

ПОЛНАЯ ПРОГРАММА КУРСА И ПРИМЕР УРОКА ↓

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 1-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

Программа курса:

День первый

Основы разметки веб-страниц

- Структура HTML документа
- Первая веб-страница с помощью языка разметки HTML
- Знакомство с редактором Sublime Text3

Результат занятия: создали первую веб-страницу , изучили базовые способы разметки текста путем использования тегов.

Практическое задание: создать веб-страницу с помощью языка разметки HTML.

День второй

Элемент div и атрибуты тегов

- Работа с веб-инспектором
- Создание многостраничного сайта
- Изучение новых элементов и атрибутов тегов

Результат занятия: научились добавлять ссылки и картинки на сайт, добавили атрибуты тегам.

Практическое задание: создать первый многостраничный сайт.

День третий Каскадные таблицы стилей

- Знакомство с каскадными таблицами стилей CSS
- Создание стилей HTML-элементов
- Использование селекторов
- Установка плагина Emmet, работа с ним и добавление текста Lorem

Результат занятия: научились изменять цвет фона и цвет текста HTML-элементам, пользоваться двумя типами селекторов, создавать текст Lorem с помощью плагина Emmet.

Практическое задание: изменить внешний вид страницы, используя CSS свойства.

День четвертый Создание сайта Jaguar

- Создание многостраничного сайта Jaguar
- Создание файловой структуры проекта
- Работа с внешними и внутренними отступами элементов

Результат занятия: научились работать с внешним файлом CSS, познакомились с внешними и внутренними отступами элементов. Практическое задание: создать многостраничный сайт Jaguar.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 2-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

День первый Box model в CSS

- Блочная модель в CSS
- Создание блоков необычной формы
- Работа со свойством box-sizing для изменения способа вычисления ширины и высоты элемента

Результат занятия: научились создавать блоки необычной формы, использовать свойство box-sizing со значением border-box для правильного вычисления ширины элемента и задавать размеры элементам в разных единицах измерения.

Практическое задание: создать карточки с текстом, используя разные значения свойства box-sizing.

День второй

Продвинутая работа с текстом в CSS

- Разные типы шрифтов
- Выбор шрифтов для сайтов
- Создание новых стилей, позволяющих настраивать отображение текста продвинутом уровне

День третий

Результат занятия: создали страницу с использованием разных типов шрифтов, улучшили вид и читаемость текста, используя пройденные свойства. Практическое задание: создать страницу с использованием основных типов шрифтов.

Позиционирование элементов в CSS

- Работа со свойством float
- Типы позиционирования элементов на странице
- Свойство position

Результат занятия: научились работать со свойствами float и position, создавать обтекание картинок текстом.

Практическое задание: создать веб-страницу и разместить на ней элементы, используя свойства float и position.

День четвертый Создание страницы блога

- Создание страницы блога
- Свойства позиционирования элементов для размещения меню и постов на странице
- Семантические теги

Результат занятия: научились работать с семантическими тегами, используя свойства позиционирования, расставили основные компоненты блога на странице.

Практическое задание: создать страницу блога, используя семантические теги.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 3-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

День первый Псевдоклассы и работа с изображениями в CSS

- Использование фоновых картинок на веб-странице
- Псевдоклассы hover, active и visited
- Изменение стиля одного элемента при активации псевдокласса у другого элемента
- Создание background patterns

Результат занятия: научились научились работать с фоновыми картинками, изучили свойства для работы с картинками.

Практическое задание: добавить внешние изменения элементам при наведении на них курсора мыши.

День второй

FlexBox в CSS

- Работа с CSS Flexible Box Layout
- Свойства для выравнивания элементов в Flex контейнере
- Обучающая игра Flex-frog

Результат занятия: научились работать с технологией Flex для создания гибких макетов, прошли игру Flex-frog для закрепления пройденного материала. Практическое задание: пройти игру Flex-frog для закрепления пройденного материала.

День третий

Разработка сайта Sneakers shop . Часть 1

- Создание структуры сайта с моделями
- Подключение к сайту шрифта
- Свойства шапки сайта и ее дочерних элементов
- Функции для создания градиентного фона страницы

Результат занятия: создали структуру сайта с моделями, подключили шрифт к сайту.

Практическое задание: выбрать и скачать картинки с моделями кроссовок, добавить тень шапке сайта.

