

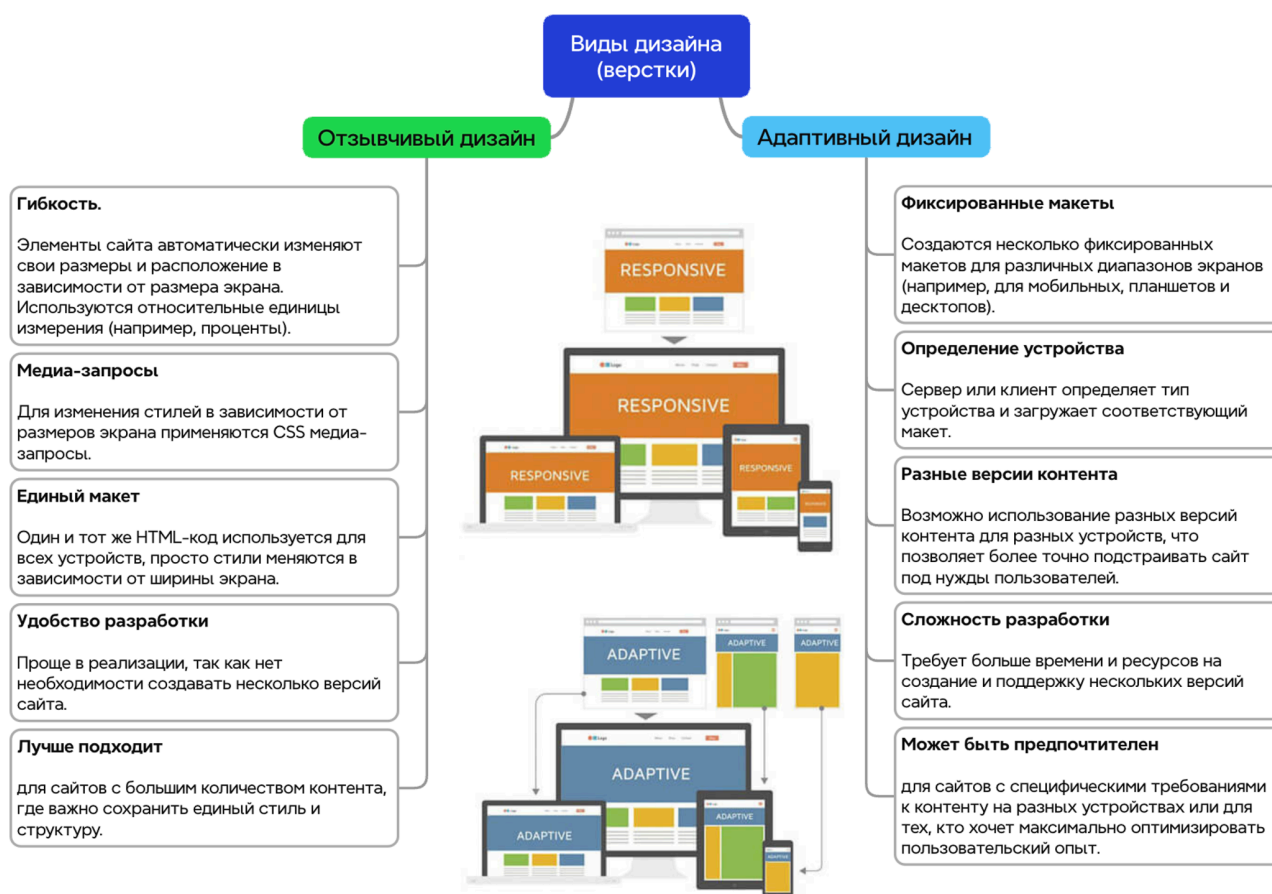
ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Загородних Николай Анатольевич
СЕМЕСТР	1 семестр, 2024-2025 гг.

