонично и соответствует требованиям мобильного интерфейса (например, наличие доступных кнопок и правильных размеров).
- Экпортируйте макет и сохраните его в удобном для вас формате.

Ресурсы для углубленного изучения

- **Онлайн-курс:** [Веб-дизайн для мобильных приложений на Coursera](#)
- **Книга:** "Мобильный дизайн. Руководство для профессионалов" (Эрик Мейер, Джошуа Дэвис) — книга о создании интерфейсов для мобильных устройств.
- **Сайт:** Material Design Guidelines — официальные рекомендации от Google по созданию мобильных интерфейсов.

Лабораторная работа 3: Создание адаптивного макета веб-сайта в Figma

Цель: Научиться разрабатывать адаптивные (responsive) макеты для веб-сайта, который будет корректно отображаться на различных устройствах (мобильных телефонах, планшетах и десктопах).

1. Создание нового проекта и подготовка рабочего пространства

1. Создание нового файла:

- Откройте Figma и создайте новый файл, кликнув на кнопку **New File**.
- Откроется пустой рабочий документ, где вы будете создавать адаптивный макет.

2. Выбор фреймов для разных устройств:

- В Figma создаются **фреймы (Frames)**, которые будут представлять собой рабочие области для различных устройств.
- Выберите инструмент **Frame (F)** в панели инструментов или нажмите F на клавиатуре.
- В правой панели Figma выберите шаблоны для разных устройств. Начните с создания фреймов для трёх популярных размеров экранов:
 - **Desktop:** 1440x1024 px
 - **Tablet:** 768x1024 px

- **Mobile:** 375x667 px
- Эти размеры фреймов станут основой для вашего адаптивного дизайна.

2. Структура веб-страницы (Основные элементы)

1. Создание базовых элементов страницы:

- На каждом фрейме (Desktop, Tablet, Mobile) создайте основные элементы страницы:
 - **Шапка (Header):** Это верхняя панель с логотипом и навигацией.
 - **Главный баннер:** Большое изображение или слайдер, который будет размещен в верхней части страницы.
 - **Блок контента:** Разместите несколько карточек с товарами или услугами.
 - **Подвал (Footer):** В нижней части страницы разместите контактную информацию, ссылки на социальные сети и другие элементы.

2. Рисуем Header (Шапку):

- Используйте **Rectangle (R)** для создания фона шапки. Например, сделайте высоту 80px для Desktop, 60px для Tablet и 50px для Mobile.
- Добавьте логотип с помощью инструмента **Text (T)** или создайте простую графику (например, круг или прямоугольник).
- Добавьте элементы навигации — для Desktop и Tablet это могут быть горизонтальные ссылки, для Mobile — иконка меню (гамбургер).

3. Главный баннер:

- Для баннера используйте **Rectangle (R)** с размерами 1440x400 px для Desktop, 768x300 px для Tablet и 375x250 px для Mobile.
- Залейте баннер изображением с помощью инструмента **Place Image**.
- Выберите картинку, которая будет хорошо смотреться в различных разрешениях.

4. Контент (карточки):

- Нарисуйте прямоугольники для карточек. Для Desktop это могут быть 3 карточки по ширине экрана, для Tablet — 2, для Mobile — 1.
- Каждая карточка будет содержать изображение, название и краткое описание. Используйте инструмент **Text (T)** для добавления текста и **Place Image** для вставки изображения.

5. Подвал (Footer):

- Нарисуйте прямоугольник для подвала с размерами 1440x120 px для Desktop, 768x100 px для Tablet и 375x80 px для Mobile.
- В подвале добавьте текст с контактной информацией и ссылки на социальные сети.

3. Применение стилей и выравнивание

1. Настройка цветов и шрифтов:

- Для всех элементов выберите подходящие цвета. Например, используйте светлый фон и темный текст для легкости восприятия.
- Настройте шрифты: для заголовков используйте более крупный шрифт (например, 24 px для Desktop, 20 px для Tablet, 16 px для Mobile), а для основного текста — меньше (например, 14 px для всех устройств).

2. Использование стилей:

- В Figma вы можете создавать **стили** для цветов и текста. Это позволяет вам быстро применять одинаковые стили ко всем элементам.
 - Выделите текст, настройте шрифт и размер, затем сохраните как стиль. Так вы сможете использовать его для всех заголовков на странице.
 - То же самое для цветов фона, текста, кнопок и других элементов.

3. Выравнивание элементов:

- Используйте **инструменты выравнивания** в Figma для того, чтобы все элементы страницы располагались симметрично.
- Выравнивайте элементы по горизонтали и вертикали с помощью панелей выравнивания или включите **Layout Grid** (сетка), чтобы точно расположить элементы.

4. Адаптация дизайна для разных устройств

1. Адаптация для Desktop:

- На фрейме Desktop создайте основной макет с широкими элементами: много информации, большие изображения и панели.
- Сконцентрируйтесь на создании гармоничного и удобного интерфейса с достаточным количеством пространства для каждого элемента.

2. Адаптация для Tablet:

- Для Tablet уменьшите размер шрифтов и уменьшите количество информации в блоках.
- Измените количество колонок: например, если на Desktop карточки размещаются в 3 колонки, на Tablet сделайте 2.
- Убедитесь, что элементы не вылазят за края экрана и что дизайн выглядит компактно, но читаемо.

3. Адаптация для Mobile:

- На Mobile сделайте макет компактным и максимально упрощенным.
- Уменьшите размеры изображений, уменьшите шрифт и обеспечьте, чтобы все важные элементы (например, кнопки и ссылки) легко нажимались пальцем.
- Преобразуйте меню в гамбургер-меню, чтобы оно не занимало много места на экране.

- Убедитесь, что элементы имеют достаточно отступов, чтобы пользователь случайно не нажал на лишние кнопки.

5. Прототипирование (интерактивность)

1. Добавление взаимодействий между экранами:

- Перейдите в вкладку **Prototype** в правой панели Figma.
- Создайте интерактивные переходы между экранами: например, клик по кнопке в **Header** может открывать подменю или переходить на новую страницу.
- Выберите элементы (например, кнопку), а затем нарисуйте соединительную линию на другие экраны (например, на экран с подробным описанием товара или услуги).

2. Настройка переходов:

- Убедитесь, что переходы между фреймами настроены корректно. Выберите **On Click** или **While Hovering**, если хотите, чтобы элементы реагировали на действия пользователя.
- Проверьте, как выглядит сайт при переходах между экранами, чтобы убедиться в корректности адаптивности.

6. Экспорт макета

1. Экспорт элементов:

- Когда макет готов, его нужно экспортировать для передачи другим участникам команды или преподавателю.
- Выделите элементы, которые хотите экспортировать, и нажмите **Export** в правом нижнем углу.
- Выберите формат файла (например, PNG, JPG, PDF) и укажите параметры экспорта.

2. Экспорт прототипа:

- Для тестирования и демонстрации макета вы можете экспортировать весь прототип в виде видео или в формате GIF.
- Для этого нажмите на кнопку **Present** в правом верхнем углу, чтобы просмотреть взаимодействия, а затем используйте функцию записи экрана.

Завершающий этап

- Проверьте, как макет выглядит на разных устройствах и убедитесь, что он выглядит корректно. Убедитесь, что все элементы дизайна соответствуют требованиям адаптивного дизайна.
- Экпортируйте макет и сохраните его в нужном формате (например, PNG или PDF).

Ресурсы для углубленного изучения

- **Онлайн-курс:** Responsive Web Design Fundamentals — курс по основам адаптивного дизайна.
- **Книга:** "Не заставляйте меня думать. О проектировании удобных веб-сайтов" Стива Круга — книга о принципах удобства использования.

- **Сайт:** A List Apart: Responsive Design — статьи по адаптивному дизайну.

Лабораторная работа 4: Создание интерактивных элементов в Figma

Цель: Научиться создавать и интегрировать интерактивные элементы (кнопки, формы, модальные окна и т.д.) в макет веб-сайта, используя Figma.

1. Создание нового проекта и подготовка рабочего пространства

1. Создание нового файла:

- Откройте Figma и создайте новый файл, кликнув на кнопку **New File**.
- На экране появится пустой рабочий документ.

2. Выбор фреймов для проекта:

- В качестве основы создайте фреймы для трех типов устройств: **Desktop**, **Tablet** и **Mobile**.
- Для Desktop используйте размер **1440x1024 px**, для Tablet — **768x1024 px**, для Mobile — **375x667 px**.
- Эти фреймы будут служить основой для вашего макета, и вы будете добавлять интерактивные элементы на них.

2. Добавление интерактивных элементов (кнопок и форм)

1. Создание кнопок:

- Нарисуйте кнопку с помощью инструмента **Rectangle (R)**. Например, создайте прямоугольник с размерами 200x50 px для Desktop, 180x50 px для Tablet и 150x40 px для Mobile.
- Используйте стильный шрифт для текста на кнопке, например, **Arial Bold**, с размером шрифта 16px.
- Разместите текст внутри кнопки, например, "Прочитать больше" или "Подписаться".
- Добавьте стили для кнопки:
 - Залейте кнопки цветом (например, синий цвет #007BFF).
 - Установите скругления углов (например, 8 px).
 - Добавьте эффект наведения: при наведении кнопка должна немного изменять свой цвет, например, становиться темнее.

2. Добавление формы для ввода данных:

- Нарисуйте прямоугольники для полей ввода, например, для **имени**, **email** и **сообщения**.
- Для каждого поля используйте **Rectangle (R)** с размерами 320x40 px для Desktop, 280x40 px для Tablet и 240x40 px для Mobile.
- Внутри прямоугольников добавьте текстовые метки (например, "Имя", "Email") с помощью инструмента **Text (T)**.
- Добавьте кнопку **Отправить** под полями ввода.

3. Создание выпадающего списка:

- Создайте выпадающий список с помощью **Rectangle (R)** и **Text (T)**.
- Нарисуйте прямоугольник с размерами 320x40 px для Desktop, 280x40 px для Tablet и 240x40 px для Mobile.
- Добавьте текст, например, "Выберите страну", чтобы показать, что это выпадающий список.
- Для отображения раскрывающегося меню используйте **Vector (P)** для создания стрелочки.

4. Создание радио-кнопок и чекбоксов:

- Для радио-кнопок и чекбоксов создайте маленькие круги с помощью **Ellipse (O)**.
- Нарисуйте круг с размером 20x20 px для радио-кнопки и для чекбокса.
- Добавьте текст рядом с каждым элементом, например, "Согласен с условиями" для чекбокса или "Мужчина/Женщина" для радио-кнопки.

3. Добавление модальных окон и взаимодействий

1. Создание модального окна:

- Модальные окна (или попапы) используются для отображения дополнительной информации без необходимости переходить на другую страницу.
- Нарисуйте прямоугольник, который будет представлять само окно (например, размер 600x400 px для Desktop, 400x300 px для Tablet и 300x250 px для Mobile).
- Внутри окна добавьте заголовок с текстом, например, "Подтверждение", и текстовое описание, например, "Вы уверены, что хотите продолжить?".
- Добавьте две кнопки внутри окна — например, "Да" и "Нет". Создайте кнопки аналогично предыдущим шагам (с прямоугольниками и текстами).

2. Настройка модального окна для взаимодействия:

- В Figma можно настроить переходы между экранами, чтобы имитировать открытие и закрытие модального окна.
- Перейдите во вкладку **Prototype** в правой панели.
- Выберите кнопку "Открыть окно" на главной странице и соедините её с фреймом, на котором расположено модальное окно. Настройте переход **On Click → Open Modal**.
- Для кнопки "Закрыть" в модальном окне создайте переход обратно, чтобы окно закрывалось при нажатии.

4. Настройка интерактивности (прототипирование)

1. Интерактивные кнопки:

- Перейдите в режим **Prototype** для настройки переходов между экранами.

- Для кнопки "Прочитать больше" установите переход на новую страницу или фрейм с дополнительной информацией. Например, при нажатии на кнопку будет открываться фрейм с подробным описанием товара или услуги.
- Настройте действия для кнопок, таких как **On Hover** (при наведении) или **On Click** (при клике).

2. Интерактивность форм:

- Для формы добавьте переходы, чтобы кнопка **Отправить** вела к фрейму с благодарностью за отправку данных.
- Для поля ввода можно настроить изменения текста или фокуса на другом элементе.

3. Использование горячих точек:

- Создайте **горячие точки (hotspots)** на изображениях или кнопках. Это области, которые можно "кликать", и они будут действовать как элементы интерфейса.
- Убедитесь, что все элементы с интерактивностью настроены так, чтобы пользователь мог взаимодействовать с интерфейсом, как если бы это был реальный веб-сайт.

5. Экспорт и финальные шаги

1. Экспорт макета:

- Когда макет готов, его нужно экспортировать для демонстрации или передачи заказчику.
- Выделите нужные элементы (например, кнопки, формы, модальные окна), затем нажмите **Export** в правом нижнем углу.
- Экспортируйте макет в нужном формате (например, PNG, JPG или PDF) для последующего использования.

