

Курс Web- программирование

Интернет и Web-сайты



Практическое занятие 1.2 / Самостоятельная работа

- 1) Изучить Классификация сайтов. Этапы создания.
https://miro.com/app/board/o9J_kxr0yg4=/
- 2) подключиться к веб-сайтам (например, rambler.ru, vk.com, yandex.ru и т.п.) через веб-браузер;
- 3) получить IP-адрес веб-сайтов по доменному имени. [Как узнать?](#)

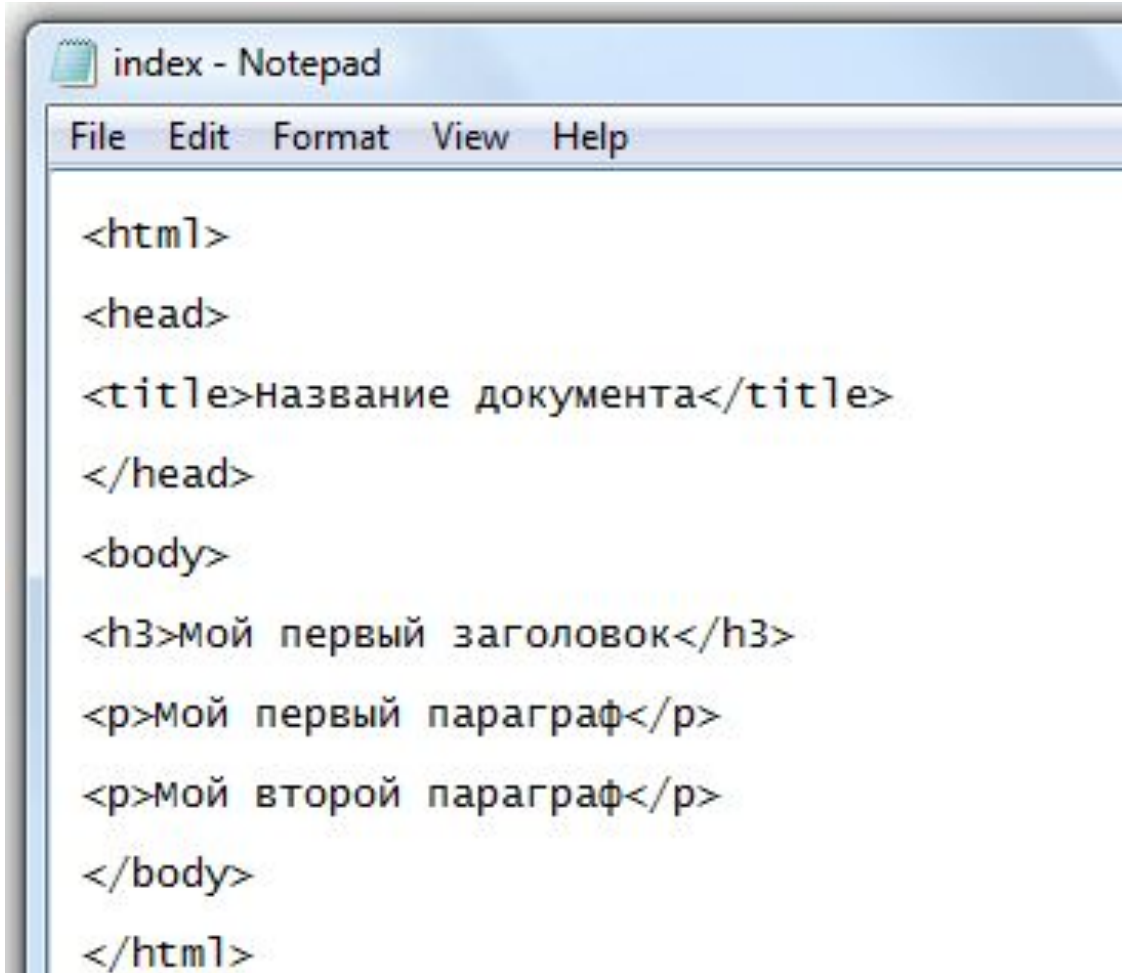
Клиентская часть веб-приложений - HTML

HTML — язык разметки, который используется в большинстве веб-страниц. Браузер интерпретирует его значения (теги) и отображает текст и элементы так, как их представляли разработчики. Без стандартизированного языка страницы сайта превращались бы в мешанину из букв, картинок и форм, которой невозможно нормально пользоваться.