# Практическое занятие 7. Адаптивная вёрстка

## Краткая теория

Адаптивный дизайн — это подход к созданию веб-сайтов, который обеспечивает их корректное отображение на различных устройствах и экранах, независимо от их размера или разрешения. Он основывается на принципе, что пользовательский опыт должен оставаться высококачественным и удобным, будь то смартфон, планшет или настольный компьютер. Адаптивный дизайн использует гибкие сетки, медиа-запросы и адаптивные изображения, чтобы автоматически подстраивать элементы интерфейса под конкретные условия отображения.

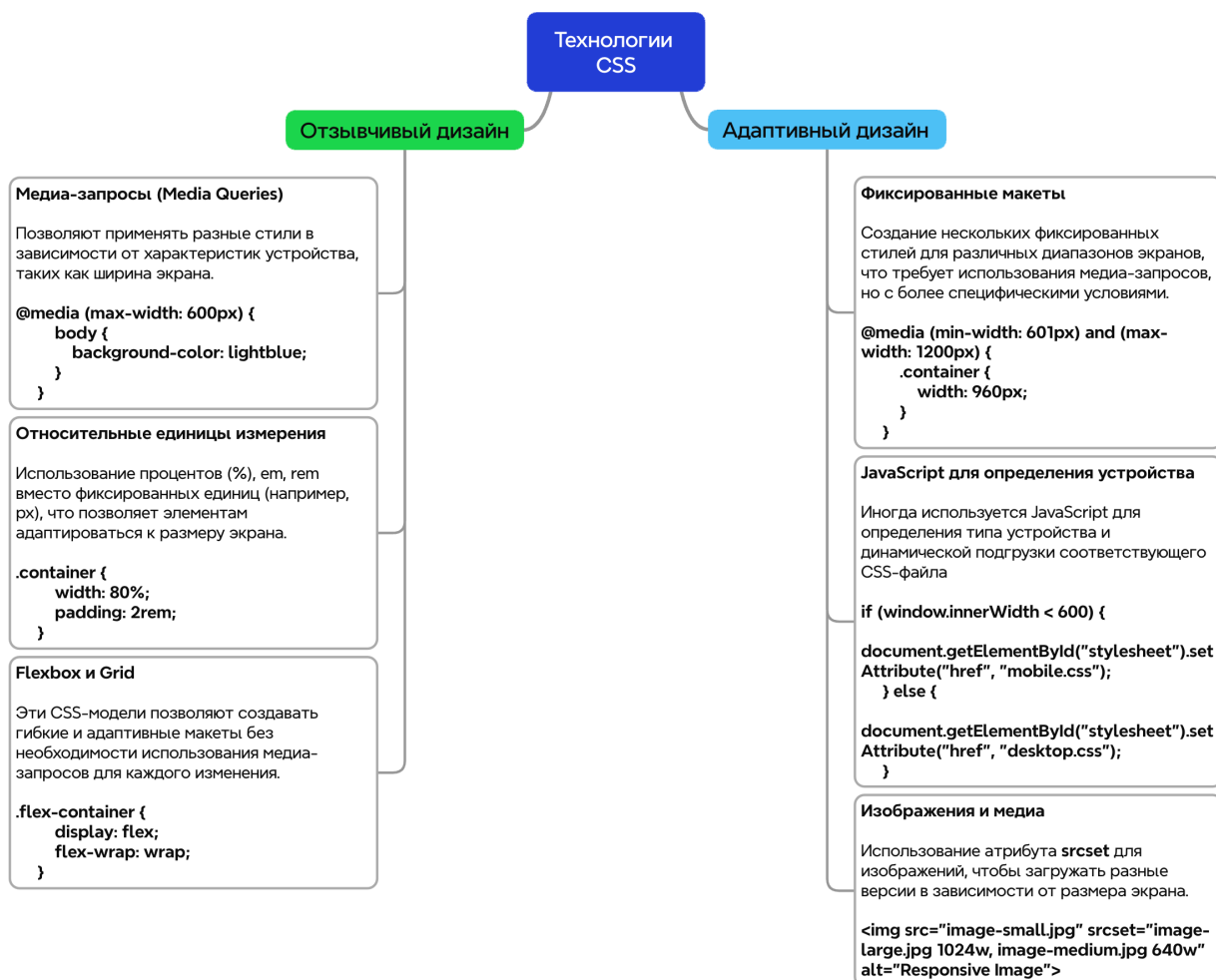
Адаптивный дизайн часто путают с отзывчивым (или респонсивным) дизайном. Несмотря на то, что оба подхода направлены на создание удобного пользовательского интерфейса на различных устройствах, между ними есть ключевые различия.