День четвертый Разработка сайта Sneakers shop . Часть 2

- Завершение работы над сайтом
- Создание блока с карточками моделей
- Изменение поведения элементов при наведении на них
- Технология FlexBox для позиционирования карточек

Результат занятия: создали сайт с карточками моделей.

Практическое задание: добавить галерею, используя технологию FlexBox.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 3-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

День первый Псевдоклассы и работа с изображениями в CSS

- Использование фоновых картинок на веб-странице
- Псевдоклассы hover, active и visited
- Изменение стиля одного элемента при активации псевдокласса у другого элемента
- Создание background patterns

Результат занятия: научились научились работать с фоновыми картинками, изучили свойства для работы с картинками.

Практическое задание: добавить внешние изменения элементам при наведении на них курсора мыши.

День второй

FlexBox в CSS

- Работа с CSS Flexible Box Layout
- Свойства для выравнивания элементов в Flex контейнере
- Обучающая игра Flex-frog

Результат занятия: научились работать с технологией Flex для создания гибких макетов, прошли игру Flex-frog для закрепления пройденного материала. Практическое задание: пройти игру Flex-frog для закрепления пройденного материала.

День третий

Разработка сайта Sneakers shop . Часть 1

- Создание структуры сайта с моделями
- Подключение к сайту шрифта
- Свойства шапки сайта и ее дочерних элементов
- Функции для создания градиентного фона страницы

Результат занятия: создали структуру сайта с моделями, подключили шрифт к сайту.

Практическое задание: выбрать и скачать картинки с моделями кроссовок, добавить тень шапке сайта.

День четвертый Разработка сайта Sneakers shop . Часть 2

- Завершение работы над сайтом
- Создание блока с карточками моделей
- Изменение поведения элементов при наведении на них
- Технология FlexBox для позиционирования карточек

Результат занятия: создали сайт с карточками моделей.

Практическое задание: добавить галерею, используя технологию FlexBox.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 4-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик
Программа курса:

День первый Grid Layout в CSS

- Знакомство с Grid системой
- Создание страницы с использованием Grid сетки
- Свойства для продвинутой работы с Grid ячейками

Результат занятия: изучили систему двумерного макета (CSS Grid Layout), научились позиционировать Grid ячейки на странице. Практическое задание: пройти игру Grid garden для закрепления пройденного материала.

День второй

Псевдоэлементы after и before

- Псевдоэлементы before и after
- Комбинирование картинки с текстом
- Элементы с псевдоэлементами first-letter и first-line

Результат занятия: научились работать с псевдоэлементами before, after, first-letter и first-line, комбинировать псевдоэлементы с разными свойствами для создания красивых блоков.

Практическое задание: создать элемент с использованием псевдоэлементов after и before.

День третий

Анимации и трансформации в CSS

- CSS трансформации
- Применение трансформаций к HTML-элементам
- Создание анимированных блоков в CSS
- Применение временных функций для ключевых кадров

Результат занятия: научились работать с трансформациями в CSS, создавать бесконечные анимации в CSS.

Практическое задание: создать кнопку, добавить ей трансформации при наведении.

День четвертый Практика в создании анимаций в CSS

- Применение анимации и трансформации на практике
- Создание анимированной орбитальной системы планет нашей Солнечной системы

Результат занятия: создали страницу с демонстрацией движения планет в Солнечной системе и анимированный фон.

Практическое задание: создать анимированную орбитальную систему планет нашей солнечной системы.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 5-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик
Программа курса:

День первый

Создание интернет магазина

- Что такое «интернет магазин»?
- Что такое «глобальные переменные» в CSS ?
- Создание структуры проекта
- Настройка проекта

Результат занятия: начали создавать интернет магазин. Практическое задание: выбрать иконки, которые будут использоваться на сайте, создать переменные.

День второй

Создание шапки интернет-магазина

- Переменные в CSS и способы их использования
- Подключение шрифтов во внешнюю таблицу стилей, используя функцию url
- Шапка интернет-магазина, ее стилизация
- Добавление дополнительной функциональности меню с помощью языка программирования JavaScript

Результат занятия: настроили глобальные переменные в проекте, создали многофункциональную шапку с меню, подключили Javascript файл для добавления дополнительной функциональности меню.

Практическое задание: добавить изменение фона шапки сайта при прокрутке страницы.