Несмотря на то, что сейчас можно создавать отличные сайты на конструкторах в визуальном режиме, использование языка HTML остаётся важным навыком разработчика. Даже если вы не пишете шаблон с нуля, может возникнуть необходимость отредактировать его или вставить виджет стороннего сервиса — через HTML-код.

HTML позволяет форматировать страницы, придавая им необходимый вид. Если такой задачи нет, то знание тегов поможет отредактировать код встраиваемого виджета. Понимая структуру HTML, вы, например, можете быстро изменить размер видео или виджета и другие особенности его отображения.

Структура HTML документа

A screenshot of a Notepad window titled "index - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following HTML code:

```
<html>
<head>
<title>название документа</title>
</head>
<body>
<h3>мой первый заголовок</h3>
<p>мой первый параграф</p>
<p>мой второй параграф</p>
</body>
</html>
```

```
<html>
<head>
<title>HTML структура
документа</title>
</head>
<body>
<h3>Мой первый заголовок</h3>
<p>Мой первый параграф</p>
<p>Мой второй параграф</p>
</body>
</html>
```

Практическое занятие 2.1 / Самостоятельная работа

Клиентская часть веб-приложений - HTML

- 1) Создать html документ
- 2) Создать в нем текстовый блок, где будет:
 - Заголовки разных типов;
 - Жирный, курсивный, подчеркнутый текст;
 - Список с минимум 3-мя элементами (1-й элемент перечеркнут, 2-)
 - 1-й элемент перечеркнут
 - 2-й выделен желтым
 - 3-й с использование подстрочного и надстрочного начертания символов
 - Изображение и аудио
 - Кнопка
 - Ссылка
- 3) Открыть созданный html файл в браузере

**Основные теги вы можете посмотреть [здесь](#), а все теги [здесь](#)*

Каскадная таблица стилей - CSS

CSS (Cascading Style Sheets) — это код, который вы используете для стилизации вашей веб-страницы.

Как и HTML, CSS на самом деле не является языком программирования. Это не язык разметки - это *язык таблицы стилей*. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML. Например, чтобы выбрать **все** элементы абзаца на HTML странице и изменить текст внутри них с черного на красный, вы должны написать этот CSS:

```
p { color: red; }
```

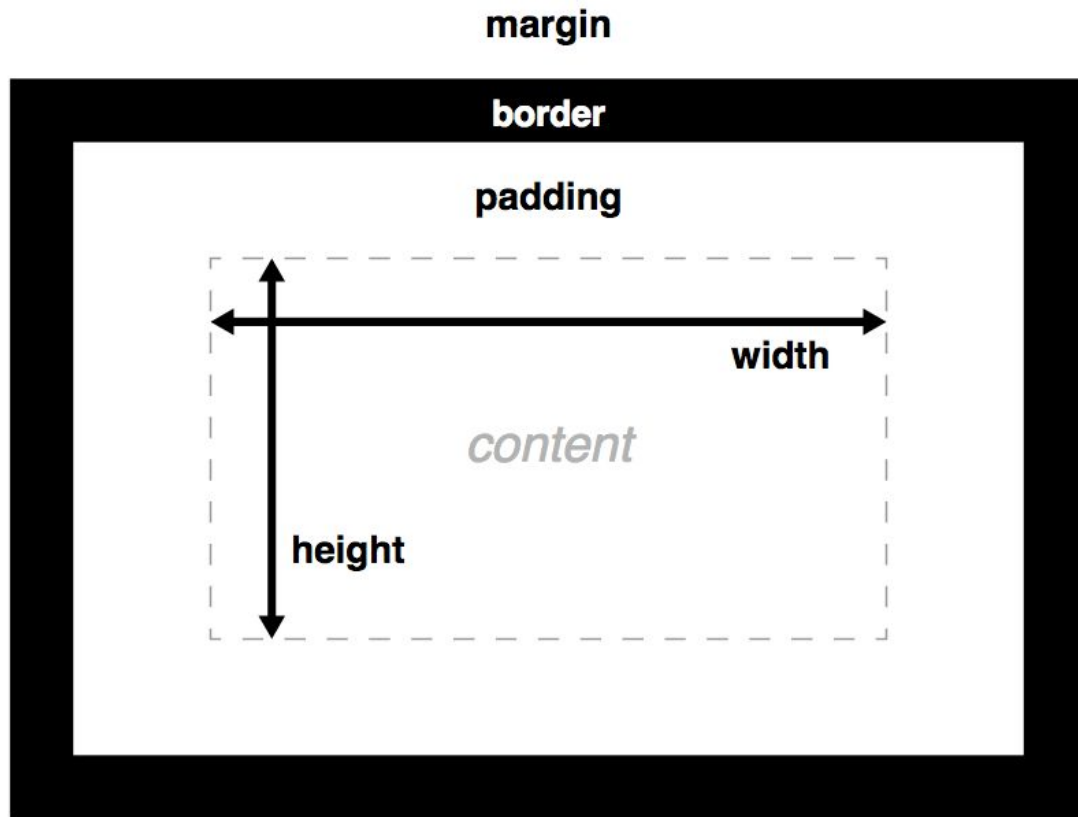
Практическое занятие 2.2 / Самостоятельная работа