**Адаптивный дизайн** предполагает создание нескольких фиксированных макетов для различных категорий устройств. Например, можно создать отдельные версии сайта для мобильных телефонов, планшетов и настольных компьютеров. Когда пользователь

заходит на сайт с определенным устройством, ему загружается наиболее подходящая версия.

**Отзывчивый дизайн**, в свою очередь, использует одну версию сайта, которая динамически изменяет свои размеры и элементы в зависимости от ширины экрана. Это достигается с помощью гибких сеток и медиа-запросов, которые позволяют изменять стили в реальном времени.



Основные принципы адаптивной вёрстки включают использование медиа-запросов, гибких сеток и изображений, а также правильное применение CSS Flexbox и Grid.

### Основные понятия:

- Медиа-запросы — это CSS-инструкции, которые применяются при определённых условиях, таких как ширина экрана.
- Flexbox и Grid — это современные CSS-технологии для создания гибких и адаптивных макетов.

## Использование медиа-запросов. Синтаксис и примеры

Синтаксис медиа-запроса выглядит следующим образом:

```
* CSS
1 @media media-type and (condition) {
2     /* CSS правила */
3 }
4
```

где **media-type** может быть screen, print и т.д.,

**condition** — это условие, которому должно соответствовать устройство.

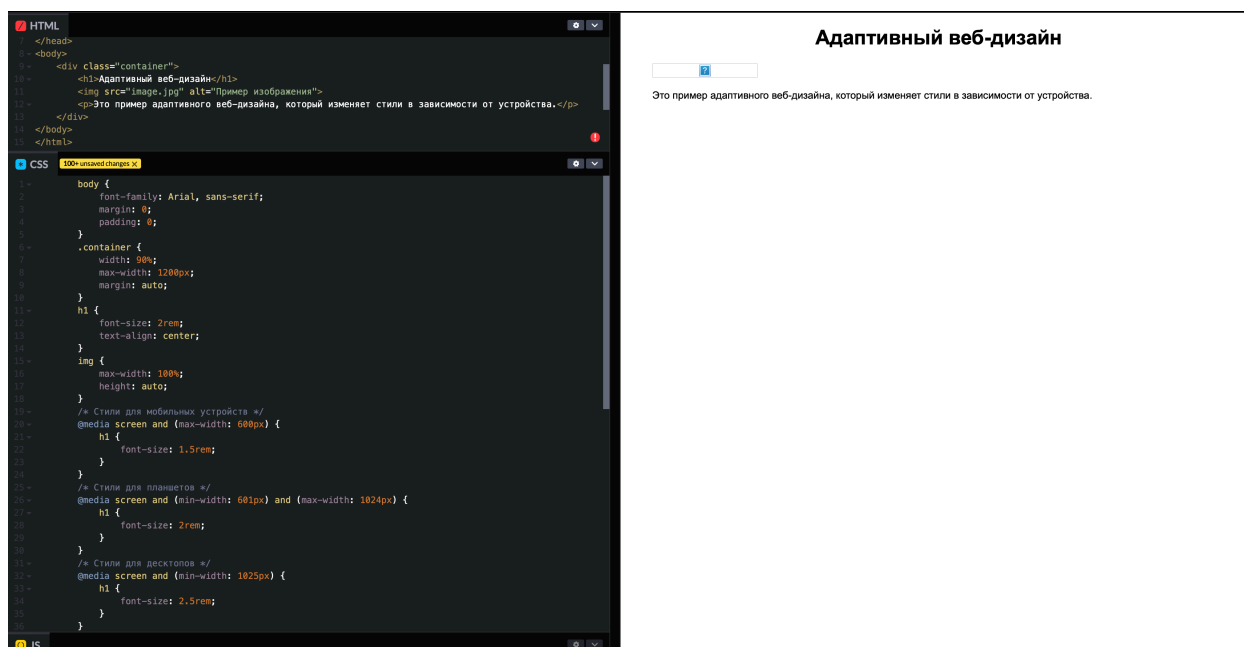
Пример простого медиа-запроса:

```
* CSS
1 /* Стили для мобильных устройств */
2 @media screen and (max-width: 600px) {
3     body {
4         background-color: lightblue;
5     }
6
7     h1 {
8         font-size: 24px;
9     }
10 }
11
12 /* Стили для планшетов */
13 @media screen and (min-width: 601px) and (max-width: 1024px) {
14     body {
15         background-color: lightgreen;
16     }
17
18     h1 {
19         font-size: 32px;
20     }
21 }
22
23 /* Стили для десктопов */
24 @media screen and (min-width: 1025px) {
25     body {
26         background-color: lightcoral;
27     }
28
29     h1 {
30         font-size: 40px;
31     }
32 }
```