День третий Создание первого экрана с главным продуктом

- HTML-стрutura презентационного блока
- Grid система для правильного отображения элементов
- Значения CSS свойств с помощью переменных
- Стилизация элементов

Результат занятия: создали первый экран с главным продуктом. Практическое задание: создать презентационную часть интернет -магазина.

День четвертый Создание блока с популярными товарами

- HTML-разметка для блока с товарами
- Стилизация для карточек товаров
- Grid-сетка для позиционирования карточек
- Стилизация блока с историей компании

Результат занятия: создали блок с популярными товарами и историей компании.

Практическое задание: добавить внешние и внутренние отступы элементам в блоке с историей компании.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 5-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик
Программа курса:

День первый

Создание интернет магазина

- Что такое «интернет магазин»?
- Что такое «глобальные переменные» в CSS ?
- Создание структуры проекта
- Настройка проекта

Результат занятия: начали создавать интернет интернет-магазин. Практическое задание: выбрать иконки, которые будут использоваться на сайте, создать переменные.

День второй

Создание шапки интернет-магазина

- Переменные в CSS и способы их использования
- Подключение шрифтов во внешнюю таблицу стилей, используя функцию url
- Шапка интернет-магазина, ее стилизация
- Добавление дополнительной функциональности меню с помощью языка программирования JavaScript

Результат занятия: настроили глобальные переменные в проекте, создали многофункциональную шапку с меню, подключили Javascript файл для добавления дополнительной функциональности меню.

Практическое задание: добавить изменение фона шапки сайта при прокрутке страницы.

День третий Создание первого экрана с главным продуктом

- HTML-структура презентационного блока
- Grid система для правильного отображения элементов
- Значения CSS свойств с помощью переменных
- Стилизация элементов

Результат занятия: создали первый экран с главным продуктом. Практическое задание: создать презентационную часть интернет -магазина.

День четвертый Создание блока с популярными товарами

- HTML-разметка для блока с товарами
- Стилизация для карточек товаров
- Grid-сетка для позиционирования карточек
- Стилизация блока с историей компании

Результат занятия: создали блок с популярными товарами и историей компании.

Практическое задание: добавить внешние и внутренние отступы элементам в блоке с историей компании.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 6-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

День первый Создание слайдера с отзывами покупателей

- Блок с отзывами
- Способы создания прокручиваемых контейнеров в CSS
- Свойства scroll-behavior и scroll-snap-type
- Точки привязки в слайдере

Результат занятия: создали слайдер с отзывами клиентов с помощью HTML и CSS.

Практическое задание: добавить точки привязки основным секциям сайта и создать автоматический scroll до этих блоков при нажатии на кнопку.

День второй

Работа с формами и видео-контентом

- HTML5 теги для вставки видео-контента на сайт
- Стилизация секции с полноэкранным видео роликом
- Тег поля ввода input и его атрибуты
- Тег для создания форм в HTML
- Стилизация элемента ввода текста

Результат занятия: создали секцию с видео и блок с формой обратной связи. Практическое задание: создать форму обратной связи, создать стилизацию элементов.

День третий

Создание отдельной страницы с галереей

- Нестандартная Grid сетка для страницы с галереей
- Добавление в сетку блоков с изображениями

- Псевдоэлементы и разные анимации блоков с изображениями
 - CSS фильтры для улучшения визуального дизайна галереи
- Результат занятия: создали отдельную страницу с фото галереей. Практическое задание: с помощью разных типов позиционирования создать полупрозрачный необычный фон.

День четвертый Адаптация контента под все виды устройств

- Способы адаптирования контента под мобильные устройства
 - Media queries для применения различных свойств одному элементу на устройствах с разной шириной
 - CSS правила для трех видов устройств
- Результат занятия: научились адаптировать сайт под планшеты и телефоны.
- Практическое задание: используя media запросы, изменить значения свойств к HTML-элементам для адаптации контента под все виды устройств.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 7-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик

Программа курса:

День первый Тестирование сайта

- Проведение тестирования сайта на разные показатели
- Использование основных инструментов для оценки скорости работы сайта
- Оптимизация картинок для сайта

Результат занятия: научились тестировать сайты на различные показатели. Практическое задание: провести тестирование сайта с помощью различных инструментов.

День второй

Знакомство с фреймворком Tailwind GSS.