- **В ходе выполнения Практической работы №2.2 необходимо:**
- 1) Ознакомиться с теоретическим материалом.
- 2) Создать HTML-страницу по образцу и применить к ней стили CSS
(http://users.uoa.gr/~nektar/arts/tributes/antoine_de_saint-exupery_le_petit_prince/the_little_prince.htm)

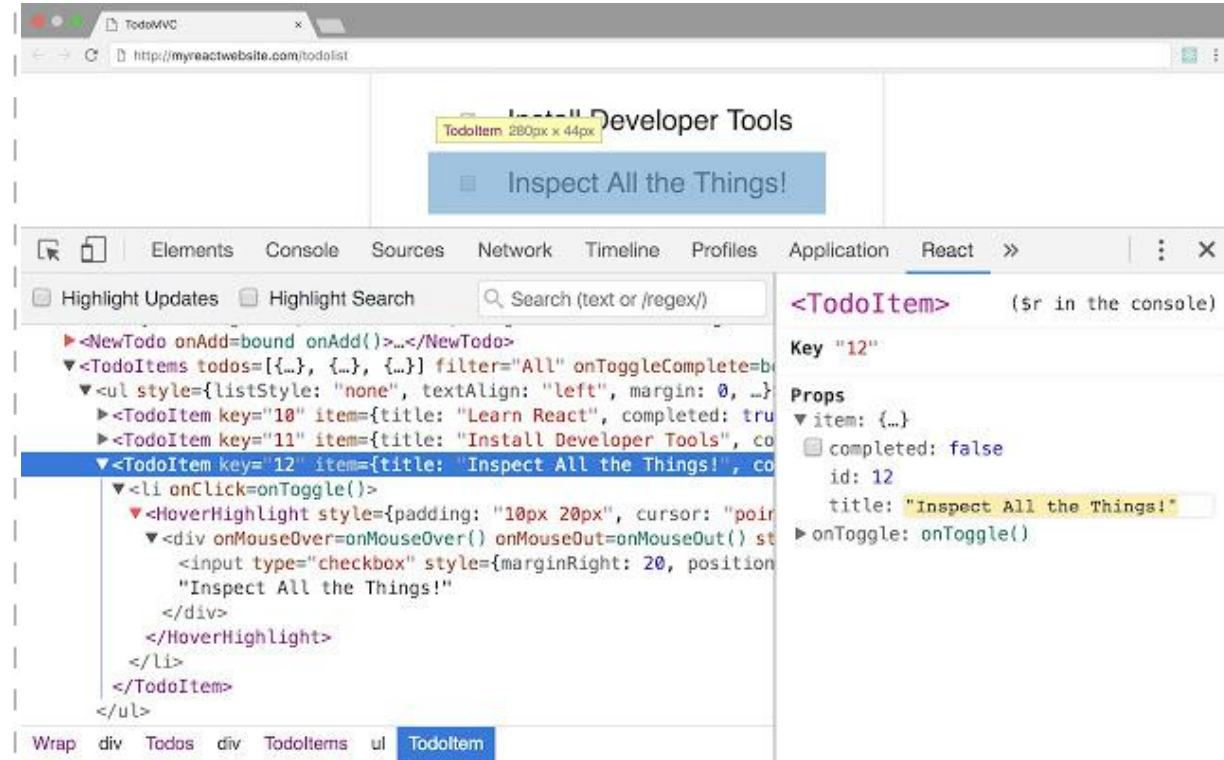
[CSS справочник](#)

[Способы подключения CSS к странице](#)

Блочная модель CSS



Developer Tool. Что это и как с ним работать?



