Подробная лекция: Основы CSS

## Введение

Сегодняшняя тема — основы CSS. Мы поговорим о том, как с помощью CSS стилизовать веб-страницы, управлять цветами, шрифтами, расположением элементов, отступами и многим другим. Веб-страницы — это не только структура (HTML), но и оформление (CSS). Без CSS сайты выглядели бы скучно и однообразно.

## Историческая справка

CSS (Cascading Style Sheets — каскадные таблицы стилей) был предложен Хаконом Виумом Ли (Håkon Wium Lie) в 1994 году, когда он работал в CERN вместе с Тимом Бернерс-Ли, создателем WWW. До появления CSS веб-страницы оформлялись через HTML-теги напрямую: цвет, размер шрифта, выравнивание — всё делалось вручную и неэффективно. CSS отделил стиль от структуры, что сделало разработку сайтов более удобной и масштабируемой. Первая спецификация CSS была официально опубликована в декабре 1996 года.

## Что такое CSS и зачем он нужен

CSS — это язык, описывающий внешний вид HTML-документов. Он позволяет разработчику описывать стили (цвет, размер, отступы, выравнивание и т.д.) для различных элементов. Основная цель CSS — отделение контента (HTML) от оформления (CSS). Это повышает удобство поддержки, повторного использования и улучшает внешний вид сайтов.

## Синтаксис CSS

Простейшее правило CSS выглядит так:  
  
селектор {  
 свойство: значение;  
}  
  
Пример:  
p {  
 color: red;  
}  
  
Здесь:  
- `p` — это селектор (абзац)  
- `color` — свойство  
- `red` — значение свойства

## Подключение CSS к HTML

Есть три способа подключения CSS:  
1. \*\*Встроенный стиль (inline):\*\*  
<p style="color: red">Текст</p>  
2. \*\*Встроенные стили внутри <style>:\*\*  
<style>  
 p { color: red; }  
</style>  
3. \*\*Внешний CSS-файл:\*\*  
<link rel="stylesheet" href="style.css">  
  
На практике рекомендуется использовать внешний CSS.

## Типы селекторов в CSS

1. Селектор по тегу: p, h1, body  
2. Селектор по классу: .main, .important (применяется ко многим элементам)  
3. Селектор по идентификатору: #header (должен быть уникальным)  
4. Универсальный (\*), комбинированный (div p), псевдоклассы (:hover), псевдоэлементы (::before)

## Примеры CSS-правил

body {  
 background-color: #f0f0f0;  
 font-family: Arial;  
}  
  
h1 {  
 color: navy;  
 text-align: center;  
}  
  
.important {  
 font-weight: bold;  
 color: darkred;  
}

## Блочная модель CSS

Каждый элемент веб-страницы можно представить как прямоугольный блок.  
Основные компоненты блочной модели:  
- \*\*Content\*\* — содержимое (например, текст, изображение)  
- \*\*Padding\*\* — внутренний отступ  
- \*\*Border\*\* — граница вокруг элемента  
- \*\*Margin\*\* — внешний отступ  
  
Эта модель позволяет точно управлять положением и размерами элементов.

## Основные свойства CSS

- color, background-color  
- font-size, font-family, font-weight  
- margin, padding, border  
- width, height  
- display, position, float, clear  
- text-align, vertical-align  
- flex, grid — для современной верстки

## Что такое CSS3

CSS3 — это развитие CSS2, разбитое на модули. Оно добавило множество новых возможностей:  
- Анимации и трансформации (transition, transform)  
- Media-запросы для адаптивности  
- Flexbox и Grid Layout для сложной верстки  
- Поддержка шрифтов @font-face  
- Улучшенные селекторы, тени, градиенты

## Примеры современных HTML+CSS сайтов

Посмотрите примеры на сайтах:  
- https://www.awwwards.com  
- https://css-tricks.com  
- https://dribbble.com  
  
На этих сайтах вы увидите примеры минималистичного дизайна, анимаций, адаптивных сеток. Большинство таких страниц используют CSS3, HTML5 и JavaScript.

## Мини-практика онлайн (30 минут)

Предлагаю прямо сейчас создать HTML-страницу с подключением CSS:  
1. Создайте файл `index.html`  
2. Создайте файл `styles.css`  
3. Подключите CSS к HTML через <link>  
4. Добавьте заголовок и абзац  
5. Задайте стили: цвет, шрифт, отступы  
6. Добавьте класс `.highlight` и примените его к абзацу

## Домашнее задание

1. Оформите свою веб-страницу, применив внешний CSS  
2. Попробуйте использовать классы и ID для стилизации  
3. Используйте хотя бы 5 разных CSS-свойств  
4. По желанию — изучите основы Flexbox или Grid