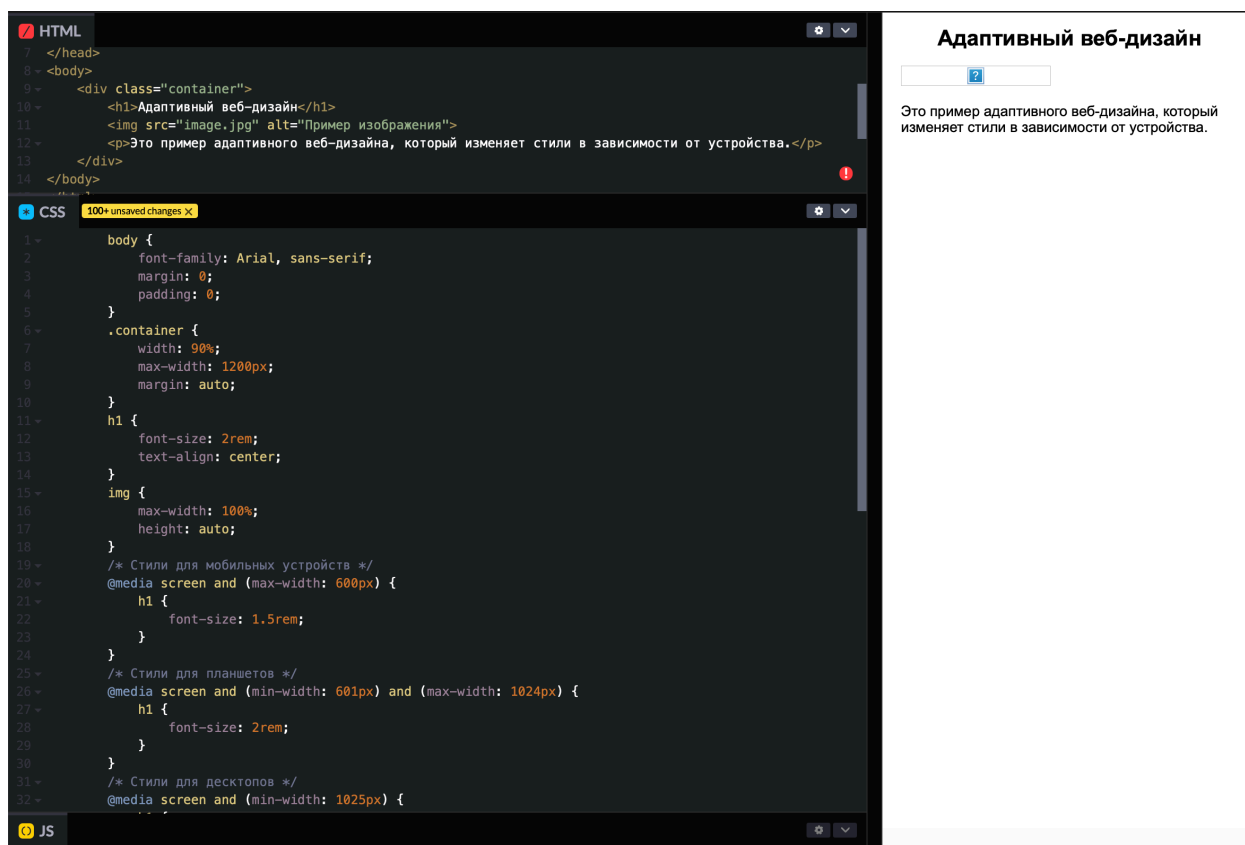
В этом примере, в зависимости от ширины экрана, цвет фона и размер заголовка изменяются. Это позволяет создать адаптивный интерфейс, который выглядит хорошо на любых устройствах.