2. Просмотр прототипа:

- Чтобы протестировать интерактивные элементы, нажмите на **Present** в правом верхнем углу.
- Просмотр будет демонстрировать ваш макет с настройками переходов и интерактивности. Убедитесь, что все элементы работают как задумано.
- Для более подробного тестирования можете записать видео или GIF.

Завершающий этап

- Проверьте, как работают все интерактивные элементы в вашем макете: кнопки, формы, выпадающие списки, радио-кнопки и чекбоксы.
- Убедитесь, что модальные окна открываются и закрываются правильно, а все формы отправляются на соответствующие страницы или показывают результат.
- Экспортируйте проект и сохраните его для отправки или представления.

Ресурсы для углубленного изучения

- **Книга:** "Don't Make Me Think" Стива Круга — книга о принципах удобства использования веб-сайтов, включая создание интерактивных элементов.
- **Онлайн-курс:** Figma for Beginners — курс по изучению Figma, который охватывает создание интерактивных элементов.

Раздел 2 Верстка страниц

Работа 1 СОЗДАНИЯ КАРТОЧЕК ТОВАРОВ

CSS (Cascading Style Sheets) — это каскадные таблицы стилей, которые используются для оформления HTML-элементов. С помощью CSS мы задаем внешний вид, такие как цвета, шрифты, размеры и расположение элементов на странице.

CSS-селекторы позволяют выбирать элементы HTML, которые необходимо стилизовать.

3. Основные CSS-свойства, используемые в этой работе:

3.1. Свойство `width` и `height`

- Свойства `width` (ширина) и `height` (высота) задают размер элемента.

```
css
Копировать код
.card {
  width: 250px; /* Ширина карточки */
  height: 300px; /* Высота карточки */
}
```

- Значения можно задавать в пикселях (px), процентах (%) и других единицах измерения.

3.2. Свойство `padding`

- Свойство `padding` определяет внутренние отступы — пространство между содержимым элемента и его границей.

```
css
Копировать код
.card {
  padding: 15px; /* Внутренние отступы со всех сторон */
}
```

- Можно задавать одинаковые отступы со всех сторон или различные значения для каждой стороны:

```
css
Копировать код
padding: 10px 20px; /* 10px сверху/снизу, 20px слева/справа */
```

3.3. Свойство `margin`

- Свойство `margin` управляет внешними отступами — расстоянием между элементом и другими элементами.

```
css
Копировать код
.card {
    margin: 10px auto; /* 10px отступ сверху и снизу, авто-отступы по
бокам для центрирования */
}
```

3.4. Свойство `background-color`

- Свойство `background-color` задает цвет фона элемента.

```
css
Копировать код
.card {
    background-color: #f8f8f8; /* Светло-серый фон */
}
```

- Цвет можно задавать с помощью стандартных цветовых имен (например, `red`, `green`, `blue`), а также с помощью шестнадцатеричных значений (например, `#f8f8f8`).

3.5. Свойство `text-align`

- Свойство `text-align` управляет выравниванием текста внутри элемента.

```
css
Копировать код
.card {
    text-align: center; /* Выравнивание текста по центру */
}
```

- Также можно использовать `left` для выравнивания по левому краю и `right` для выравнивания по правому краю.

3.6. Свойство `font-size`

- Свойство `font-size` задает размер шрифта текста.

```
css
Копировать код
.card h2 {
    font-size: 1.2em; /* Размер шрифта для заголовков */
}
```

4. Как подключить CSS к HTML?

Есть несколько способов подключить CSS-файл к HTML-документу:

- **Внешний файл CSS** — наиболее распространенный способ: В HTML-документе в теге `<head>` добавляется ссылка на внешний CSS-файл:

```
html
Копировать код
<link rel="stylesheet" href="styles.css">
```

Этот способ позволяет использовать один CSS-файл для нескольких HTML-страниц.

- **Встроенные стили (inline)** — стили добавляются непосредственно в теги:

```
html
Копировать код
<p style="color: red;">Этот текст будет красным.</p>
```

Этот способ подходит для единичных случаев, но не рекомендуется для больших проектов.

- **Встроенные стили в теге <style>** — стили можно добавить в <head> внутри тега <style>, но этот метод также не подходит для крупных проектов:

```
html
Копировать код
<style>
  p {
    color: red;
  }
</style>
```

5. Как работать с кнопками?

- **Кнопка <button>** — стандартный HTML-элемент для создания кнопок. В CSS можно настроить внешний вид кнопки, например:

```
css
Копировать код
button {
  padding: 10px 20px; /* Внутренние отступы */
  background-color: #007BFF; /* Синий фон */
  color: white; /* Белый цвет текста */
  border: none; /* Без рамки */
  cursor: pointer; /* Курсор в виде руки */
}
button:hover {
  background-color: #0056b3; /* Темно-синий цвет фона при наведении */
}
```

Цель:

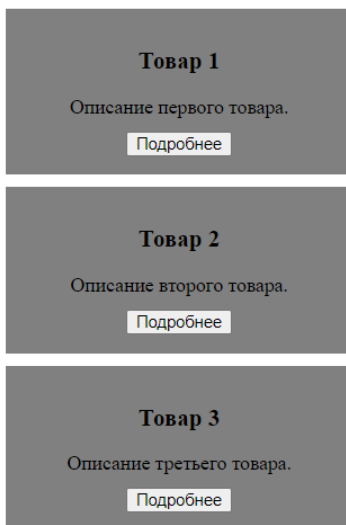
Создать HTML-документ с тремя карточками товаров и применить минимальный набор стилей через CSS.

Задание

1. Создайте HTML-документ с тремя карточками товаров.
В каждой карточке добавьте:
 - Название товара (h2);
 - Описание (p);
 - Кнопку (button) с текстом "Подробнее".
2. Подключите внешний CSS-файл.
3. Стилизируйте карточки так, чтобы:

- У всех карточек был одинаковый размер;
- У каждой карточки был светло-серый фон;
- Заголовок (h2) был чуть крупнее текста;
- Кнопка имела стандартный вид.

Примерный результат:



Работа 2 Позиционирование элементов

ЗАДАНИЕ «Яблоки и Корзины»

Скачайте файлы `cart_apples.html`, `cart_apples.css` в свою папку или на рабочий стол.

Необходимо открыть файлы для работы. Создать `div` элементы и присвоить им классы стилей зеленого, желтого и гнилого яблок. Далее необходимо используя теорию по позиционированию и файл `cart_apples.css` положить яблоки в соответствующие корзины.

1. Откройте `cart_apples.html` в браузере, чтобы увидеть исходные блоки.
2. Сконцентрируйтесь на блоке `.overlay-box` и его позиционировании.
3. Используйте `position: relative;` для родительского блока `.target-box` (если необходимо).
4. Используйте `position: absolute;` для блока `.overlay-box`.
5. Используйте свойства `top`, `right`, `bottom`, `left`, чтобы точно позиционировать `.overlay-box` внутри каждой рамки по очереди.
6. Помните, что координаты `0,0` (`top: 0;` `left: 0;`) – это левый верхний угол родительского контейнера.

7. Переместите блок `.overlay-box` внутрь каждого `.target-box` последовательно.
8. Меняйте стили в `cart_apples.css` файле.
9. Проверьте адаптивность макета и убедитесь, что все работает на разных размерах экрана.
10. Постарайтесь понять, как работает позиционирование. Не просто скопируйте решение, а постарайтесь применить знания, полученные в гайде.

Раздел 3 Разработка клиентской части информационных ресурсов

JavaScript

Лабораторная работа №1: Работа с переменными, типами данных и операторами

Цель работы:

Закрепить знания о переменных и их объявлении (`var`, `let`, `const`).

Научиться работать с основными типами данных JavaScript (`number`, `string`, `boolean`, `null`, `undefined`, `object`, `symbol`).

Понять принцип работы операторов (арифметические, сравнения, логические, присваивания).

Применять изученные концепции на практике для решения простых задач.

Задания для выполнения:

Объявление переменных:

Создайте переменные для хранения имени пользователя, его возраста и признака совершеннолетия. Используйте `var`, `let` и `const` для разных переменных, и объясните почему.

Выведите значения этих переменных в консоль.

Типы данных:

Создайте переменные разных типов (число, строка, булево значение, `null`, `undefined`, объект, символ).

Выведите тип каждой переменной в консоль, используя оператор `typeof`.

Попробуйте преобразовать переменные разных типов (например, строку в число). Выведите результаты преобразования.

Арифметические операции:

Создайте переменные для хранения двух чисел.

Выполните над ними сложение, вычитание, умножение, деление и остаток от деления.

Выведите результаты операций в консоль.

Операторы сравнения и логические операторы:

Создайте переменные для хранения двух значений (числа, строки или булевы значения).

Выполните операции сравнения (`==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`) и выведите результаты в консоль.

Используйте логические операторы (`&&`, `||`, `!`) для создания сложных условий. Выведите результаты в консоль.

Практическое задание:

Создайте переменные для хранения ширины и высоты прямоугольника.

Вычислите площадь и периметр прямоугольника.

Выведите результаты в консоль.

Контрольные вопросы:

В чем разница между `var`, `let` и `const`?

Какие типы данных есть в JavaScript?

Как преобразовать переменную из одного типа в другой?

Какие операторы сравнения и логические операторы вы знаете?

Как использовать переменные и операторы для решения простых задач?

Лабораторная работа №2: Условные операторы и циклы

Цель работы:

Научиться использовать условные операторы (`if`, `else if`, `else`, `switch`).

Понять принцип работы циклов (`for`, `while`, `do...while`).

Применять эти конструкции для реализации логики программ.

Решать задачи, требующие повторения действий или ветвления.

Задания для выполнения:

Условные операторы:

Создайте программу, которая проверяет, является ли введенное число положительным, отрицательным или нулем. Используйте `if`, `else if`, `else`.

Создайте программу, которая определяет оценку студента на основе его балла (например, "отлично", "хорошо", "удовлетворительно", "неудовлетворительно"). Используйте `switch`.

Циклы for:

Выведите числа от 1 до 10, используя цикл `for`.

Выведите все четные числа от 20 до 30, используя цикл `for` и условный оператор.

Выведите таблицу умножения на 7 (от 1 до 10) с помощью цикла `for`.

Циклы while и do...while:

Реализуйте программу, которая запрашивает у пользователя ввод числа до тех пор, пока не будет введено положительное число. Используйте цикл while.

Реализуйте программу, которая будет выводить случайные числа от 1 до 10 до тех пор, пока не выпадет число 7. Используйте цикл do...while.

Практическое задание:

Напишите программу, которая принимает от пользователя число и выводит все делители этого числа

Напишите программу которая считает сумму всех чисел от 1 до введенного пользователем числа

Используйте оба вида цикла и сравните результаты

Контрольные вопросы:

Когда следует использовать if, else if и else?

В чем разница между switch и if?

Когда лучше использовать цикл for, а когда while или do...while?

Как прервать выполнение цикла?

Как использовать циклы для решения задач?

Лабораторная работа №3: Функции

Цель работы:

Понять концепцию функций и их роль в программировании.

Научиться объявлять и вызывать функции.

Изучить передачу аргументов в функции и возврат значений.

Применять функции для разделения кода на логические блоки и его повторного использования.

Познакомиться с рекурсивными функциями.

Задания для выполнения:

Объявление и вызов функций:

Создайте функцию, которая принимает имя пользователя в качестве аргумента и выводит приветствие в консоль.

Вызовите эту функцию несколько раз с разными именами.

Функции с возвратом значений:

Создайте функцию, которая принимает два числа в качестве аргументов и возвращает их сумму.

Создайте функцию, которая принимает число в качестве аргумента и возвращает его квадрат.

Выведите результаты работы этих функций в консоль.

Рекурсивные функции:

Создайте рекурсивную функцию для вычисления факториала числа.

Проверьте её работу, вызвав для различных чисел.

Функции как аргументы:

Создайте функцию, которая принимает массив чисел и функцию-колбэк.

Используйте колбэк для обработки каждого элемента массива.

Примените ее для вывода каждого элемента массива в консоль и для нахождения суммы элементов.

Практическое задание:

Создайте функцию, которая принимает строку и возвращает ее в обратном порядке

Создайте функцию, которая вычисляет площадь круга, используя функции из предыдущих заданий

Протестируйте эти функции

Контрольные вопросы:

Что такое функция и для чего она нужна?

Как объявить и вызвать функцию?

Как передать аргументы в функцию?

Как вернуть значение из функции?

Что такое рекурсивная функция?

Лабораторная работа №4: Объекты

Цель работы:

Понять, что такое объекты в JavaScript и как они используются для хранения данных.

Научиться создавать объекты и добавлять в них свойства.

Изучить разные способы доступа к свойствам объекта.

Познакомиться с методами объекта и ключевым словом `this`.

Работать с массивами объектов.