- Что такое фреймворки, и как они ускоряют процесс разработки проекта
- Установка фреймворка Tailwind CSS и плагинов
- Создание карточки с помощью Tailwind CSS

Результат занятия: создали первую веб-страницу с использованием фреймворка Tailwind CSS.

Практическое задание: провести эксперимент по созданию карточки с помощью простого CSS и с помощью Tailwind CSS.

День третий

Tailwind классы для создания адаптивного сайта

- Работа с Tailwind классами для создания адаптивного сайта
 - Изучение понятия «адаптивности сайта»
 - Работа с Flex в Tailwind
 - Работа с Grid в Tailwind
- Результат занятия: создали адаптивную страницу под все виды.
- Практическое задание: создать кнопки, добавить эффекты при нажатии на них с помощью классов Tailwind.

День четвертый Grid система в Tailwind CSS.

- Работа с Grid системой в Tailwind CSS
- Свойства элементов адаптивной страницы сайта в Grid сетке
- Страница с отзывами

Результат занятия: создали адаптивную страницу с сеткой блоков с использованием системы Tailwind Grid.

Практическое задание: добавить grid элементы с отзывами на адаптивную страницу с помощью Tailwind CSS.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 8-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик
Программа курса:

День первый Настройка проекта Manage

- Настройка проекта сайта-приложения Manage
- Создание адаптивной шапки сайта
- Создание меню Hamburger с использованием JavaScript
- Создание презентационной секции с продуктом

Результат занятия: создали первый экран сайта-приложения Manage. Практическое задание: добавить фоновую картинку первому экрану.

День второй

Создание секции с описанием приложения Manage

- Секция с описанием приложения и его преимуществами
- Адаптировать созданный блок под все виды устройств
- Секция с работой приложения
- Применить Tailwind фильтры к изображению

Результат занятия: создали две секции сайта Manage. Практическое задание: создать дополнительную секцию с историей создания приложения.

День третий

Создание секции FAQ для сайта-приложения Manage

- Работа над секцией FAQ
- Добавление анимации и функционала collapse в секцию FAQ
- Добавление интерактивных блоков

Результат занятия: создали секцию FAQ для сайта-приложения Manage. Практическое задание: создать секцию FAQ и добавить интерактивные блоки.

День четвертый Создание footer'a сайта-приложения Manage

- Создание адаптивного footer сайта
- Создать блок со значками социальных сетей и формой обратной связи
- Доработка проекта сайта-приложения Manage

- Размещение сайта в интернет

Результат занятия: создали footer сайта-приложения Manage. Практическое задание: создать footer, выложить сайт в интернет.

Курс «Frontend-разработчик: сайты на HTML/CSS/Javascript». 9-й модуль

Цель курса: пройти путь от основ HTML и CSS до собственного современного сайта, осваивая навыки веб-программирования и формируя представление о профессии веб-разработчик
Программа курса:

День первый Создание первого экрана собственного сайта

- Выбор темы для итогового проекта
- Выбор цветовой палитры для сайта
- Создание первого экрана сайта

Результат занятия: выбрали цветовую палитру для сайта, создали первый экран сайта.

Практическое задание: создание первого экрана.

День второй

Создание блока с карточками для собственного проекта

- Создание Feature секции
- Создание блока с карточками
- Использование видео-контента и эффектов для Feature секции

Результат занятия: создали Feature секцию. Практическое задание: создать Feature секцию и слайдер.

День третий

Создание footer■а и СТА для собственного проекта

- Создание подвала сайта
- Создание блока с СТА
- Итоговое тестирование и подготовка к защите проектов

Результат занятия: создали footer сайта и блок СТА, прошли итоговое тестирование.

Практическое задание: создать footer сайта и блок СТА, подготовить презентацию проекта, выложить сайт в интернет.

День четвертый Защита проекта

- Репетиция защиты проектов и доработка презентации
- Защита проектов перед аудиторией

Результат занятия: презентовали проект.

Практическое задание: доработать презентацию, защитить проект.