### Пример адаптивного дизайна с использованием CSS

Рассмотрим пример адаптивного веб-сайта, который использует все вышеперечисленные принципы.

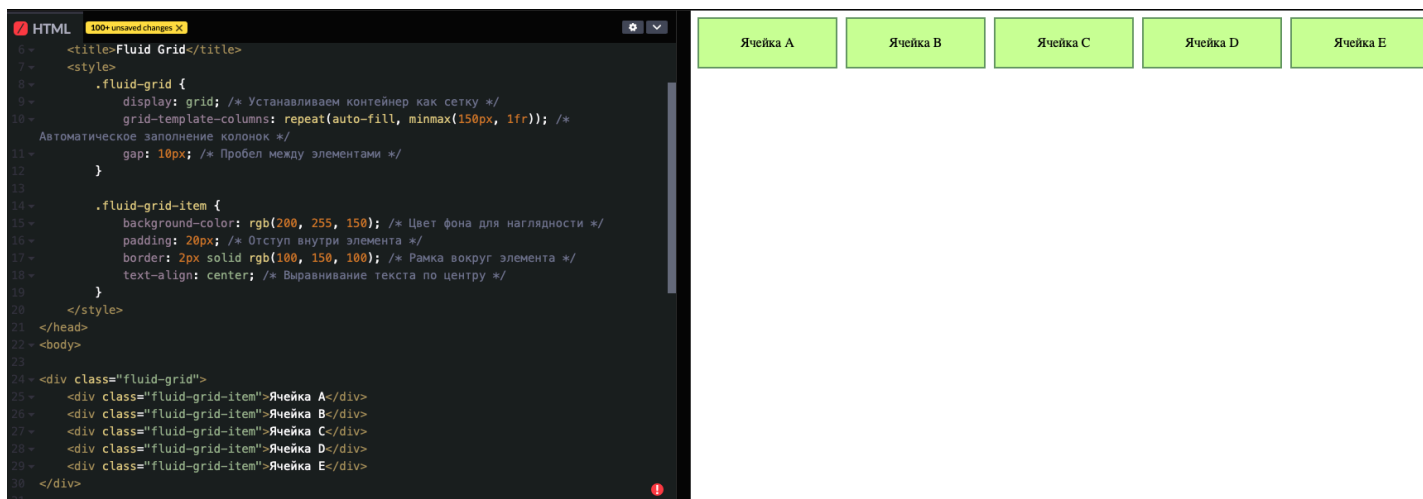


В этом примере мы создали простую веб-страницу, использующую адаптивный дизайн. Заголовок и изображения адаптируются под размер экрана, обеспечивая удобное восприятие информации пользователем.



