**HTML. Урок 12: Адаптивная и отзывчивая вёрстка (Responsive Web Design)**

**📌 Цели урока**

* Понять разницу между адаптивной и отзывчивой (responsive) вёрсткой.
* Научиться делать сайты, которые хорошо выглядят на любом устройстве.
* Освоить медиазапросы, относительные единицы, гибкие сетки и изображения.
* Практически сверстать простую адаптивную страницу.
* Избегать типовых ошибок при разработке “под мобильные”.

**🧠 Теория**

**1. Почему это важно?**

Сегодня до 80% пользователей заходят на сайты с мобильных!  
Если сайт не адаптирован — теряется большая часть аудитории и поисковики занижают позиции.

**2. Ключевые принципы Responsive Web Design**

* **Гибкая сетка** (fluid grid) — ширина блоков задаётся в % или fr, а не в px.
* **Гибкие изображения** — изображения “сжимаются” вместе с контейнером.
* **Медиазапросы (media queries)** — CSS-правила, которые срабатывают только на определённых экранах.
* **Относительные единицы** — em, rem, vw, vh, %, а не только px.

**3. Мета-тег viewport**

Обязательный тег в <head>, чтобы сайт работал на мобильных:

html

Копировать код

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

**4. Медиазапросы: синтаксис и примеры**

css

Копировать код

@media (max-width: 900px) {

.container { width: 95%; }

nav ul { flex-direction: column; gap: 8px; }

.sidebar { display: none; }

}

@media (max-width: 600px) {

h1 { font-size: 24px; }

.card { width: 100%; margin-bottom: 18px; }

}

* max-width — правило срабатывает для экранов уже указанной ширины.
* min-width — для экранов шире указанного значения.

**5. Гибкие сетки (flex и grid)**

* **Flexbox** — удобно для выравнивания и размещения блоков по горизонтали/вертикали.

css

Копировать код

.cards {

display: flex;

flex-wrap: wrap;

gap: 20px;

}

.card {

flex: 1 1 300px; /\* Растягивается, но не меньше 300px \*/

}

* **CSS Grid** — удобно для сложных сеток.

css

Копировать код

.gallery {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

gap: 20px;

}

**6. Гибкие изображения и видео**

css

Копировать код

img, video {

max-width: 100%;

height: auto;

display: block;

}

* Так картинки и видео не “выходят” за пределы блока.

**7. Мобильное меню и навигация**

* На мобильном чаще делают “гамбургер-меню” — меню скрыто, появляется по клику.
* Для простоты — делай меню вертикальным, крупные кнопки, увеличь отступы.

**8. Тестирование адаптива**

* Проверь сайт на разных устройствах и браузерах.
* Используй инструменты разработчика (DevTools) — эмуляция размеров экранов.

**9. Частые ошибки**

* Не задан viewport — сайт ломается на телефонах.
* Фиксированная ширина в px — элементы “вываливаются”.
* Картинки без max-width — не сжимаются.
* Слишком мелкие кнопки/текст.
* Не адаптировано меню.

**📚 Глоссарий**

| **Понятие** | **Описание** |
| --- | --- |
| Responsive | “Отзывчивый”, т.е. подстраивается под экран |
| Медиазапрос | Спец. CSS-правило для разных устройств |
| Flexbox | Модель гибкой компоновки блоков |
| CSS Grid | Мощная система сетки |
| Viewport | Область просмотра браузера |
| Адаптивная вёрстка | Вёрстка, которая “подстраивается” |
| Гамбургер-меню | Кнопка-три полоски для вызова меню |

**🛠 Практика**

1. Сделай адаптивную “шапку”: лого слева, меню справа — на мобильном всё по центру, меню вертикально.
2. Сверстай галерею товаров: flex или grid, чтобы 4 в ряд на desktop, 2 — на планшете, 1 — на мобильном.
3. Оформи карточки товаров: чтобы фото, название и кнопка красиво уменьшались на разных экранах.
4. Проверь отображение на экранах 1920, 1280, 900, 600 и 360 пикселей.

**🎯 Мини-проекты**

**Проект 1:**  
Лендинг “Услуги”: шапка, три секции, подвал. Всё адаптивно, красиво, текст и картинки не “убегают”.

**Проект 2:**  
Портфолио с галереей: 6 работ в сетке grid, адаптация для планшета и мобильного.

**💡 Лайфхаки**

* Делай сначала desktop, потом “сжимай” и добавляй медиазапросы (“mobile-first” — тоже хорошая стратегия).
* Не злоупотребляй скрытием блоков — лучше перестраивать.
* Используй относительные размеры (%, fr, em, rem) для гибкости.
* Не бойся тестировать на реальных устройствах!
* Сохрани единый стиль и логику для всех экранов.

**❓ Вопросы для самопроверки**

* Что такое медиазапрос и зачем он нужен?
* Как задать гибкую сетку товаров?
* Как не дать изображению “вылезти” за блок?
* Зачем нужен meta viewport?
* Как адаптировать меню для телефона?

**📑 Итоги**

Теперь ты умеешь:

* Верстать сайты, которые отлично выглядят и работают на любых устройствах.
* Использовать медиазапросы, flex, grid, относительные единицы.
* Не допускать ошибок с размерами, картинками, меню.
* Проверять и тестировать свои проекты на всех популярных устройствах.