Практическое занятие 2.3 / Самостоятельная работа

1) Создать HTML-страницу по образцу и применить к ней стили CSS используя блочную вёрстку

Пример 1

Создание сайтов	Продвижение сайтов	Монетизация	Полезное
С чего начать	С чего начать?	Заработок в интернете	Уроки фотошоп
Сайт на HTML, CSS	Ключевые слова	Заработок на сайте	Транслитерация
Сайт на Joomla	Внутренняя оптимизация	Фриланс	Счетчик знаков
Сайт на Wordpress	Поведенческие факторы	Партнерские программы	Инструменты
Конструкторы сайтов	Реклама в интернете	Бизнес в интернете	Рекомендую

Практическое занятие 2.3 / Самостоятельная работа

Пример



CSS-позиционирование

[Изучите теорию](#)

```
Заголовок {position: absolute; top: 10px;}
```

```
Заголовок {position: static}
```

Заголовок с `position static` сохраняет своё место в потоке документа

```
Заголовок {position: relative; top: 10px;}
```

Заголовок с `position relative` смещается относительно своего текущего положения в макете

Заголовок с `position absolute` смещается относительно верхнего левого угла браузера. При этом его ширина становится равной ширине содержимого

Практическое занятие 2.4 / Повторите страницы по данному

1



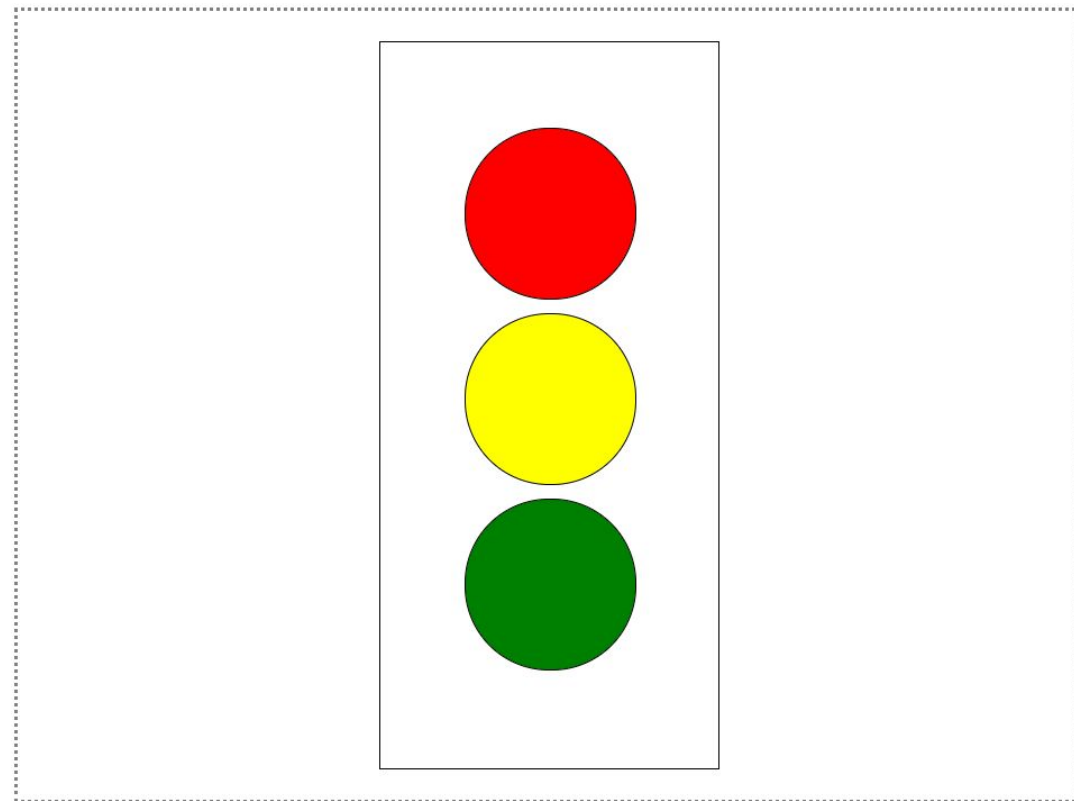
2



3



4



5



Каскадность в CSS

Аббревиатура CSS расшифровывается как Cascading Style Sheets (каскадные таблицы стилей), где одним из ключевых слов выступает «каскад». Под каскадом в данном случае понимается одновременное применение разных стилевых правил к элементам документа — с помощью подключения нескольких стилевых файлов, наследования свойств и других методов. Чтобы в подобной ситуации браузер понимал, какое в итоге правило применять к элементу, и не возникало конфликтов в поведении разных браузеров, введены некоторые приоритеты.

[Подробнее...](#)

Создайте файлы *index.html* и *style.css* на своём компьютере.

Внутри HTML расположите следующую запись:

```
<div id="main" class="text-white alert alert-warning">Какой-то текст</div>
```

и следующие CSS стили:

Проанализируйте получившийся результат.

```
div {
  width: 500px;
  height: 500px;
  background: #333333;
}
#main {
  color: white;
  width: 750px;
}
.text-white {
  color: white;
}
.alert {