## Создание fluid grid

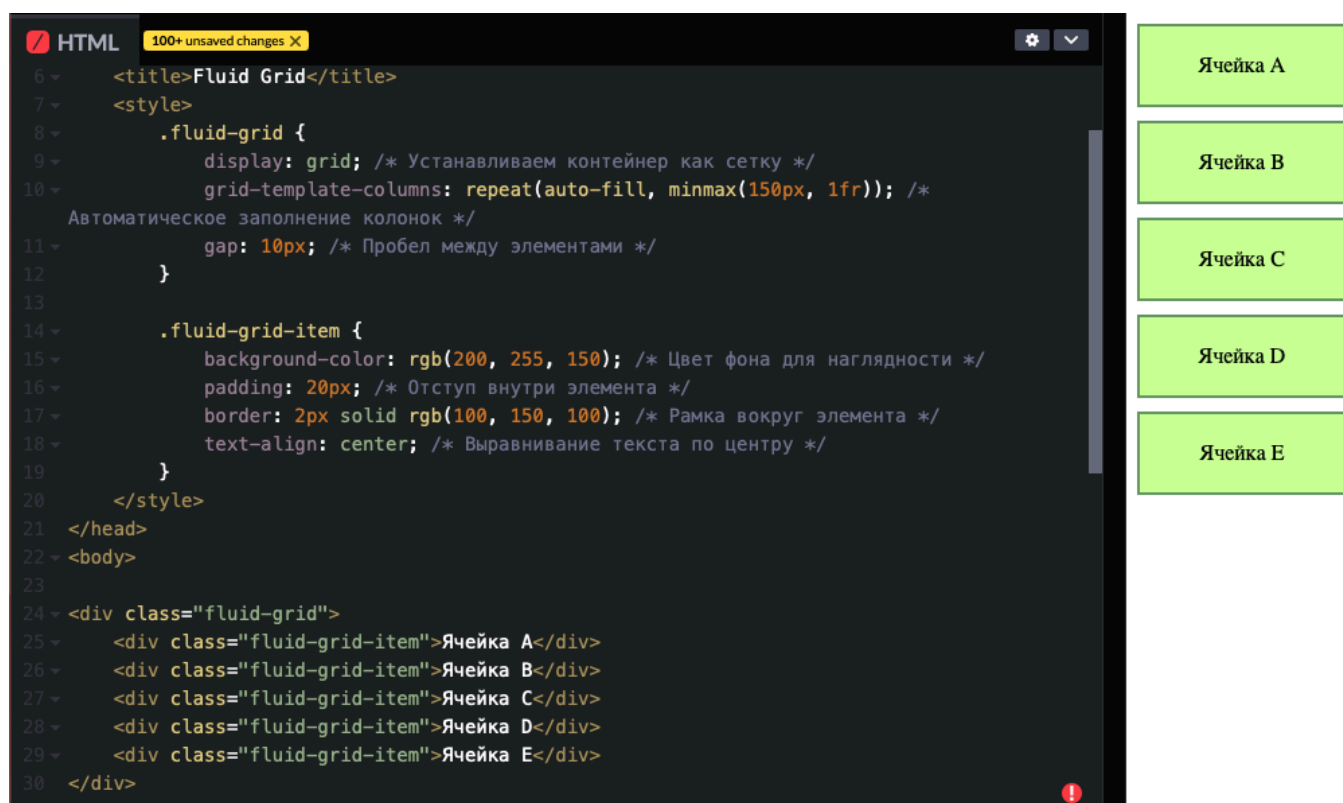
Fluid grid (жидкая сетка) адаптируется к размеру экрана, позволяя элементам изменять свои размеры в зависимости от доступного пространства. Это особенно полезно для создания адаптивных веб-дизайнов.