Задания для выполнения:

Создание и свойства объектов:

Создайте объект, представляющий пользователя (с полями `name`, `age`, `email`).

Выведите все свойства объекта в консоль.

Добавьте новое свойство в объект (например, `city`) и выведите измененный объект.

Методы объекта:

Добавьте в объект метод, который будет выводить приветствие, используя имя пользователя.

Добавьте метод, который будет возвращать возраст пользователя.

Вызовите эти методы у созданного объекта.

Доступ к свойствам объекта:

Попробуйте получить доступ к свойствам как через `.`, так и через `[]`

Создайте массив объектов пользователей и выведите данные каждого пользователя в консоль.

this:

В методах объекта выведите `this` и посмотрите что он значит

Добавьте новые методы, использующие `this`.

Практическое задание:

Создайте объект, представляющий книгу с полями title, author, pages.

Создайте метод, который будет выводить информацию о книге в формате "Название: ..., Автор: ..., страниц: ..."

Создайте массив книг и выведите информацию о каждой книге.

Контрольные вопросы:

Что такое объект и зачем он нужен?

Как создать объект в JavaScript?

Как добавить, изменить и удалить свойства объекта?

Что такое метод объекта и как его вызвать?

Что означает this в методе объекта?

Лабораторная работа №5: Массивы и методы массивов

Цель работы:

Понять, что такое массивы в JavaScript и как они используются для хранения коллекций данных.

Научиться создавать массивы и добавлять в них элементы.

Изучить методы массивов (push, pop, shift, unshift, splice, slice, concat, join, indexOf, includes, forEach, map, filter, reduce, sort).

Применять эти методы для обработки и преобразования массивов.

Задания для выполнения:

Создание массивов и добавление элементов:

Создайте массив чисел и массив строк.

Добавьте элементы в конец и в начало массива, используя push и unshift.

Удалите элементы с конца и начала массива, используя pop и shift.

Методы splice и slice:

Используйте метод splice для добавления, удаления и замены элементов в массиве.

Используйте метод slice для создания копии части массива.

Методы concat, join, indexOf, includes:

Объедините два массива в один, используя метод concat.

Преобразуйте массив в строку, используя метод join.

Найдите индекс элемента в массиве с помощью indexOf и проверьте наличие элемента с помощью includes.

Методы forEach, map, filter, reduce:

Выведите все элементы массива с помощью forEach.

Создайте новый массив, умножив каждый элемент исходного массива на 2, с помощью map.

Создайте новый массив, содержащий только четные числа из исходного массива, с помощью filter.

Вычислите сумму всех элементов массива, используя reduce.

Метод sort:

Отсортируйте массив чисел по возрастанию и по убыванию

Отсортируйте массив строк в алфавитном порядке.

Практическое задание:

Создайте массив объектов книг

Выведите названия книг, у которых количество страниц больше 300

Создайте массив названий всех книг

Отсортируйте массив книг по количеству страниц

Контрольные вопросы:

Что такое массив и зачем он нужен?

Как создать массив в JavaScript?

Какие методы массивов вы знаете и как их использовать?

В чем разница между forEach, map, filter и reduce?

Как использовать массивы для обработки коллекций данных?

Фреймворки и библиотеки

Лабораторная работа №1: Интеграция React в HTML-страницу и создание простых компонентов

Цель работы:

Получить практические навыки добавления React-компонентов в существующую HTML-страницу.

Научиться создавать и рендерить базовые React-компоненты.

Познакомиться с основами работы с DOM через React.

Практическое применение стилизации через Bootstrap.

Понять принцип разделения контента на компоненты.

Задания для выполнения:

Подготовка HTML-страницы:

Создайте новую HTML-страницу (например, index.html) или откройте существующую, если таковая имеется.

Внутри тега <body> добавьте пустой <div> с уникальным id, где вы планируете разместить React-компоненты. Например:

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Моя страница с React</title>
```

```
  <link                                rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.
    css">
```

```
</head>
```

```
<body>
```

```

<header class="bg-light py-3">
  <div class="container">
    <h1>Мой Блог</h1>
  </div>
</header>
<main class="container mt-4">
  <div id="root"></div>
</main>
<footer class="bg-light py-2 mt-4">
  <div class="container text-center">
    <p>© 2023 My Blog</p>
  </div>
</footer>

<!-- Загрузка React -->
<script      src="https://unpkg.com/react@16/umd/react.development.js"
crossorigin></script>
<script      src="https://unpkg.com/react-dom@16/umd/react-
dom.development.js" crossorigin></script>
<script src="like_button.js"></script>
</body>
</html>

```

content_copydownload

Use code [with caution](#).Html

Подключение React и Bootstrap:

Подключение React: Перед закрывающим тегом </body> добавьте три тега <script> для подключения библиотек React и React DOM. Обратите внимание, что в продакшене нужно использовать production.min.js вместо development.js:

```

<!-- Загрузка React -->
<script      src="https://unpkg.com/react@16/umd/react.development.js"
crossorigin></script>
<script      src="https://unpkg.com/react-dom@16/umd/react-
dom.development.js" crossorigin></script>

<!-- Загрузка нашего компонента React -->
<script src="like_button.js"></script>

```

content_copydownload

Use code [with caution](#).Html

Подключение Bootstrap:

Внутри тега <head> добавьте ссылку на CDN-версию Bootstrap CSS:

```

<link      rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

```

content_copydownload

Use code [with caution](#).Html

Создание файла `like_button.js` и начального кода:

Создайте файл с именем `like_button.js` в той же папке, что и ваш `index.html`.

Вставьте в него следующий код:

```
'use strict';
```

```
const e = React.createElement;
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked: false };
  }

  render() {
    if (this.state.liked) {
      return 'Вы поставили лайк.';
    }

    return e(
      'button',
      { onClick: () => this.setState({ liked: true }) },
      'Лайк'
    );
  }
}
```

```
const domContainer = document.querySelector('#root');
```

```
ReactDOM.render(e(LikeButton), domContainer);
```

content_copydownload

Use code [with caution](#).JavaScript

Добавление нескольких кнопок и данных:

Измените `like_button.js`, чтобы создать несколько кнопок и передать им данные через `props`:

```
'use strict';
```

```
const e = React.createElement;
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked: false };
  }
}
```

```

render() {
  if (this.state.liked) {
    return `Вы поставили лайк посту ${this.props.postId}`;
  }

  return e(
    'button',
    { onClick: () => this.setState({ liked: true }), className: "btn btn-
    primary" },
    `Лайк посту ${this.props.postId}`
  );
}
}

```

```

class Post extends React.Component {
  render() {
    return e('div', { className: 'card mb-3' },
      e('div', { className: 'card-body' },
        e('h5', { className: 'card-title' }, this.props.title),
        e('p', { className: 'card-text' }, this.props.text),
        e(LikeButton, { postId: this.props.id })
      )
    );
  }
}

```

```

class App extends React.Component {
  render() {
    return e('div', { },
      e(Post, {id: 1, title: "Заголовок первого поста", text: "Текст
      первого поста"}),
      e(Post, {id: 2, title: "Заголовок второго поста", text: "Текст
      второго поста"}),
      e(Post, {id: 3, title: "Заголовок третьего поста", text: "Текст
      третьего поста"})
    );
  }
}

```

```

const domContainer = document.querySelector('#root');
ReactDOM.render(e(App), domContainer);

```

content_copydownload

Use code [with caution](#).JavaScript

Стилизация и контент (социальная сеть):

Включен Bootstrap для стилизации кнопок и основного контента.

Добавлен компонент Post для представления поста.

Создан компонент App, который рендерит все посты.

Вывод всего контента с помощью App:

Все компоненты отрисовываются через App.

Загрузка на GitHub:

Создайте репозиторий на GitHub с именем Student.

Создайте в нем папку по формату Фамилия_1 (латиницей).

Загрузите туда файлы index.html и like_button.js.

Методические указания:

Убедитесь, что HTML-файл содержит правильный id для контейнера, в котором React должен рендерить контент.

Используйте `querySelector()` для получения элемента `domContainer` в `like_button.js`.

Изучите документацию React для более глубокого понимания работы компонентов, состояний и props: <https://reactjs.org/docs/getting-started.html>

Для более глубокого изучения bootstrap: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

Контрольные вопросы:

1. С какой целью создавался React?
2. Что такое компоненты в React?
3. Чем отличаются props и state?
4. Как React работает с DOM?
5. Зачем нужен Bootstrap?

Дополнительные задачи (повышенный уровень сложности):

1. Добавьте возможность отображать количество лайков на каждом посте.
2. Создайте компонент для ввода комментария к посту.
3. Используйте API для загрузки постов и комментариев.
4. Изучите использование jsx для более читаемого кода.

Лабораторная работа №2: JSX, ключи и метод render в React

Цель работы:

Понять, что такое JSX и как он используется для описания UI в React.

Научиться использовать ключи (keys) для эффективного рендеринга списков.

Углубить понимание метода render в компонентах React.

Применять полученные знания на практике при создании списков и компонентов.

Задания для выполнения:

Знакомство с JSX:

Начните с существующего файла `like_button.js` из предыдущей лабораторной.

Замените вызов `React.createElement` (например `e('div', {}, ...)`) на эквивалентный JSX-синтаксис.

// Было:

```
// return e('div', {className: 'card-body'},  
//       e('h5', { className: 'card-title'}, this.props.title),
```

// Стало:

```
    return (  
      <div className="card-body">  
        <h5 className="card-title">{this.props.title}</h5>  
      ...  
    </div>  
    );
```

[content_copydownload](#)

Use code [with caution](#).JavaScript

Перепишите код компонента `Post` и `App`, используя JSX-синтаксис. Убедитесь, что все работает так же, как и раньше.

Работа со списками и ключами:

Создайте компонент `PostList`, который будет получать массив объектов постов в качестве `props`.

Используйте метод `map` для итерации по массиву постов и рендеринга каждого поста в виде компонента `Post`.

Добавьте атрибут `key` каждому отрендеренному компоненту `Post`. Убедитесь, что `key` является уникальным идентификатором для каждого поста (например, `post.id`).

```
class PostList extends React.Component {  
  render() {  
    const posts = this.props.posts.map(post => (  
      <Post key={post.id} id={post.id} title={post.title} text={post.text}  
    />  
    ));  
    return <div>{posts}</div>;  
  }  
}
```

[content_copydownload](#)

Use code [with caution](#).JavaScript

Измените компонент `App`, чтобы он отображал компонент `PostList` и передавал ему массив постов:

```
class App extends React.Component {  
  render() {  
    const posts = [  

```

```

        {id: 1, title: "Заголовок первого поста", text: "Текст первого поста"},
        {id: 2, title: "Заголовок второго поста", text: "Текст второго поста"},
        {id: 3, title: "Заголовок третьего поста", text: "Текст третьего поста"}
    ]

    return (
    <div>
        <PostList posts={posts} />
    </div>
    );
}
}

```

content_copydownload

Use code [with caution](#).JavaScript

Углубленное изучение метода render:

Добавьте в компонент LikeButton логику для изменения состояния компонента, чтобы показать, что рендеринг происходит повторно при изменении состояния.

Добавьте в компонент Post метод shouldComponentUpdate, чтобы предотвратить повторный рендеринг, если props не изменились.