ПРИМЕР УРОКА ↓

Занятие 7

Позиционирование элементов в CSS

Цель занятия

- Научиться работать со свойством float
- Изучить разные типы позиционирования элементов на странице
- Познакомиться со свойством position

Результат занятия

- Научились работать со свойствами float и position
- Научились создавать обтекание картинок текстом

План занятия

1. Начало занятия и создание документа (20 мин)	2
2. Обтекание текстом (25 мин)	4
3. Позиционирование в CSS (45 мин)	7
3.1 Что такое позиционирование	7
3.2 Относительное позиционирование	8
3.3 Абсолютное позиционирование	11
3.4 Фиксированное позиционирование	15
3.5 Свойство z-index	16
3.6 Липкие элементы	20
4. Самостоятельная работа (20 мин)	22
5. Итоги занятия. Домашнее задание (10 мин)	23

1. Начало занятия и создание документа (20 мин)

В начале занятия нужно проверить домашнюю работу ребят и, если у них возникли трудности с чем-то, помочь им сделать задание, вместе повторить теорию.

Далее нам нужно создать файл fonts.html в папке "обучение".

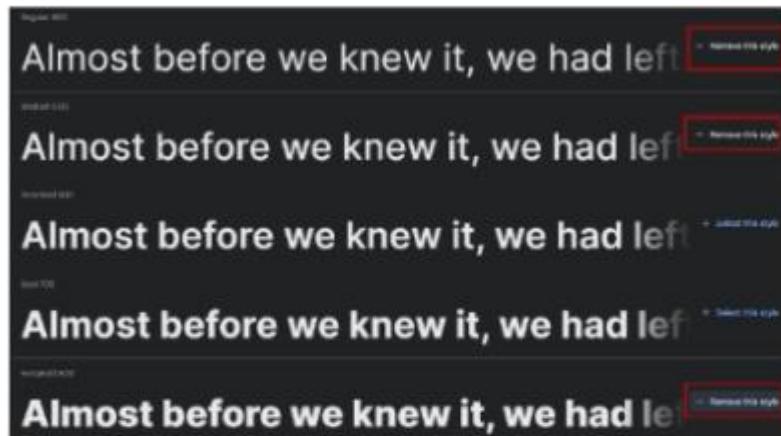
После чего открываем файл fonts.html, а начале документа ставим "?" и нажимаем Tab. Меняем содержимое тега <title> на "Position CSS".

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Position CSS</title>
</head>
<body>
</body>
</html>
```

Добавим тег <style> в котором будем писать код CSS. Используя универсальный селектор, обнулим отступы у элементов и добавим свойство box-sizing:

```
7     <style>
8         * {
9             margin: 0;
10            padding: 0;
11            box-sizing: border-box;
12        }
13    </style>
```

Зайдем на сайт [Google Fonts: Browse Fonts](#) и добавим три стиля (400, 500, 800) шрифта Inter:



Скопируем теги <link> и вставим их в тег <head>:

```
7   <link rel="preconnect" href="https://fonts.googleapis.com">
8   <link rel="preconnect" href="https://fonts.gstatic.com"
9     crossorigin>
9   |<link href="https://fonts.googleapis.com/css2?
family=Inter:wght@400;500;800&display=swap" rel="stylesheet">
```

Добавим свойство font-family и минимальное значение высоты элементу <body>:

```
body {
    font-family: 'Inter', sans-serif;
    min-height: 2000px;
}
```

2. Обтекание текстом (25 мин)

Float - это свойство CSS для позиционирования элементов.

Для того чтобы понять его назначение и происхождение, можно обратиться к полиграфическому дизайну. В компоновке при печати изображения могут располагаться на странице так, чтобы текст "обтекал" их. Обычно это так и называется: "обтекание текстом".



В веб-дизайне элементы страницы, с установленным свойством float, ведут себя точно так же, как и изображения в полиграфии, когда текст "обтекает" их.

Синтаксис: `float: left/right/clear`

Добавим обтекание изображений текстом на страницу.

Для этого создадим контейнер `<div>` с классом `float` и запишем в него два тега `` и пару абзацев `<p>` с классом `p-img` в текстом Lorem из 500 символов в файл `position.html`:

```
8 <body>
9     <div class="float">
10        <img src="" alt="">
11        <p>Lorem500</p>
12        <img src="" alt="">
13        <p>Lorem500</p>
```

```
<body>
<div class="float">
    <img src="" alt="">
    <p>Lorem ipsum dolor sit amet, consectetur
    adipisicing elit. Repellat eos eius, pariatur!
    Molestiae dolor, at sint laborum id, iure
    obcaecati ducimus sapiente saepe voluptatem.
    Adipisci placeat labore harum consectetur
    recusandae corrupti ad saepe ullam dolores
```

В браузере найдем две небольших картинки (ширина до 500px). И установим их в папку "обучение". Называем первую left, вторую right:

```
<> index3.html
 left.png
 right.png
<> position.html
```

В атрибуте src укажем путь до скачанных изображений и добавим тегам имена классов left и right, а абзацам класс p-img:

```
<body>
  <div class="float">
    
    <p>Lorem ipsum dolor sit amet, consectetur
      adipisicing elit. Quod laboriosam deserunt nulla maxime quibusdam
```

Добавим внешний отступ и свойство float двум изображениям, а также увеличим размер шрифта абзацам p-img:

```
13   .left {
14     margin: 30px;
15     float: left;
16   }
17   .right {
18     margin: 30px;
19     float: right;
20   }
21   .p-img {
22     font-size: 20px;
23 }
```

Результат на странице:



3. Позиционирование в CSS (45 мин)

3.1 Что такое позиционирование

Вся идея позиционирования заключается в том, чтобы позволить нам переопределять поведение базового потока документа.

Что если вам захочется слегка изменить позицию каких-либо блоков внутри макета относительно их позиции в потоке макета по умолчанию?

Ваш инструмент - позиционирование. Вы можете создать элемент пользовательского интерфейса, который плавает над другими частями страницы и/или всегда располагается на одном и том же месте в окне браузера, независимо от того, сколько прокручивалась страница.

Модуль позиционирования описывает, как любой из элементов в HTML-документе может быть размещен на странице (независимо от других элементов)

Расположение блоков в документе регулируется:

- размерами и типом элемента
- схемой позиционирования (нормальный поток, обтекание и абсолютное позиционирование)
- отношениями между элементами в дереве документа
- внешней информацией (например, размер области просмотра, внутренними размерами изображений и т.д.).

Существует несколько разных типов позиционирования, которые вы можете применить к HTML-элементам.

Для добавления специфического типа позиционирования у элемента, мы используем свойство [position](#).

Свойство position в CSS может принимать пять значений:

- static (статичное позиционирование)
- relative (относительное)
- absolute (абсолютное)
- fixed (фиксированное)
- sticky (липкое).

Static - это значение по умолчанию. Оно означает, что элемент находится в базовом потоке.

3.2 Относительное позиционирование

Относительно позиционированный элемент ведет себя как обычный элемент в потоке, за исключением того, что его текущее положение можно изменить при помощи определенных CSS свойств.

К этим CSS свойствам относятся: left, top, right и bottom.

Добавим свойство display со значением none контейнеру float, чтобы мы могли начать работу с чистого листа:

```
24         .float {  
25             display: none;  
26         </style>  
27     </head>  
28     <body>  
29         <div class="float">
```

Создадим контейнер div с классом relative, в него поместим тег <h1>:

```
94      <div class="float">=
99      </div>
100     <div class="relative">
101       <h1>CSS Position</h1>
102     </div>
```

Зададим ширину элементу relative в 500px и поставим его в центр (margin: auto).
Добавим свойство position со значением relative и свойство top со значением в 300px,
чтобы сдвинуть элемент вниз.

```
.relative {
    width: 500px;
    margin: auto;
    position: relative;
    top: 300px;
}
```

Результат на странице:



Добавим заголовку свойства: font-size, font-weight, text-align

```
h1 {  
    font-size: 64px;  
    font-weight: 800;  
    text-align: center;  
}
```

Результат на странице:



3.3 Абсолютное позиционирование

Абсолютное позиционирование удаляет элемент из общего потока документа.

Что касается элементов вокруг, то в этом случае они просто игнорируют искомый, как будто ему установлено свойство `display: none`.

Позиционирование выполняется относительно ближайшего позиционированного предка.

Ключевой момент в абсолютном позиционировании - это понимание того, что является точкой отсчета. Если свойству `top` указано значение `20px`, то откуда их необходимо отсчитывать?

Ответ прост: такие элементы позиционируются относительно ближайшего родительского элемента, для которого задано позиционирование, отличное от `static`.

Если такого элемента нет, то элемент позиционируется относительно основного документа. То есть при установке абсолютного позиционирования, CSS сообщает браузеру, чтобы он посмотрел на родительский элемент и, если его позиционирование не статическое, то надо выровнять текущий элемент относительно него.

Добавим свойство position со значением relative тегу body:

```
body {  
    font-family: 'Inter', sans-serif;  
    min-height: 2000px;  
    position: relative;  
}
```

Теперь создадим два дочерних контейнера тега <body>. Зададим им классы circle-green и circle-blue:

```
55      <div class="relative">  
56          <h1>CSS Position</h1>  
57      </div>  
58      <div class="circle-green"></div>  
59      <div class="circle-blue"></div>
```

Добавим элементу circle-green свойство position absolute со значением absolute, фон, свойство border-radius, ширину и минимальную высоту (min-height).

Также сдвинем элемент вправо, используя свойства top и right с отрицательным значением, чтобы блок выехал за пределы экрана:

```
.circle-green {  
    position: absolute;  
    background-color: #ADFCBA;  
    width: 600px;  
    min-height: 600px;  
    border-radius: 50%;  
    right: -300px;  
    top: 200px;  
}
```

Результат на странице



Снизу у нас появилась прокрутка.

Чтобы ее убрать, тегу html нужно добавить свойство overflow-x со значением hidden:

```
html {  
    overflow-x: hidden;  
}
```

Зададим стилизацию для элемента circle-blue.

Используем те же свойства, но вместо right и top, используем top и left, чтобы сдвинуть блок в верхний левый угол:

```
.circle-blue {  
    position: absolute;  
    background-color: #ADE9FC;  
    width: 800px;  
    min-height: 800px;  
    border-radius: 50%;  
    left: -400px;  
    top: -400px;  
}
```

Результат на странице:



3.4 Фиксированное позиционирование

Фиксированное позиционирование действует подобно абсолютному, с небольшими различиями.

Во-первых, элемент с фиксированным позиционированием всегда располагается относительно окна браузера, родительские элементы при этом игнорируются.

Второе отличие исходит из его названия. Фиксированные элементы зафиксированы на странице. Они не смещаются при её прокручивании.

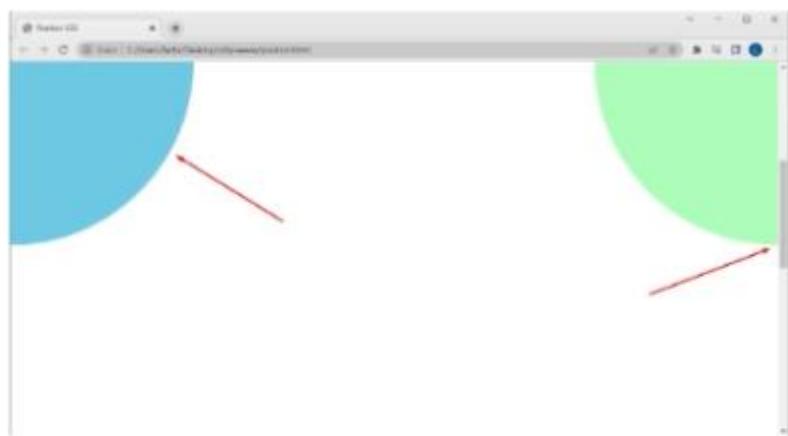
Создадим еще один контейнер с классом circle-dark:

```
<div class="circle-green"></div>
<div class="circle-blue"></div>
<div class="circle-dark"></div>
```

Добавим ему свойство position со значением fixed и свойства предыдущих контейнеров:

```
.circle-dark{
    position: fixed;
    background-color: #6DC8E4;
    width: 600px;
    min-height: 600px;
    border-radius: 50%;
    left: -300px;
    top: -300px;
}
```

Результат на странице:



3.5 Свойство z-index

Свойство z-index определяет положение элемента и ныкестоящих элементов по оси Z.

В случае перекрытия элементов, это значение определяет порядок наложения.

В общем случае, элементы с большим z-index перекрывают элементы с меньшим.

Если мы контейнеру circle-dark добавим свойство z-index со значением 0, а контейнеру circle-blue z-index равный 1, то первый контейнер не будет видно, так как второй перекрыл его, выехав вперед:

```
.circle-blue {
    position: absolute;
    background-color: #ADE9FC;
    width: 800px;
    min-height: 800px;
    border-radius: 50%;
    left: -400px;
    top: -400px;
    z-index: 1;
}