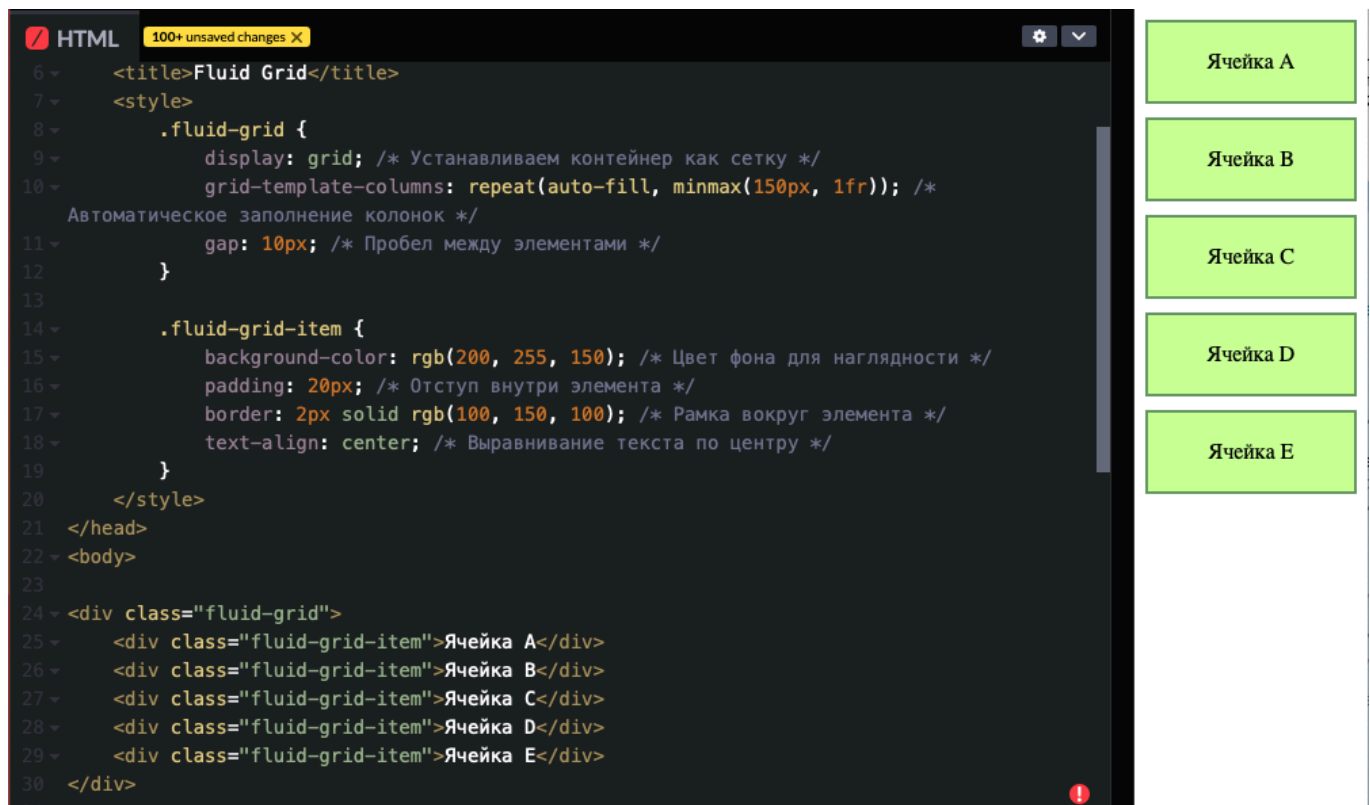


- .fluid-grid контейнер с автоматическим заполнением колонок. Каждая колонка будет иметь минимальную ширину 150 пикселей и будет занимать оставшееся пространство.

-.fluid-grid-item элементы сетки с цветом фона и рамкой, которые будут адаптироваться к размеру экрана.

При изменении разрешения экрана контент перестраивается.