```

class Post extends React.Component {
    shouldComponentUpdate(nextProps, nextState) {
        if(this.props.title !== nextProps.title || this.props.text !==
nextProps.text)
        {
            return true
        }
        return false
    }

    render() {
        console.log('render Post ' + this.props.id)
        return (
        <div className="card mb-3">
            <div className="card-body">
                <h5 className="card-title">{this.props.title}</h5>
                <p className="card-text">{this.props.text}</p>
                <LikeButton postId={this.props.id} />
            </div>
        </div>
        );
    }
}

```

```
}  
}  
content_copydownload  
Use code with caution.JavaScript
```

Выведите в консоль сообщение перед каждым рендерингом каждого компонента LikeButton и Post , чтобы было видно, когда они перерендериваются.

Зафиксируйте в отчете, как ведет себя приложение при изменении состояния.

Стилизация:

Примените классы Bootstrap для дополнительной стилизации компонентов списков.

Обновление GitHub:

Загрузите обновленный файл like_button.js в тот же репозиторий, в папку Фамилия_1.

Методические указания:

Используйте онлайн-конвертеры JSX для понимания, как JSX трансформируется в JavaScript (например, <https://babeljs.io/repl>).

Особое внимание уделите правильному использованию ключей (key) при рендеринге списков.

Изучите документацию React по методу render и shouldComponentUpdate.

Поэкспериментируйте с различными вариантами JSX-синтаксиса.

Контрольные вопросы:

1. Что такое JSX и зачем он нужен?
2. Почему использование key важно при рендеринге списков?
3. Какую роль играет метод render в компоненте React?
4. В каких случаях может понадобиться метод shouldComponentUpdate?
5. Как можно передавать данные через JSX?
6. В чем преимущество JSX перед React.createElement?

Дополнительные задачи (повышенный уровень сложности):

1. Добавьте возможность редактировать посты.
2. Реализуйте поиск по списку постов.
3. Используйте useEffect для отслеживания изменений в данных.
4. Попробуйте использовать фреймворки для работы с css, например styled components.

Лабораторная работа №3: Способы описания компонентов с реквизитами (props) и реструктурирование

Цель работы:

Углубить понимание передачи данных через props в React.

Изучить различные способы передачи данных (строки, числа, объекты, функции, JSX).

Научиться использовать деструктуризацию (реструктуризацию) props для удобства.

Понимать, как и когда использовать defaultProps и propTypes.

Задания для выполнения:

Ход работы:

Подготовка среды:

Убедитесь, что у вас есть рабочая среда React (можно использовать create-react-app или существующий проект).

Создайте новый компонент UserCard.js в папке src/components (или аналогичной).

Создание компонента UserCard с разными типами props:

Откройте файл UserCard.js.

Начните с создания базовой структуры компонента (функционального компонента):

```
import React from 'react';
```

```
function UserCard(props) {  
  return (  
    <div className="user-card">  
      {/* Здесь будет отображение props */}  
    </div>  
  );  
}
```

```
export default UserCard;
```

```
content_copydownload
```

Use code [with caution](#).JavaScript

Добавьте отображение props внутри div:

```
function UserCard(props) {  
  return (  
    <div className="user-card">  
      <h2>Имя: {props.name}</h2>  
      <p>Возраст: {props.age}</p>  
      <p>Активен: {props.isActive ? 'Да' : 'Нет'}</p>  
      <div>  
        <p>Аватар: <img src={props.profile.avatarUrl} alt="avatar" /></p>  
        <p>Биография: {props.profile.bio}</p>  
      </div>  
      <button onClick={props.onClick}> Нажать </button>  
      <div>  
        {props.children}  
      </div>  
    </div>  
  );  
}
```

content_copydownload

Use code [with caution](#).JavaScript

Обратите внимание, как отображаются строки, числа, булевы значения, объекты и JSX.

Деструктуризация props:

Измените сигнатуру функции, чтобы использовать деструктуризацию props:

```
function UserCard({ name, age, isActive, profile, onClick, children }) {  
  // ...  
}
```

content_copydownload

Use code [with caution](#).JavaScript

Внутри компонента можно использовать name, age, isActive и т.д. напрямую, без props..

Отредактируйте компонент и убедитесь, что все работает как и раньше.

defaultProps:

Добавьте свойства по умолчанию после компонента:

```
UserCard.defaultProps = {  
  age: 18,  
  isActive: false,  
  profile: {},  
};
```

content_copydownload

Use code [with caution](#).JavaScript

Теперь, если не передать age, isActive или profile, будут использоваться значения по умолчанию.

Попробуйте удалить эти props из App и убедитесь, что значения подставляются по умолчанию.

propTypes:

Установите propTypes после defaultProps:

```
import PropTypes from 'prop-types';
```

```
UserCard.propTypes = {  
  name: PropTypes.string.isRequired,  
  age: PropTypes.number,  
  isActive: PropTypes.bool,  
  profile: PropTypes.shape({  
    avatarUrl: PropTypes.string,  
    bio: PropTypes.string,  
  }),  
  onClick: PropTypes.func,  
  children: PropTypes.node,  
};
```

content_copydownload

Use code [with caution](#).JavaScript

Теперь React будет выдавать предупреждения в консоли, если переданы props неверного типа или нет props обязательного типа.

Попробуйте передать age в виде строки, чтобы увидеть предупреждение.

Примеры использования в App.js:

Импортируйте компонент UserCard в App.js:

```
import UserCard from './components/UserCard';
```

```
content_copydownload
```

Use code [with caution](#).JavaScript

Используйте компонент несколько раз, передавая различные props:

```
function App() {  
  return (  
    <div>  
      <UserCard  
        name="John Doe"  
        age={30}  
        isActive={true}  
        profile={{avatarUrl: "https://via.placeholder.com/150", bio: "Some  
        bio"}}}  
        onClick={() => console.log("clicked")}  
      >  
        
    </UserCard>  
    <UserCard  
      name="Jane Smith"  
      profile={{avatarUrl: "https://via.placeholder.com/150", bio: "Some  
      other bio"}}  
      onClick={() => alert("clicked too!")}  
    />  
    <UserCard  
      name="Test user"  
      onClick={() => alert("clicked three!")}  
    />  
    </div>  
  );  
}
```