.circle-dark{
    position: fixed;
    background-color: #6DC8E4;
    width: 600px;
    min-height: 600px;
    border-radius: 50%;
    left: -300px;
    top: -300px;
    z-index: 0;
}
```



Контейнеру circle-blue изменим значение свойства z-index на 10:

```
.circle-dark{  
    position: fixed;  
    background-color: #6DC8E4;  
    width: 600px;  
    min-height: 600px;  
    border-radius: 50%;  
    left: -300px;  
    top: -300px;  
    z-index: 10;  
}
```

3.6 Липкие элементы

Липкие элементы (CSS position sticky) очень похожи на фиксированные (position: fixed), поскольку они сохраняют свое положение на экране, даже когда пользователь прокручивает страницу вверх или вниз.

Но липкий элемент остается ограниченным родительским контейнером, в котором он находится.

Интересное и общее использование position: sticky заключается в создании индексных страниц с прокруткой, где разные заголовки липнут к верху страницы, когда они достигают его.

Создадим два тега <p> в контейнере relative с классами sticky и text. Второму абзацу добавим текст Lorem:

```
<div class="relative">
  <h1>CSS Position</h1>
  <p class="sticky">Sticky title</p>
  <p class="text">Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi veritatis optio, cumque. Veritatis reiciendis, sapiente harum, exercitationem, beatae voluptatum voluptates officia nemo neque architecto culpa. Quod, provident, nobis? Atque neque, culpa. Quod beatae ipsam cumque odio laudantium nemo similique quam eos laborum sapiente accusamus harum, quae, neque quas, recusandae provident explicabo ducimus tempore esse maiores illum a. Corporis illo natus ipsum vero commodi ducimus adipisci id quod minima magnam, eveniet fugit, sunt provident aut possimus iste inventore dolores quibusdam non distinctio corrupti. Fugiat saepe, officia ipsam similique, nam ut voluptatum reiciendis laboriosam, odio excepturi sit cum qui. Ratione, maiores, at!</p>
```

Элемент будет становиться плавающим, как только область видимости достигнет определенной позиции, например, top: -1px.

Элементу sticky добавим свойства font-size, font-weight, text-align, margin-top и свойство position со значением sticky:

```
.sticky {  
    font-size: 24px;  
    font-weight: 500;  
    text-align: center;  
    margin-top: 20px;  
    position: sticky;  
    top: -1px;  
}
```

Увеличим верхний внешний отступ элементу text:

```
.text {  
    margin-top: 14px;  
}
```

Как только box-model элемента sticky выйдет за пределы области видимости страницы на 1px, текст останется сверху, пока мы не пролистаем весь родительский элемент relative.

Результат на странице:



При прокрутке страницы можно увидеть разницу между фиксированным элементом и липким:



4. Самостоятельная работа (20 мин)

Пример заданий:

- Создать контейнер <div> с шириной в 100% и максимальной высотой (max-height) 1000 пикселей.
- Поместить в него пять тегов div. Задать им цвет фона, отступы, ширину и минимальную высоту в 200px. Добавить свойство position с разными значениями и посмотреть, как будут двигаться блоки при прокрутке страницы.

5. Итоги занятия. Домашнее задание (10 мин)

Подведите итоги занятия. Еще раз напомните ребятам темы, которые прошли на сегодняшнем занятии. Ответьте на возможные вопросы.

Вопросы для самопроверки:

1. Что такое абсолютное позиционирование?
2. Что такое относительное позиционирование?
3. Для чего используется свойство float?
4. Как создать липкий элемент?

Как можно подвести итоги урока - пример:

"Сегодня на уроке мы узнали, как установить , как устроено наше рабочее окно, как создать... что такое.... Также начали создавать Все это нам пригодится на следующих занятиях.

Что сегодня было для вас самым интересным? Что показалось самым сложным?
Какие есть вопросы?"

Выслушайте ответы детей.

"Отдельное спасибо Сереже за то, что он быстро отвечал на вопросы и старался помочь нам на уроке. Так же хотелось бы похвалить Мишу, за то, что он был внимателен еще занятие, и задавал хорошие вопросы. Вы все сегодня замечательные молодцы!"

Объясните домашнее задание, которое сегодня состоит в следующем:

Создать необычную композицию из разных блоков, а именно:

- Создать несколько контейнеров <div> , задать им фон, border-radius, border, ширину и минимальную высоту
- Используя свойства позиционирования, поместить их в разные места веб-страницы