  height: 350px;
  color: gray;
}
div {
  background: blue;
}
.alert-warning {
  background: #000000;
  color: yellow;
}
```

Знаете ли вы селекторы?

Основных видов селекторов всего несколько:

* – любые элементы.

div – элементы с таким тегом.

#id – элемент с данным id.

.class – элементы с таким классом.

[name="value"] – селекторы на атрибут (см. далее).

:visited – «псевдоклассы», остальные разные условия на элемент (см. далее).

Селекторы можно комбинировать, записывая последовательно, без пробела:

.c1.c2 – элементы одновременно с двумя классами c1 и c2

a#id.c1.c2:visited – элемент a с данным id, классами c1 и c2, и псевдоклассом visited

[Далее...](#)

Практическое занятие 2.5 / Отступ между элементами, размер одна строка

Вы можете использовать информацию выше как справочную для решения задач ниже, которые уже реально покажут, владеете вы CSS-селекторами или нет.

Есть список UL/LI.

Текст вверху без отступа от списка.

```
<ul>
  <li>Маша</li>
  <li>Паша</li>
  <li>Даша</li>
  <li>Женя</li>
  <li>Саша</li>
  <li>Гоша</li>
</ul>
```

Текст внизу без отступа от списка.
Сделайте, чтобы между элементами был вертикальный отступ.

- Размер отступа: ровно 1 строка.
- Нужно добавить только одно правило CSS с одним псевдоселектором, можно использовать CSS3.
- Не должно быть лишних отступов сверху и снизу списка

Размеры шрифта и строки заданы стилем:

```
body {
  font: 14px/1.5 serif;
}
```

Результат:

Текст вверху без отступа от списка.

- Маша
- Паша
- Даша
- Женя
- Саша
- Гоша

Текст внизу без отступа от списка.

Практическое занятие 2.5 / Отступ между парами, размером со строку

Есть список UL/LI.

Текст вверху без отступа от списка.

```
<ul>
  <li>Маша</li>
  <li>Паша</li>
  <li>Даша</li>
  <li>Женя</li>
  <li>Саша</li>
  <li>Гоша</li>
</ul>
```

Текст внизу без отступа от

Сделайте, чтобы между каждой парой элементов был вертикальный отступ.

- Размер отступа: ровно 1 строка.
- Нужно добавить только одно правило CSS, можно использовать CSS
- Не должно быть лишних отступов сверху и снизу списка.

Размеры шрифта и строки заданы стилем:

```
body {
  font: 14px/1.5 serif;
}
```

Результат:

Текст вверху без отступа от списка.