```
content_copydownload
```

Use code [with caution](#).JavaScript

Попробуйте передавать разные типы данных в props, добавлять контент.

Запустите приложение и убедитесь, что все работает правильно.

Откройте консоль разработчика и посмотрите, какие предупреждения выводятся.

Отчет:

Зафиксируйте, как изменяется отображение компонента при изменении props.

Отметьте, как работает деструктуризация.

Опишите, как и когда стоит использовать `defaultProps` и `propTypes`.

Примеры использования

Создайте компонент `App`, который будет отображать несколько `UserCard`, передавая в них разные props и контент.

Покажите пример использования функции в props.

Методические указания:

Изучите документацию React по props, `defaultProps` и `propTypes`.

Поэкспериментируйте с различными способами передачи данных.

Обратите внимание на удобство деструктуризации.

Используйте `propTypes` для проверки правильности передачи данных.

Контрольные вопросы:

Какие типы данных можно передавать через props?

Зачем нужна деструктуризация props?

Как использовать `defaultProps`?

Для чего нужны `propTypes` и как они работают?

Как передать JSX контент через props?

Дополнительные задачи:

Создайте компонент `Button`, который принимает `children`, `onClick` и `variant` (строка).

Реализуйте кастомную проверку `propTypes` (например, для проверки email).

Используйте `spread`-оператор (...) для передачи части props в дочерний компонент.

Лабораторная работа №4: Node.js и условный рендеринг

Цель работы:

Понять, что такое условный рендеринг и как его использовать в React.

Изучить различные способы условного рендеринга (`if/else`, тернарный оператор, логическое `&&`).

Познакомиться с основами работы с Node.js и REST API.

Научиться загружать данные с сервера и отображать их на основе состояния.

Задания для выполнения:

Условный рендеринг:

Создайте компонент `Message`, который отображает разные сообщения в зависимости от значения пропса `type` (например, "success", "warning", "error").

Реализуйте это с помощью `if/else`, тернарного оператора и логического `&&` в разных вариантах.

Покажите пример с использованием `switch case`.

Настройте компоненты для каждого типа сообщения.

Основы Node.js (простой сервер):

Создайте простую Node.js-приложение с использованием express и JSON-файла.

Настройте роут для получения списка постов.

Установите nodemon для автоматического перезапуска сервера.

Загрузка данных с сервера:

В компоненте PostList из предыдущей работы добавьте загрузку данных о постах с вашего Node.js-сервера с помощью fetch (или axios). Используйте useState для хранения загруженных данных и состояния загрузки.

Покажите сообщение о загрузке (например, "Загрузка...") пока данные не будут получены.

Обработайте ошибки, если запрос не удался.

Используйте useEffect для загрузки данных после первого рендеринга.

Условный рендеринг на основе состояния:

Отобразите данные, когда они загрузятся. Если данные не загрузились, покажите сообщение об ошибке или loading.

Покажите в компоненте PostList сообщение "Нет постов", если массив posts пустой.

Методические указания:

Познакомьтесь с основами Node.js и express.

Изучите документацию React по условному рендерингу.

Поэкспериментируйте с разными способами условного рендеринга.

Убедитесь, что ваш сервер работает правильно, и данные успешно загружаются.

Контрольные вопросы:

Для чего нужен условный рендеринг?

Какие есть способы реализации условного рендеринга в React?

Как обрабатывать ошибки при загрузке данных?

Зачем нужен useEffect и как его использовать?

Как создать простой REST API на Node.js?

Дополнительные задачи:

Добавьте пагинацию на сервере и на клиенте.

Реализуйте фильтрацию постов по различным параметрам.

Используйте библиотеки для работы с fetch (например, axios).

Лабораторная работа №5: Props, State и события

Цель работы:

Углубить понимание props и state и их взаимодействия.

Научиться обрабатывать события в React (onClick, onChange, onSubmit и др.)

Управлять состоянием компонента на основе событий.

Создавать интерактивные компоненты.

Задания для выполнения:

Компонент Counter:

Создайте компонент Counter с кнопками "+" и "-".

Используйте useState для хранения текущего значения счетчика.

Обработайте события клика на кнопки для увеличения и уменьшения счетчика.

Компонент Input:

Создайте компонент Input с полем ввода и кнопкой "Сохранить".

Используйте useState для хранения значения поля ввода.

Обработайте событие onChange в поле ввода для обновления состояния.

Обработайте событие onClick кнопки "Сохранить", чтобы вывести введенное значение в консоль.

Управление state родительского компонента:

Измените компонент App, чтобы он хранил состояние.

Создайте компонент ControlPanel, который будет принимать функцию для обновления state в App.

Используйте эту функцию в ControlPanel для изменения состояния родительского компонента.

В App отобразите состояние родителя.

Формы:

Создайте форму для добавления новых постов.

Используйте controlled components.

Добавьте валидацию форм.

Методические указания:

Изучите документацию React по props, state и событиям.

Поэкспериментируйте с различными типами событий и их обработкой.

Обратите внимание на разницу между props и state.

Попробуйте создать более сложные интерактивные компоненты.

Контрольные вопросы:

Чем отличаются props и state?

Как обрабатывать события в React?

Как управлять state компонента при помощи событий?

Что такое "controlled components" и для чего они нужны?

Как передавать функции как props?

Дополнительные задачи:

Создайте компонент Modal, который будет отображаться при клике на кнопку.

Реализуйте Drag and Drop для перемещения элементов в списке.

Создайте компонент-слайдер.

Цель работы:

Понять, как создавать многостраничные приложения в React.

Познакомиться с React Router для навигации между страницами.

Научиться использовать массивы в state и управлять ими.

Создавать списки, формы и интерактивные элементы.

Задания для выполнения:

Установка и настройка React Router:

Установите react-router-dom.

Настройте BrowserRouter в корне приложения.

Создайте несколько компонентов (например, Home, About, Posts).

Используйте Route для сопоставления путей с компонентами.

Добавьте Link для навигации между страницами.

Массивы в state:

В компоненте Posts храните список постов в state (используя массив объектов).

Отобразите посты на странице.

Реализуйте возможность добавлять, удалять и редактировать посты с использованием state массива.

Используйте map для отображения массива.

Формы:

Создайте форму для добавления новых постов.

Используйте controlled components.

Добавьте валидацию форм.

Работа с ключами (key) в списках:

Убедитесь, что при добавлении, удалении или редактировании постов, React правильно отслеживает изменения.

Методические указания:

Изучите документацию по React Router.

Поймите, как работают массивы в state и как их правильно обновлять.

Изучите разницу между нативным array методами и тем, что возвращает map

Обратите внимание на то, что нужно использовать key в списках.

Контрольные вопросы:

Как работает React Router и для чего он нужен?

Как настроить роутинг в React-приложении?

Как использовать массивы в state и их обновлять?

Для чего нужны ключи (key) в списках?

Как использовать формы в React?

Дополнительные задачи:

Добавьте аутентификацию для доступа к страницам.

Используйте параметры в пути.

Реализуйте поиск постов.

Добавьте модальные окна.

ТЕСТОВЫЕ ЗАДАНИЯ

Тема 1.1 Проектирование и дизайн интерфейсов

Профессиональная(-ые) компетенция(-ии):

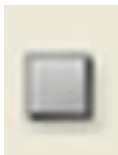
ПК.8.1 Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика

Задание №1

Выполните тестовые задания *"Графический редактор Adobe Photoshop"*

1. Какой инструмент "переносит" цвет пиксела из изображения на цветовые поля палитры инструментов?
 - a) штамп;
 - b) кисть;
 - c) прямоугольник;
 - d) пипетка.
2. Что такое кадрирование изображения?
 - a) изменение контрастности изображения;
 - b) отсечение частей изображения с целью фокусирования или улучшения компоновки;
 - c) улучшение качества изображения;
 - d) изменение цветовой гаммы.
3. Инструмент, позволяющий плавно переходить от одного цвета к другому – это...
 - a) губка;
 - b) штамп;
 - c) кисть;
 - d) градиент.
4. Какой из заголовков верхнего меню нужно выбрать, чтобы изменить размеры холста?
 - a) файл;
 - b) редактирование;
 - c) изображение;
 - d) слой.
5. Какого инструмента нет в фотошопе?
 - a) лассо;
 - b) круглое лассо;
 - c) многоугольное лассо;
 - d) магнитное лассо.
6. Какой из этих режимов НЕ является настройками ластика?
 - a) обычный;
 - b) кисть;
 - c) карандаш;
 - d) блок.

7. Какой режим выделения необходимо включить, чтобы выделить лампочки, как показано на рисунке?



8. При работе с операцией Трансформирование, чтобы отменить эту операцию необходимо нажать



9. Гистограмма — это ...

- а) диаграмма специального вида, который показывает распределение цвета от белого к черному;
- б) график специального вида, который показывает распределение пикселей по яркости, от белого к черному.

10. Особенность цветовой коррекции заключается в ...

- а) изменение одного цвета влечет за собой изменение других цветов изображения;

- б) настройки баланса цветов;
- с) изменении конкретного цвета.

<i>Оценк а</i>	<i>Показатели оценки</i>
3	Выполнены 3-6 заданий.
4	Выполнены 7-8 заданий.
5	Выполнены 9-10 заданий.

Задание №2 (из текущего контроля)

Ответить на вопросы:

Сопоставьте цветовые сочетания и экспрессивные или дисгармоничные цвета, получающиеся при их смешении.

Как выглядит формула Гете активности цвета в зависимости от его яркости? Как изменяется цветовая гармония в зависимости от изменения насыщенности и светлоты исходных цветов?

<i>Оценк а</i>	<i>Показатели оценки</i>
3	Дан ответ на один вопрос .
4	Даны ответы на 2 вопроса из 3.
5	Даны ответы на все 3 вопроса.

Дидактическая единица для контроля:

современные тенденции дизайна

Профессиональная(-ые) компетенция(-ии):

ПК.8.1 Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика

Задание №1

Ответить на вопросы:

1. Аппаратные средства мультимедиа технологии.
2. Программные средства мультимедиа технологии.
3. Классификация мультимедиа-технологий.

<i>Оценк а</i>	<i>Показатели оценки</i>
3	Дан ответ на 1 вопрос из 3.
4	Дан ответ на 2 вопроса из 3.
5	Даны ответы на все вопросы.

Задание №2 (из текущего контроля)

Приведите пример не менее 8 современных трендов web дизайна.

Оценк а	Показатели оценки
3	Привиден пример 4 трендов.
4	Приведен пример 6 трендов.
5	Приведен пример 8 трендов.

Дидактическая единица для контроля:

инструменты для разработки эскизов, схем интерфейсов и прототипа дизайна веб-приложений

Профессиональная(-ые) компетенция(-ии):

ПК.8.1 Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика

Задание №1

Установите соответствие между видами компьютерной графики (КГ) и их назначением.

Вид КГ	Назначение
1. Научная графика	а) используется в работе инженеров, архитекторов, изобретателей новой техники; является обязательным элементом САПР; представлена как плоскими изображениями (проекция, сечения), так и пространственными трёхмерными изображениями.
2. Деловая графика	б) произвольное рисование и черчение на экране компьютера с помощью графических редакторов.
3. Конструкторская графика	в) это получение движущихся изображений на экране дисплея; рисунки, выводимые последовательно на экран с определённой частотой, создают иллюзию движения.
4. Иллюстративная графика	г) наглядное представление различных показателей работы учреждения: плановые показатели, отчетная документация, статистические сводки.
5. Художественная и рекламная графика	д) объединение высококачественного изображения на экране компьютера со звуковым сопровождением; наибольшее распространение получили в области обучения, рекламы, развлечений.
6. Компьютерная анимация	е) визуализация объектов научных исследований, графическая обработка расчетов, проведение вычислительных экспериментов с наглядным представлением их результатов.
7. Мультимедиа	ж) Создание реалистических изображений и «движущихся картинок», стала популярной во многом благодаря телевидению; графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти.

Оценк а	Показатели оценки
3	Установлено 3 - 4 соответствия.
4	Установлено 5-6 соответствий.
5	Установлены все соответствия

Задание №2 (из текущего контроля)

Ответить на вопросы теста:

1. Использование цвета для оформления текста...

- а) только стандартные 16 цветов;
- б) 48 цветов палитры редактора;
- в) любые;
- г) только черный.

2. Использоваться на странице могут дополнительные элементы оформления:

а
)
з
в
у
к
и
;
б
)
в
и
д
е
о
;

- в) бегущие строки;
- г) все вышеперечисленные.

3. Изображения, вставляемые на страницу:

- а) переводятся в двоичную форму и помещаются в HTML-код;
- б) записываются в архив и прилагаются к HTML-файлу;
- в) изображения не сохраняются, а при просмотре используются из библиотеки пользователя;
- г) сохраняются как отдельные файлы, а в HTML-код вставляется только ссылка на них.

4. Графика, представляемая в памяти компьютера в виде совокупности точек, называется:

а)
растро
вой; б)
вектор
ной; в)
трехмер
ной;
г)
фрактал
ной.

5. Элементарным объектом растровой графики является следующий:

- а) рисуется одним инструментом;
- б) пиксель;
- в) символ;
- г) примитив.

6. Инструмент, позволяющий залить изображение двумя, плавно

перетекающими друг в друга цветами, называется:

- а) банка
- краски;
- б)
- заливка
- ;
- в)
- гра
- ди
- ент
- ;г)
- узо
- р.

<i>Оценк а</i>	<i>Показатели оценки</i>
3	Даны ответы на 4 из 6 вопросов правильно.
4	Даны ответы на 5 из 6 вопросов правильно.
5	Даны ответы на все вопросы правильно.

Раздел 2 Верстка страниц

Тест по верстке сайта (1 вариант)

Инструкция: Выберите один правильный ответ из предложенных вариантов.

1. Какой тег используется для создания заголовка первого уровня в HTML?
 - а) <heading>
 - б) <h1/>
 - в) <title>
 - г) <header>
2. Какой тег используется для создания абзаца?
 - а) <text>
 - б) <paragraph>
 - в) <p>
 - г) <par>
3. Какой тег используется для создания ссылки?
 - а) <link>
 - б) <href>
 - в) <url>
 - г) <a>
4. Какой тег используется для вставки изображения?
 - а) <image>
 - б) <src>
 - в)
 - г) <picture>