## Полезные ссылки:

- Адаптивный дизайн на hcdev <https://hcdev.ru/learn/design>
- Flex <https://hcdev.ru/learn/flex/>
- Grid <https://hcdev.ru/learn/grid/>

## Тренажёры:

<https://codepen.io/pen>

## Задание

Переверстайте весь ваш проект так, чтобы он корректно отображался на разных устройствах и разных разрешениях экрана, для этого:

### 1. Создайте медиа-запросы для адаптивности:

- Настройте стили для различных размеров экранов (например, для мобильных устройств, планшетов и десктопов).

- Используйте медиа-запросы для изменения расположения элементов на странице.

### 2. Используйте Flexbox и Grid:

- Примените Flexbox чтобы элементы располагались горизонтально и адаптировались к ширине экрана.

- Используйте CSS Grid для размещения элементов дизайна в теле сайта.

3. Адаптация изображений - убедитесь, что изображения корректно масштабируются и не выходят за пределы контейнера.

Используйте свойство `max-width: 100%;` для изображений.

4. Форма - сделайте форму адаптивной, изменяя размеры полей ввода и кнопок в зависимости от ширины экрана.

5. Таблицы - настройте стили для таблиц так, чтобы они были читабельны на мобильных устройствах (например, используйте горизонтальную прокрутку).

6. Встроенное видео и PDF - убедитесь, что встроенное видео и PDF корректно отображаются на различных устройствах.

7. Закрепите меню так чтобы при прокрутке сайта оно всегда оставалось сверху.

8. Добавьте внизу страницы кнопку с фиксированным позиционированием при клике на которую осуществлялась прокрутка в верх страницы.

## **Критерии оценивания**

- Код. Чистота и структурированность HTML и CSS кода.

- Адаптивность. Правильное использование медиа-запросов, Flexbox и Grid.

- Функциональность. Корректная работа всех элементов на разных устройствах.

## **Форма сдачи**

Студенты должны представить свой код в виде файлов `.html` `.css`, который можно открыть в браузере. Также важно прокомментировать код, чтобы объяснить, что делает каждый элемент.



# Макет мобильной версии сайта

Logo

Группа \_\_\_\_\_  
Иванов Иван Иванович

☰

Разработка веб-приложений

Сайт-визитка

Заказать

Корпоративный сайт

Заказать

Интернет-магазин

Заказать

© Иванов Иван Иванович, 2024

Logo

Группа \_\_\_\_\_  
Иванов Иван Иванович

☰

Добро пожаловать на мой сайт

Меня зовут Иванов Иван Иванович и я начинающий специалист по веб-разработке.

Фронтенд

Фронтенд (или клиентская часть) — это то, что видит пользователь. Он включает в себя все визуальные элементы веб-сайта или приложения: кнопки, формы, изображения и текст. Фронтенд разработчики используют языки разметки и стилизации, такие как HTML, CSS и JavaScript, чтобы создавать интерактивные и привлекательные интерфейсы. Основная задача фронтенда — обеспечить удобство использования и эстетическую привлекательность.

Бэкенд

Бэкенд (или серверная часть) — это "за кулисами" веб-приложения. Он отвечает за обработку данных, управление пользователями и взаимодействие с базами данных. Бэкенд разработчики используют языки программирования, такие как Python, Ruby, Java или PHP, а также фреймворки и базы данных для создания логики приложения. Основная задача бэкенда — обеспечить стабильную работу приложения и безопасность данных.

© Иванов Иван Иванович, 2024

Logo

Группа \_\_\_\_\_  
Иванов Иван Иванович

☰

Главная

Первый семестр

Второй семестр

Услуги

Авторизация

Разработка веб-приложения

Заполните форму и мы обязательно свяжемся с вами в ближайшее время для уточнения деталей

Как вас зовут

Иванов Иван Иванович

Ваш e-mail

test@mail.ru

Телефон

+7 999 999 99 99

Страна

Россия

Дата

//

Комментарий

☑️ Согласен на обработку моих персональных данных

Отправить заявку