- Маша
- Паша

- Даша
- Женя

- Саша
- Гоша

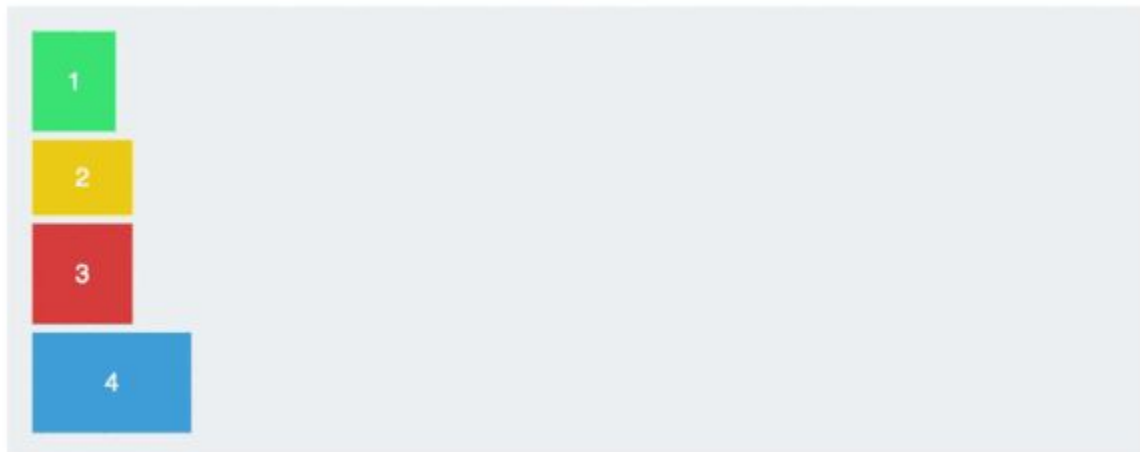
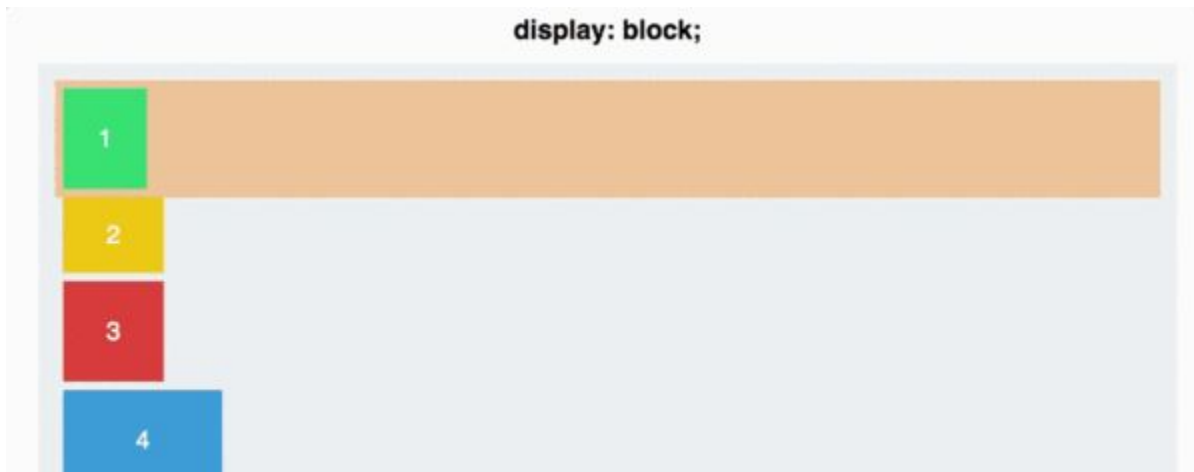
Текст внизу без отступа от списка.

Технология Flexbox

Display: Flex

Вот пример страницы:

(включите слайд на полный экран для включения анимации)



Чтобы начать работать с Flexbox, нам нужно сделать наш контейнер flex-контейнером. Делается это так:

```
#container {  
  display: flex;  
}
```

Для подробного ознакомления [кликните здесь](#)

Для закрепления практических навыков сыграйте в игру: (нажмите на картинку)



Практическое занятие 3.1 /Создаем сайт на

ЧИСТОМ HTML и CSS

Откройте папку layouts и откройте файл [3.1-WordPress2.psd](#).

Создайте собственную папку с проектом, в которой будет:

- Файл Index.html
- Файл Style.css
- Папка с картинками и задним фоном