5. Какой атрибут используется для указания адреса ссылки?
- a) src
 - b) url
 - c) href
 - d) link
6. Какой тег используется для создания нумерованного списка?
- a)
 - b) <list>
 - c)
 - d)
7. Какой тег используется для создания нenumерованного списка?
- a)
 - b)
 - c) <list>
 - d)
8. Какой тег используется для создания элемента списка?
- a) <list>
 - b)
 - c)
 - d)
9. Какой атрибут используется для указания пути к изображению?
- a) href
 - b) url
 - c) src
 - d) link
10. Какой тег используется для создания таблицы?
- a) <table_box>
 - b) <table>
 - c) <table>
 - d) <grid>
11. Какой тег используется для создания строки в таблице?
- a) <td>
 - b) <tr>
 - c) <th>
 - d) <row>
12. Какой тег используется для создания ячейки в таблице?
- a) <tr>
 - b) <th>
 - c) <td>
 - d) <cell>
13. Какой тег используется для создания заголовка столбца в таблице?
- a) <td>
 - b) <th>
 - c) <tr>
 - d) <header_cell>
14. Что такое CSS?

- a) Язык программирования
 - b) Язык разметки
 - c) Язык стилей
 - d) База данных
15. Как добавить CSS-стили в HTML-документ?
- a) Через атрибут style
 - b) Через тег <style>
 - c) Через внешний файл
 - d) Все вышеперечисленные варианты
16. Как выбрать элемент с классом "my-class" в CSS?
- a) #my-class
 - b) .my-class
 - c) my-class
 - d) class-my-class
17. Как выбрать элемент с id "my-id" в CSS?
- a) .my-id
 - b) my-id
 - c) #my-id
 - d) id-my-id
18. Какое свойство CSS используется для изменения цвета текста?
- a) background-color
 - b) color
 - c) text-color
 - d) font-color
19. Какое свойство CSS используется для изменения размера шрифта?
- a) font-size
 - b) text-size
 - c) font-height
 - d) size
20. Какое свойство CSS используется для задания отступов вокруг элемента?
- a) padding
 - b) margin
 - c) border
 - d) spacing
21. Какое свойство CSS используется для задания внешних отступов вокруг элемента?
- a) padding
 - b) margin
 - c) border
 - d) spacing
22. Какое свойство CSS используется для задания границ элемента?
- a) padding
 - b) margin
 - c) border
 - d) outline

23. Что такое адаптивная верстка?
- a) Верстка для больших экранов
 - b) Верстка для маленьких экранов
 - c) Верстка, которая адаптируется под разные размеры экранов
 - d) Верстка только для мобильных устройств
24. Какой CSS-метод используется для создания адаптивного макета?
- a) flexbox
 - b) grid
 - c) float
 - d) Все вышеперечисленные варианты
25. Что такое медиа-запросы в CSS?
- a) Запросы для работы с сервером
 - b) Запросы к базе данных
 - c) Условия для применения стилей в зависимости от характеристик устройства
 - d) Запросы для работы с API
26. Какое свойство CSS используется для создания сетки?
- a) flex-grid
 - b) layout
 - c) grid
 - d) template
27. Какое свойство CSS используется для задания ширины элемента?
- a) size
 - b) width
 - c) height
 - d) size-width
28. Какое свойство CSS используется для задания высоты элемента?
- a) size
 - b) width
 - c) height
 - d) size-height
29. Какой псевдокласс используется для добавления стиля при наведении курсора на элемент?
- a) :hover
 - b) :focus
 - c) :active
 - d) :visited
30. Какой HTML-тег используется для добавления метаданных к документу?
- a) <meta>
 - b) <head>
 - c) <header>
 - d) <information>

Ключи ответов:

1. b

- 2. c
- 3. d
- 4. c
- 5. c
- 6. c
- 7. b
- 8. d
- 9. c
- 10. b
- 11. b
- 12. c
- 13. b
- 14. c
- 15. d
- 16. b
- 17. c
- 18. b
- 19. a
- 20. a
- 21. b
- 22. c
- 23. c
- 24. d
- 25. c
- 26. c
- 27. b
- 28. c
- 29. a
- 30. a

Раздел 3 Разработка клиентской части информационных ресурсов

Тест по JavaScript (2 варианта)

Инструкция: Выберите один правильный ответ из предложенных вариантов.

Вариант 1

1. Какой оператор используется для сравнения двух значений на строгое равенство (с учетом типа данных)?
 - a) =
 - b) ==
 - c) ===
 - d) !=
2. Какой тип данных в JavaScript используется для представления логических значений (истина или ложь)?
 - a) number
 - b) string

- c) boolean
 - d) null
- 3. Какой цикл используется для выполнения блока кода определенное количество раз?
 - a) while
 - b) do...while
 - c) for
 - d) if
- 4. Какое ключевое слово используется для объявления переменной, значение которой нельзя будет изменить после присваивания?
 - a) var
 - b) let
 - c) const
 - d) new
- 5. Что выведет в консоль следующий код: `console.log(typeof null);`?
 - a) "null"
 - b) "object"
 - c) "undefined"
 - d) "number"
- 6. Как вызвать функцию `myFunction`?
 - a) `call myFunction`
 - b) `run myFunction`
 - c) `myFunction()`
 - d) `new myFunction`
- 7. Что такое рекурсивная функция?
 - a) Функция, которая выполняется один раз
 - b) Функция, которая вызывает сама себя
 - c) Функция, которая никогда не заканчивает выполнение
 - d) Функция которая возвращает массив
- 8. Как добавить новый элемент в конец массива `myArray`?
 - a) `myArray.add(element)`
 - b) `myArray.insert(element)`
 - c) `myArray.push(element)`
 - d) `myArray[element]`
- 9. Какой метод массива используется для создания нового массива, который состоит только из элементов, удовлетворяющих условию?
 - a) `map`
 - b) `forEach`
 - c) `filter`
 - d) `reduce`
- 10. Как получить доступ к свойству `name` объекта `user`?
 - a) `user(name)`
 - b) `user[name]`
 - c) `user.name`
 - d) `user->name`
- 11. Что делает метод `join()` массива?
 - a) Складывает элементы массива

- b) Разворачивает массив
 - c) Преобразует массив в строку
 - d) Сортирует массив
- 12. Что выведет этот код `console.log(2 + '2')`?
 - a) 4
 - b) "4"
 - c) "22"
 - d) Error
- 13. Что делает `splice` метод?
 - a) Возвращает часть массива
 - b) Добавляет, удаляет и заменяет элементы массива
 - c) Копирует массив
 - d) Сортирует массив
- 14. Какой метод массива используется для перебора элементов?
 - a) `sort`
 - b) `push`
 - c) `map`
 - d) `forEach`
- 15. Какое ключевое слово указывает на текущий объект в методах объекта?
 - a) `this`
 - b) `that`
 - c) `it`
 - d) `self`

Вариант 2

1. Какой оператор используется для присваивания значения переменной?
 - a) `==`
 - b) `===`
 - c) `=`
 - d) `!=`
2. Какой тип данных в JavaScript используется для представления текстовых значений?
 - a) `number`
 - b) `string`
 - c) `boolean`
 - d) `null`
3. Какой цикл выполняется хотя бы один раз, даже если условие ложно?
 - a) `while`
 - b) `do...while`
 - c) `for`
 - d) `if`
4. Какое ключевое слово используется для объявления переменной, значение которой может быть изменено?
 - a) `const`
 - b) `let`
 - c) `new`
 - d) `var`

5. Что выведет в консоль следующий код: `console.log(typeof undefined);`?
- a) "null"
 - b) "object"
 - c) "undefined"
 - d) "number"
6. Как передать аргумент `value` в функцию `myFunction` при вызове?
- a) `myFunction(value)`
 - b) `myFunction = value`
 - c) `myFunction as value`
 - d) `value myFunction`
7. Что такое колбэк функция?
- a) Функция, которую возвращает другая функция
 - b) Функция которая вызывает сама себя
 - c) Функция, которая передается в качестве аргумента другой функции
 - d) Функция которая возвращает объект
8. Какой метод массива используется для удаления последнего элемента?
- a) `myArray.delete()`
 - b) `myArray.pop()`
 - c) `myArray.remove()`
 - d) `myArray.splice()`
9. Какой метод массива используется для преобразования каждого элемента массива и создания нового массива?
- a) `map`
 - b) `forEach`
 - c) `filter`
 - d) `reduce`
10. Как получить значение свойства `email` у объекта `user` через `[]`?
- a) `user.email`
 - b) `user["email"]`
 - c) `user(email)`
 - d) `user->email`
11. Что делает метод `concat()` массива?
- a) Объединяет массивы
 - b) Разделяет массив
 - c) Удаляет элементы массива
 - d) Сортирует массив
12. Что выведет этот код `console.log(2 * '2')`?
- a) 4
 - b) "4"
 - c) "22"
 - d) Error
13. Что делает `slice` метод?
- a) Добавляет элемент массива
 - b) Удаляет элементы массива
 - c) Возвращает часть массива

- d) Изменяет размер массива

14. Какой метод массива позволяет сложить все элементы массива и получить 1 значение?

- a) map
- b) forEach
- c) filter
- d) reduce

15. Как вывести значение свойства age в методе объекта user?

- a) this.age
- b) user.age
- c) age
- d) user->age

Ключи ответов

Вариант 1

1. c
2. c
3. c
4. c
5. b
6. c
7. b
8. c
9. c
10. c
11. c
12. c
13. b
14. d
15. a

Вариант 2

1. c
2. b
3. b
4. b
5. c
6. a
7. c
8. b
9. a
10. b
11. a
12. a
13. c
14. d
15. a

Тест по React (2 варианта)

Инструкция: Выберите один правильный ответ из предложенных вариантов.

Вариант 1

1. Какой метод используется для рендеринга React-компонента в DOM?
 - a) ReactDOM.mount()
 - b) ReactDOM.render()
 - c) ReactDOM.update()
 - d) ReactDOM.display()
2. Что такое JSX?
 - a) JavaScript-расширение для написания стилей
 - b) JavaScript-расширение для описания UI
 - c) JavaScript-библиотека для работы с DOM
 - d) JavaScript-библиотека для работы с API
3. Что такое props в React?
 - a) Внутреннее состояние компонента
 - b) Данные, передаваемые в компонент от родителя
 - c) Метод компонента
 - d) Событие компонента
4. Какой хук используется для работы с состоянием компонента в функциональных компонентах?
 - a) useState()
 - b) useEffect()
 - c) useRef()
 - d) useContext()
5. Как передать данные из родительского компонента в дочерний?
 - a) Через state
 - b) Через props
 - c) Через refs
 - d) Через context
6. Какой метод используется для получения доступа к DOM-элементу через React?
 - a) getElement()
 - b) findNode()
 - c) useRef()
 - d) querySelector()
7. Что такое условный рендеринг в React?
 - a) Метод стилизации элементов
 - b) Метод обработки данных в форме
 - c) Отображение элементов в зависимости от условий
 - d) Метод работы с API
8. Какой метод используется для загрузки данных с сервера в React?
 - a) send()
 - b) ajax()
 - c) fetch()
 - d) import()
9. Для чего нужен key при рендеринге списков в React?

- a) Для стилизации элементов
 - b) Для определения уникальности элемента
 - c) Для добавления анимации
 - d) Для добавления обработчиков событий
10. Какой метод React используется для предотвращения повторного рендеринга компонента, если его props не изменились?
- a) `componentDidUpdate()`
 - b) `shouldComponentUpdate()`
 - c) `componentWillUpdate()`
 - d) `render()`
11. В каком порядке выполняются хуки в компоненте React?
- a) `useEffect`, `useState`, `useRef`
 - b) `useState`, `useEffect`, `useRef`
 - c) `useRef`, `useState`, `useEffect`
 - d) Порядок не важен
12. Как отобразить контент внутри React-компонента?
- a) С помощью HTML
 - b) Через `document.body`
 - c) С помощью `render()`
 - d) С помощью CSS
13. Какой тег используется для контейнера, в который будет рендериться React?
- a) `<body>`
 - b) `<app>`
 - c) `<div>`
 - d) `<span>`
14. Что такое компонент в React?
- a) Функция для отрисовки html
 - b) Функция, для получения данных с сервера
 - c) Функция или класс, которые описывают интерфейс
 - d) Функция для работы с css
15. Какой пакет используется для навигации между страницами в React?
- a) `react-navigation`
 - b) `react-routing`
 - c) `react-router-dom`
 - d) `react-pages`
16. Какой метод используется для обработки события клика в React?
- a) `click()`
 - b) `onClick()`
 - c) `onMouseClick()`
 - d) `handleClick()`
17. Зачем нужен `defaultProps` в компоненте React?
- a) Для передачи обязательных пропсов
 - b) Для задания значений по умолчанию для пропсов
 - c) Для стилизации
 - d) Для валидации пропсов

18. Какой хук используется для выполнения побочных эффектов (загрузка данных, подписки)?
- a) useState()
 - b) useRef()
 - c) useEffect()
 - d) useContext()
19. Что выведет в консоль следующий код?
- ```
20. function MyComponent({prop}) {
21. return (
22. <h1>{prop}</h1>
23.);
24. }
25. MyComponent.defaultProps = {
26. prop: 'hello'
27. }
ReactDOM.render(<MyComponent />, document.getElementById('root'));
```
- content\_copydownload
- Use code [with caution](#).JavaScript
- a) undefined
  - b) null
  - c) hello
  - d) Пустая строка
28. Как добавить стили bootstrap к React-компоненту?
- a) Через style
  - b) Через css
  - c) Через классы
  - d) Невозможно

## Вариант 2

1. Какой метод используется для обновления React-компонента в DOM?
- a) ReactDOM.mount()
  - b) ReactDOM.render()
  - c) ReactDOM.update()
  - d) ReactDOM.display()
2. Что такое state в React?
- a) Данные, передаваемые от родителя
  - b) Внутреннее состояние компонента
  - c) Метод компонента
  - d) Событие компонента
3. Какой хук используется для работы с побочными эффектами в функциональных компонентах?
- a) useState()
  - b) useEffect()
  - c) useRef()
  - d) useContext()
4. Как обновить состояние компонента в React?
- a) this.setState()

- b) `this.updateState()`
  - c) `setMyState()`
  - d) `state = new value`
5. Что такое controlled component?
- a) Компонент, который не управляет своим состоянием
  - b) Компонент, состояние которого управляется родителем
  - c) Компонент, который никогда не перерендеривается
  - d) Компонент только для отображения данных
6. Как правильно использовать key при map?
- a) Как index массива
  - b) Как название элемента
  - c) Как уникальный id
  - d) Как строку
7. В каких случаях необходимо использовать условный рендеринг?
- a) Для получения данных с сервера
  - b) Для добавления анимаций
  - c) Для отображения разных UI в зависимости от состояния
  - d) Для стилизации элементов
8. Какой метод используется для отслеживания изменений состояния в React?
- a) `stateChange`
  - b) `onStateChange`
  - c) `useEffect`
  - d) `stateUpdate`
9. Почему важно деструктурировать props?
- a) Для стилизации компонента
  - b) Для краткости кода
  - c) Для обеспечения безопасности
  - d) Для того, чтобы избежать ошибок
10. Какой метод React используется для добавления обработчика события на DOM-элемент?
- a) `addListener`
  - b) `addEventListener`
  - c) `onEvent`
  - d) `on`
11. Для чего нужны ключи в компонентах React?
- a) Для стилизации компонента
  - b) Для того, чтобы избежать ошибок
  - c) Для оптимизации рендеринга
  - d) Для привязки к конкретному dom-элементу
12. Как использовать JSX-код внутри React компонента?
- a) В виде css-классов
  - b) В виде обычного html
  - c) В виде строк
  - d) В виде возвращаемого значения функции
13. Где обычно размещают id контейнера для React-приложения?

- a) <head>
- b) <header>
- c) <body>
- d) <footer>

14. Какое из этих утверждений не верно?

- a) Компонент может принимать props
- b) Компонент может содержать состояние
- c) Компонент может принимать другие компоненты
- d) Компонент может менять props родителя

15. Какой хук используется для получения ref?

- a) useState()
- b) useEffect()
- c) useRef()
- d) useCallback()

16. Какой метод нужно использовать для отображения массива в JSX?

- a) for
- b) while
- c) map
- d) forEach

17. Что делает propTypes в React?

- a) Передает данные
- b) Задаёт стили
- c) Проверяет тип данных пропсов
- d) Задаёт значения по умолчанию

18. Что такое побочный эффект в React?

- a) Любое действие, не связанное с отрисовкой UI
- b) Функция, которая возвращает массив
- c) Код, который изменяет состояние
- d) Функция, которая вызывает другую функцию

19. Что выведет в консоль следующий код?

20. `function MyComponent({prop}) {`

21. `return (`

22. `<h1>{prop}</h1>`

23. `);`

24. `}`

`ReactDOM.render(<MyComponent prop="hello"/>,`

`document.getElementById('root'));`

`content_copydownload`

Use code [with caution](#).JavaScript

- a) undefined
- b) null
- c) hello
- d) Пустая строка

25. Как можно использовать bootstrap классы в React?

- a) В виде строк
- b) Через style

- c) Через className
- d) Через id

**Ключи ответов**

**Вариант 1**

1. b
2. b
3. b
4. a
5. b
6. c
7. c
8. c
9. b
10. b
11. b
12. c
13. c
14. c
15. c
16. b
17. b
18. c
19. c
20. c

**Вариант 2**

1. b
2. b
3. b
4. a
5. b
6. c
7. c
8. c
9. b
10. b
11. c
12. d
13. c
14. d
15. c
16. c
17. c
18. a
19. c
20. c

# ЭКЗАМЕНАЦИОННЫЕ БИЛЕТЫ

## Экзаменационные билеты по дисциплине «МДК 04.01 Проектирование и дизайн интерфейсов»

Инструкция: Каждый билет содержит два теоретических вопроса и одно практическое задание. Ответьте на вопросы и выполните практическое задание.

---

### Билет № 1

Вопрос 1: Что такое растровые и векторные изображения и в каких случаях лучше использовать каждый из них?

Вопрос 2: Назовите основные инструменты Photoshop для работы с выделениями.

Практическое задание: Создайте в Photoshop коллаж из нескольких фотографий, используя инструменты выделения, слои и маски.

---

### Билет № 2

Вопрос 1: Что такое слои в Photoshop и зачем они нужны?

Вопрос 2: Как создать и использовать стили слоев (layer styles) в Photoshop?

Практическое задание: В Photoshop создайте кнопку с градиентом, тенью и обводкой, используя стили слоя.

---

### Билет № 3

Вопрос 1: Что такое маски в Photoshop и какие их виды вы знаете?

Вопрос 2: Как использовать инструменты для ретуширования изображений в Photoshop (healing brush, clone stamp)?

Практическое задание: Ретушируйте в Photoshop портретную фотографию, удалив мелкие дефекты и неровности кожи.

---

### Билет № 4

Вопрос 1: Что такое цветовые модели RGB и CMYK и в каких случаях они используются?

Вопрос 2: Как создать и использовать смарт-объекты в Photoshop?

Практическое задание: В Photoshop создайте несколько вариантов баннера, используя смарт-объекты для текста и логотипа.

---

### Билет № 5

Вопрос 1: Что такое гайдлайны и зачем они нужны при работе с UI/UX дизайном?

Вопрос 2: Что такое компоненты в Figma и как они помогают ускорить процесс проектирования?

Практическое задание: Создайте в Figma библиотеку компонентов для базовых элементов веб-интерфейса (кнопки, поля ввода, иконки).

---

### Билет № 6

Вопрос 1: Что такое фреймы (frames) в Figma и чем они отличаются от групп (groups)?

Вопрос 2: Как использовать автолейаут (auto layout) в Figma для создания адаптивных элементов?

Практическое задание: В Figma создайте навигационную панель с автолейаутом, которая

автоматически изменяет размеры элементов при изменении размера экрана.

---

#### **Билет № 7**

Вопрос 1: Какие виды шрифтов вы знаете и какие особенности их использования в веб-дизайне?

Вопрос 2: Что такое плагины в Figma и как их устанавливать и использовать?

Практическое задание: В Figma разработайте макет простого лендинга с использованием плагинов для работы с текстом, изображениями и иконками.

---

#### **Билет № 8**

Вопрос 1: Что такое UI kit и для чего он нужен?

Вопрос 2: Как использовать прототипирование в Figma для создания интерактивных макетов?

Практическое задание: В Figma создайте прототип мобильного приложения с несколькими экранами и интерактивными переходами.

---

#### **Билет № 9**

Вопрос 1: Что такое пользовательский опыт (UX) и пользовательский интерфейс (UI)?

Вопрос 2: Как использовать стили (styles) в Figma для унификации элементов?

Практическое задание: В Figma создайте макет страницы сайта, используя общие стили для текста, цветов и элементов.

---

#### **Билет № 10**

Вопрос 1: Что такое сетка (grid) в дизайне и как ее использовать для создания гармоничного макета?

Вопрос 2: Как экспортировать элементы из Photoshop или Figma в разных форматах?

Практическое задание: Создайте в Photoshop или Figma макет страницы с элементами, расположенными в соответствии с модульной сеткой.

---

#### **Билет № 11**

Вопрос 1: Что такое композиция в дизайне и как она влияет на восприятие интерфейса?

Вопрос 2: Как создавать и использовать маски слоя в Figma?

Практическое задание: В Figma создайте макет карточки продукта с использованием масок слоя для создания эффекта глубины и размытия фона.

---

#### **Билет № 12**

Вопрос 1: Что такое иконографика и какие основные правила ее создания?

Вопрос 2: Как использовать варианты (variants) компонентов в Figma для создания гибких элементов?

Практическое задание: В Figma разработайте набор иконок для веб-приложения, используя варианты компонентов для разных состояний (активно, неактивно, наведение).

---

#### **Билет № 13**

Вопрос 1: Что такое типографика и как подбирать шрифты для UI-дизайна?

Вопрос 2: Какие принципы доступности (accessibility) следует учитывать при проектировании интерфейса?

Практическое задание: В Figma создайте макет веб-страницы, где используются

различные шрифты для заголовков и текста. Примените контрастные цвета для обеспечения читаемости.

---

#### **Билет № 14**

Вопрос 1: Как выбрать цветовую палитру для веб-интерфейса?

Вопрос 2: Как использовать эффекты (shadows, blurs) в Figma для создания реалистичного вида?

Практическое задание: В Figma создайте несколько вариантов карточки товара с разными вариантами цветовой палитры и добавьте эффекты для придания объема.

---

#### **Билет № 15**

Вопрос 1: Что такое wireframe и для чего он нужен?

Вопрос 2: Как использовать constraints в Figma для адаптивного дизайна?

Практическое задание: В Figma создайте wireframe для мобильного приложения с базовыми элементами, используя constraints для адаптивности.

---

#### **Билет № 16**

Вопрос 1: Что такое мокап и для чего он нужен?

Вопрос 2: Как использовать component sets в Figma?

Практическое задание: Создайте мокап для мобильного приложения, используя компонент sets.

---

#### **Билет № 17**

Вопрос 1: Какие есть способы создания UI анимаций?

Вопрос 2: Как работать с auto layout?

Практическое задание: Создайте анимацию переключения между страницами, используя smart animate в Figma.

---

#### **Билет № 18**

Вопрос 1: Какие есть правила создания UI элементов?

Вопрос 2: Что такое accessibility в дизайне?

Практическое задание: Разработайте дизайн формы с акцентом на accessibility.

---

#### **Билет № 19**

Вопрос 1: В чем разница между vector и raster изображениями?

Вопрос 2: Какие есть способы задания цвета в Figma?

Практическое задание: Сделайте редизайн существующего сайта, с акцентом на цветовую палитру.

---

#### **Билет № 20**

Вопрос 1: Как использовать стили для унификации дизайна?

Вопрос 2: Что такое micro interactions?

Практическое задание: Разработайте форму с микро анимациями, с использованием плагинов в Figma.

---

#### **Билет № 21**

Вопрос 1: Как выбрать шрифт для сайта?

Вопрос 2: Что такое UI patterns?

Практическое задание: Разработайте несколько вариантов дизайна одной и той же страницы, используя разные UI patterns.

---

#### **Билет № 22**

Вопрос 1: Что такое UX research?

Вопрос 2: Что такое atomic design?

Практическое задание: Разработайте дизайн сайта используя atomic design.

---

#### **Билет № 23**

Вопрос 1: Что такое responsive design?

Вопрос 2: Как работать с constraints?

Практическое задание: Разработайте дизайн для нескольких размеров экрана.

---

#### **Билет № 24**

Вопрос 1: Как создавать UI анимации?

Вопрос 2: Какие инструменты в Figma вы знаете?

Практическое задание: Создайте анимацию и выгрузите ее в mp4.

---

#### **Билет № 25**

Вопрос 1: Какие есть правила верстки для UI дизайна?

Вопрос 2: Как работать со слоями в Figma?

Практическое задание: Создайте экран с несколькими сложными UI элементами, используя Figma.

---

### **Экзаменационные билеты по дисциплине «МДК 04.02 Верстка страниц»**

**Инструкция:** Каждый билет содержит два теоретических вопроса и одно практическое задание. Ответьте на вопросы и выполните практическое задание.

---

#### **Билет № 1**

**Вопрос 1:** Что такое HTML и какова его основная роль в веб-разработке?

**Вопрос 2:** Какие существуют основные способы подключения CSS к HTML-документу?

**Практическое задание:** Создайте HTML-структуру для главной страницы простого блога. Включите заголовок (h1) с названием блога, панель навигации (nav) с тремя ссылками, блок с несколькими статьями (article), где у каждой статьи есть заголовок (h2) и краткое описание (p), и подвал (footer).

---

#### **Билет № 2**

**Вопрос 1:** Опишите разницу между тегами <div> и <span> в HTML.

**Вопрос 2:** Как задать цвет текста с помощью CSS? Приведите пример.

**Практическое задание:** Создайте HTML-список (нумерованный ol или нумерованный ul) и добавьте к нему CSS-стили. Список должен иметь разный цвет маркеров или нумерации, разный шрифт и цвет текста для четных и нечетных элементов списка.

---

### Билет № 3

**Вопрос 1:** Какие типы списков существуют в HTML и чем они отличаются?

**Вопрос 2:** Что такое селекторы в CSS и какие основные типы селекторов вы знаете?

**Практическое задание:** Создайте HTML-таблицу, представляющую расписание занятий. Таблица должна иметь заголовок, не менее 4 строк и 5 столбцов. Добавьте стили для ячеек заголовка (th), а также чередующийся фон для строк таблицы.

---

### Билет № 4

**Вопрос 1:** Что такое атрибуты в HTML и приведите пример их использования.

**Вопрос 2:** Как изменить размер шрифта с помощью CSS? Приведите пример.

**Практическое задание:** Создайте HTML-форму для регистрации пользователя. Форма должна включать поля для ввода имени, email, пароля (маскированного), выбора пола (radio) и флажок (checkbox) для согласия с условиями. Добавьте кнопку "Зарегистрироваться".

---

### Билет № 5

**Вопрос 1:** Какие существуют типы ссылок в HTML и как они работают?

**Вопрос 2:** Как задать отступы вокруг элемента с помощью CSS? Приведите пример.

**Практическое задание:** Создайте блок, содержащий три изображения (img) в ряд. Каждое изображение должно иметь свою ширину, заданную в процентах. Добавьте отступы между изображениями.

---

### Билет № 6

**Вопрос 1:** Что такое теги <head>, <header>, <main> и <footer> в HTML5?

**Вопрос 2:** Что такое псевдоклассы в CSS и для чего они используются?

**Практическое задание:** Создайте блок с текстом и кнопкой. При наведении курсора на блок, меняется цвет фона блока и цвет текста кнопки.

---

### Билет № 7

**Вопрос 1:** Как вставить изображение на веб-страницу и какие атрибуты для этого используются?

**Вопрос 2:** Как выровнять элементы по горизонтали и вертикали с помощью CSS?

**Практическое задание:** Создайте макет, состоящий из заголовка (h1), блока с текстом (p) и блока с изображением. Выровняйте блок с изображением по правому краю, а текст по левому, используя flexbox.

---

### Билет № 8

**Вопрос 1:** Что такое мета-теги и для чего они используются?

**Вопрос 2:** Как задать фон элемента с помощью CSS? Приведите пример.

**Практическое задание:** Создайте кнопку и задайте ей: цвет фона (с градиентом), цвет текста, границу, закругленные углы и эффект изменения цвета при наведении.

---

### Билет № 9

**Вопрос 1:** Что такое формы в HTML и для чего они используются?

**Вопрос 2:** Что такое блочная модель CSS и как она работает?

**Практическое задание:** Создайте форму обратной связи с полями для ввода имени, email, номера телефона (с проверкой на ввод только цифр) и текстового поля для

сообщения. Добавьте кнопку "Отправить". При фокусе на поле, должна меняться граница поля.

---

#### **Билет № 10**

**Вопрос 1:** Какие типы полей ввода вы знаете?

**Вопрос 2:** Как создать границу вокруг элемента с помощью CSS? Приведите пример.

**Практическое задание:** Выведите таблицу с 3 строками и 4 столбцами. Добавьте рамку вокруг всей таблицы, а также вокруг каждой ячейки. Фон ячеек заголовка должен быть серым, а текст белым.

---

#### **Билет № 11**

**Вопрос 1:** Как использовать комментарии в HTML и CSS?

**Вопрос 2:** Как применить несколько классов CSS к одному элементу?

**Практическое задание:** Выведите три изображения в ряд, используя flexbox. Добавьте подписи к каждому изображению. При наведении на изображение - увеличьте его размер и добавьте тень.

---

#### **Билет № 12**

**Вопрос 1:** Что такое атрибут id и чем он отличается от атрибута class?

**Вопрос 2:** Что такое media queries в CSS и для чего они нужны?

**Практическое задание:** Создайте навигационное меню (список ul), которое будет располагаться горизонтально на больших экранах и вертикально на мобильных.

---

#### **Билет № 13**

**Вопрос 1:** Что такое семантическая верстка?

**Вопрос 2:** Как задать прозрачность элемента с помощью CSS? Приведите пример.

**Практическое задание:** Создайте два блока, один поверх другого. Верхний блок должен быть полупрозрачным и иметь границу.

---

#### **Билет № 14**

**Вопрос 1:** Как использовать шрифты, отличные от стандартных в HTML?

**Вопрос 2:** Что такое flexbox и как он помогает создавать макеты?

**Практическое задание:** Создайте макет из блока с текстом и блока с изображением, которые должны быть расположены рядом друг с другом на больших экранах, и друг под другом на мобильных. Используйте flexbox.

---

#### **Билет № 15**

**Вопрос 1:** Какие атрибуты можно использовать для ссылок?

**Вопрос 2:** Что такое CSS-переменные и как их использовать?

**Практическое задание:** Создайте панель навигации (вертикальную или горизонтальную), использующую CSS переменные для цветов и стилей. Добавьте анимацию при наведении на элементы меню.

---

#### **Билет № 16**

**Вопрос 1:** Что такое viewport и для чего он нужен?

**Вопрос 2:** Какие есть способы позиционирования в CSS?

**Практическое задание:** Создайте страницу, где в верхнем правом углу зафиксировано

изображение, не прокручивающееся с основным контентом.

---

#### **Билет № 17**

**Вопрос 1:** Как использовать разные стили для разных устройств?

**Вопрос 2:** Какие есть способы задания цвета в CSS?

**Практическое задание:** Выведите изображение с подписью. Изображение должно быть адаптивным. При наведении на него текст подписи должен менять цвет.

---

#### **Билет № 18**

**Вопрос 1:** Какие есть способы выравнивания элементов в CSS?

**Вопрос 2:** Как сделать картинку адаптивной?

**Практическое задание:** Создайте страницу с заголовком, тремя абзацами, кнопкой и формой. Все элементы должны быть с отступами. Используйте flexbox для выравнивания элементов.

---

#### **Билет № 19**

**Вопрос 1:** В чем отличие padding от margin?

**Вопрос 2:** Что такое grid в CSS?

**Практическое задание:** Создайте таблицу с 3 строками и 3 столбцами, используя grid. В таблице выведите заголовок таблицы и данные в каждой ячейке. Добавьте анимацию при наведении на ячейку.

---

#### **Билет № 20**

**Вопрос 1:** Как использовать background images?

**Вопрос 2:** Что такое относительные и абсолютные единицы измерения в CSS?

**Практическое задание:** Создайте список, где при наведении на элемент меняется цвет его фона и появляется иконка справа. Используйте background images.

---

#### **Билет № 21**

**Вопрос 1:** Как выравнивать текст в блоке?

**Вопрос 2:** Как использовать transition в CSS?

**Практическое задание:** Разместите три изображения в ряд. При наведении на изображение оно должно немного увеличиваться и сдвигаться вверх. Используйте transition.

---

#### **Билет № 22**

**Вопрос 1:** Что такое кроссбраузерность?

**Вопрос 2:** Как задать font-weight в CSS?

**Практическое задание:** Создайте заголовок, два абзаца, список и кнопку. Задайте им разные шрифты (используя @font-face), цвета и размеры.

---

#### **Билет № 23**

**Вопрос 1:** Как использовать icon-font?

**Вопрос 2:** Что такое z-index в CSS?

**Практическое задание:** Создайте круглый блок. При наведении на него появлялась иконка в центре и прозрачность должна увеличиваться. Используйте z-index для позиционирования иконки.

---

**Билет № 24**

**Вопрос 1:** Как использовать svg изображения?

**Вопрос 2:** Что такое Box sizing в CSS?

**Практическое задание:** Создайте страницу с формой (разными полями) и таблицей (с рамками и фоном для шапки). Разместите форму и таблицу рядом.

---

**Билет № 25**

**Вопрос 1:** Что такое препроцессоры CSS?

**Вопрос 2:** Как работают float и clear в CSS?

**Практическое задание:** Разместите блок слева, а текст справа от него, с обтеканием текста вокруг блока. Используйте float и clear.

---

**Экзаменационные билеты по дисциплине «МДК 04.03 Разработка клиентской части информационных ресурсов»**

**Инструкция:** Каждый билет содержит два теоретических вопроса и одно практическое задание. Ответьте на вопросы и выполните практическое задание.

---

**Билет № 1**

**Вопрос 1:** Что такое DOM (Document Object Model) и как JavaScript взаимодействует с ним?

**Вопрос 2:** Объясните разницу между var, let и const в JavaScript.

**Практическое задание:** Создайте HTML-страницу с кнопкой и текстом. При нажатии на кнопку текст должен меняться. Используйте JavaScript для управления DOM.

---

**Билет № 2**

**Вопрос 1:** Что такое функции высшего порядка в JavaScript? Приведите пример.

**Вопрос 2:** Что такое JSX и почему он используется в React?

**Практическое задание:** Создайте простой React-компонент, который выводит заголовок и абзац с текстом.

---

**Билет № 3**

**Вопрос 1:** Что такое замыкания (closures) в JavaScript?

**Вопрос 2:** Что такое props в React и как они используются для передачи данных?

**Практическое задание:** Создайте React-компонент Button и передайте ему пропсы text (текст кнопки) и onClick (функцию-обработчик).

---

**Билет № 4**

**Вопрос 1:** Что такое асинхронность в JavaScript и как работают async/await?

**Вопрос 2:** Что такое state в React и чем он отличается от props?

**Практическое задание:** Создайте React-компонент Counter с кнопками "+" и "-". Используйте useState для управления состоянием счетчика.

---

**Билет № 5**

**Вопрос 1:** Объясните разницу между `==` и `===` в JavaScript.

**Вопрос 2:** Что такое хуки в React и для чего они используются?

**Практическое задание:** Создайте React-компонент, который отображает текущее время. Используйте `useEffect` для обновления времени каждую секунду.

---

#### Билет № 6

**Вопрос 1:** Что такое JSON и как он используется в веб-разработке?

**Вопрос 2:** Что такое условный рендеринг в React и как его реализовать?

**Практическое задание:** Создайте React-компонент, который отображает сообщение в зависимости от значения `boolean` пропса (например, сообщение об успехе или ошибке).

---

#### Билет № 7

**Вопрос 1:** Что такое прототипное наследование в JavaScript?

**Вопрос 2:** Что такое `key` при рендеринге списков в React и для чего он нужен?

**Практическое задание:** Создайте React-компонент, который выводит список из 5 элементов (с использованием `map`).

---

#### Билет № 8

**Вопрос 1:** Как работают обработчики событий в JavaScript?

**Вопрос 2:** Какие существуют способы загрузки данных с сервера в React?

**Практическое задание:** Создайте React-компонент, который загружает данные о пользователе с помощью `fetch` (используйте любой публичный API).

---

#### Билет № 9

**Вопрос 1:** Что такое `localStorage` и `sessionStorage` в JavaScript?

**Вопрос 2:** Что такое `controlled components` в React и как их использовать?

**Практическое задание:** Создайте форму с текстовым полем и кнопкой отправки. Используйте `useState` для управления состоянием поля.

---

#### Билет № 10

**Вопрос 1:** Что такое API (Application Programming Interface) и как его использовать в веб-разработке?

**Вопрос 2:** Что такое React Router и для чего он используется?

**Практическое задание:** Настройте React Router для создания двух страниц: домашней и страницы контактов.

---

#### Билет № 11

**Вопрос 1:** Какие методы массивов вы знаете?

**Вопрос 2:** Какие существуют способы стилизации компонентов в React?

**Практическое задание:** Создайте список из 5 элементов, используя метод `map`. Стилизируйте его.

---

#### Билет № 12

**Вопрос 1:** Что такое `debounce` и `throttle`?

**Вопрос 2:** В чем разница между `state` и `props`?

**Практическое задание:** Создайте `input`, который будет выводить значение в консоль. Используйте `debounce`.

---

**Билет № 13**

**Вопрос 1:** Какие есть способы создать компонент в React?

**Вопрос 2:** Что такое Context API и для чего он нужен?

**Практическое задание:** Создайте компонент, который выводит значение из контекста.

---

**Билет № 14**

**Вопрос 1:** Что такое тестирование?

**Вопрос 2:** Что такое ref в React и как его использовать?

**Практическое задание:** Создайте поле ввода и получите на него ref. При нажатии кнопки сделать фокус на этом поле.

---

**Билет № 15**

**Вопрос 1:** Что такое асинхронные функции и как они работают?

**Вопрос 2:** Какие методы жизненного цикла есть у классовых компонентов в React?

**Практическое задание:** Создайте компонент и выведите в консоль какой метод сработал при монтировании и при обновлении.

---

**Билет № 16**

**Вопрос 1:** Что такое Event bubbling и Event capturing?

**Вопрос 2:** Что такое render props?

**Практическое задание:** Создайте компонент, который отображает контент в зависимости от переданного render prop.

---

**Билет № 17**

**Вопрос 1:** Что такое Virtual DOM в React?

**Вопрос 2:** Как работает shouldComponentUpdate?

**Практическое задание:** Создайте компонент, который не будет перерендериваться при обновлении пропсов.

---

**Билет № 18**

**Вопрос 1:** Что такое higher-order components?

**Вопрос 2:** В чем разница между useLayoutEffect и useEffect?

**Практическое задание:** Создайте хук который будет проверять авторизацию.

---

**Билет № 19**

**Вопрос 1:** Что такое memoization?

**Вопрос 2:** Какие есть способы оптимизации производительности React-приложений?

**Практическое задание:** Создайте компонент, использующий React.memo для избежания ненужных ререндеров.

---

**Билет № 20**

**Вопрос 1:** Что такое SSR?

**Вопрос 2:** В чем разница между useCallback и useMemo?

**Практическое задание:** Создайте компонент и оптимизируйте коллбеки через useCallback.

---

**Билет № 21**

**Вопрос 1:** Что такое immutability?

**Вопрос 2:** Для чего нужны ключи при использовании map?

**Практическое задание:** Создайте список, в котором будет добавлена анимация при добавлении элемента.

---

**Билет № 22**

**Вопрос 1:** Что такое CORS и как его обойти?

**Вопрос 2:** Как настроить роутинг в React?

**Практическое задание:** Настройте приложение с тремя страницами, используйте NavLink.

---

**Билет № 23**

**Вопрос 1:** Что такое Redux или Context API?

**Вопрос 2:** Что такое styled-components?

**Практическое задание:** Создайте приложение и используйте styled-components.

---

**Билет № 24**

**Вопрос 1:** Что такое unit test?

**Вопрос 2:** Что такое controlled components?

**Практическое задание:** Создайте поле ввода, которое будет возвращать результат в зависимости от валидации.

---

**Билет № 25**

**Вопрос 1:** Что такое rest api?

**Вопрос 2:** Какие есть способы обработки ошибок?

**Практическое задание:** Создайте компонент, который получает данные с сервера и обрабатывает ошибку.

**Макет программы государственной итоговой аттестации**  
Государственное бюджетное профессиональное образовательное учреждение  
Республики Хакасия  
Техникум коммунального хозяйства и сервиса

УТВЕРЖДЕНА

Приказ № \_\_\_\_\_

От «\_\_\_\_» \_\_\_\_\_ 20  
\_\_\_\_\_ г.

Рассмотрена:

на заседании педагогического  
совета техникума

Протокол № \_\_\_\_ от \_\_\_\_\_ 2019 г.

Программа согласована:

председатель государственной  
экзаменационной комиссии: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

(ФИО, подпись)

**ПРОГРАММА  
ГОСУДАРСТВЕННОЙ ИТОГОВОЙ АТТЕСТАЦИИ**

для подготовки \_\_\_\_\_ по профессии/специальности (выбрать)

| Код | наименование |
|-----|--------------|
|-----|--------------|

Абакан, 20\_\_\_\_

## Содержание

|                                                          |  |
|----------------------------------------------------------|--|
| 1. Паспорт программы .....                               |  |
| 2. Требования к дипломным проектам (работам) .....       |  |
| 3. Методика оценивания дипломного проекта (работы) ..... |  |
| 4. Задание на демонстрационный экзамен.....              |  |
| 5. Условия проведения.....                               |  |

## 1. Паспорт программы

1.1. Наименование специальности/профессии: \_\_\_\_\_

«\_\_\_\_\_».

Квалификация (и): \_\_\_\_\_;

Срок получения образования по образовательной программе - \_\_\_\_\_.

1.2. Форма ГИА \_\_\_\_\_

1.3. Уровень демонстрационного экзамена \_\_\_\_\_

1.4. Программа ГИА разработана на основе:

- Федерального государственного образовательного стандарта среднего профессионального образования \_\_\_\_\_;
- Порядка проведения государственной итоговой аттестации по образовательным программам среднего профессионального образования, утв. Приказом Министерства просвещения РФ №800 от 8.11.2021г;
- Оценочных материалов для демонстрационного экзамена в 20\_\_\_\_году;

1.3. Цель государственной итоговой аттестации.

Государственная итоговая аттестация проводится с целью определения соответствия результатов освоения студентами образовательной программы требованиям федерального государственного образовательного стандарта среднего профессионального образования по специальности/профессии \_\_\_\_\_  
«\_\_\_\_\_».

Задачи: (выбрать)

демонстрационного экзамена – определение уровня освоения выпускниками материала, предусмотренного образовательной программой, и степени сформированности профессиональных умений и навыков путем проведения независимой экспертной оценки выполнения выпускником практических заданий в условиях реальных или смоделированных производственных процессов.

Дипломного проекта (работы) – систематизация и закрепление знаний выпускника по специальности, а также определения уровня готовности выпускника к самостоятельной деятельности.

1.4. Требования к результатам освоения основной профессиональной образовательной программы

Выпускник, получивший квалификации: \_\_\_\_\_

\_\_\_\_\_ должен быть подготовлен к выполнению следующих основных видов деятельности:

1. ВД 01. \_\_\_\_\_
2. \_\_\_\_\_
3. ВД 0.... \_\_\_\_\_

| Требования ФГОС по профессии в соответствии с квалификациями | Формы проверки освоения результатов             |                                              |                                 |     |
|--------------------------------------------------------------|-------------------------------------------------|----------------------------------------------|---------------------------------|-----|
|                                                              | Промежуточная аттестация                        |                                              |                                 | ГИА |
|                                                              | Дифференциро в анный зачет по учебным практикам | Дифференциро в анный зачет по производственн | Экзамен по модулю (квалификацио |     |

|  |  |              |        |  |
|--|--|--------------|--------|--|
|  |  | ом практикам | н ный) |  |
|--|--|--------------|--------|--|

|         |  |  |  |  |
|---------|--|--|--|--|
| ПК 1.1. |  |  |  |  |
| ПК 1.2. |  |  |  |  |
| ПК 1.3. |  |  |  |  |
| ПК 1.4. |  |  |  |  |
| ....    |  |  |  |  |

## **2. Требования к дипломным проектам (работам)**

## **3. Методика оценивания дипломного проекта (работы)**

## **4. Задание на демонстрационный экзамен**

описать

Приложение 1 комплект оценочной документации с критериями оценки

## **5. Условия проведения**

### **5.1. План проведения ДЭ:**

- дата и время начала проведения ДЭ,
- расписание экзаменов в составе экзаменационных групп
- планируемая продолжительность времени
- технические перерывы