

Курс «Основы разработки сайтов»

Занятие №2

Основы CSS3

Время для вопросов

- Какие вопросы остались открытыми после предыдущего занятия?
- Что удалось (не удалось) сделать дома?

Дополнительные настройки в VS Code

- Необходимо установить в настройках:
 - Параметр Files: Auto Save - значение onFocusChange
 - Параметр Editor: Format On Save - поставить галочку
 - Параметр Editor: Default Formatter - поставить Prettier
- Для желающих интерфейса на русском:
 - Зажать комбинацию клавиш Ctrl+Shift+P
 - Ввести Configure Display Language
 - Выбрать русский (потребуется перезагрузка VS Code)

Основы CSS3

План сегодняшнего занятия:

- Основы CSS. Способы подключения. Синтаксис и элементы стилей.
- Виды селекторов и их вес.
- Наследование и каскадирование селекторов.
- Единицы измерения.
- Особенности позиционирования элементов на странице.
- Псевдоклассы и псевдоэлементы.

CSS. Что это и как использовать?

- CSS (англ. Cascading Style Sheets - «каскадные таблицы стилей») — язык для описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки HTML.
- Простыми словами - CSS используется для изменения стилей любых элементов в HTML. Мы можем менять размеры, цвета, расположение элементов и тд.
- Пример:

```
div {  
    background-color:red;  
    width: 100px;  
    height: 60px;  
}
```

[Making businesses more productive with digital tools and applications for the web.](#)

Latest blog post

Building an Application: What to Expect from the Discovery Phase



[Oliver Creswell](#)

01/03/2017



- Absolutely spot on article! The magic of microcopy <https://t.co/aEewVGKoGV> #UX #applicationdesign

@browserlondon 08:55 AM Mar 7th

Making businesses more productive with digital tools and applications for the web.

Latest blog post

Building an Application: What to Expect from the Discovery Phase



Oliver Creswell
01/03/2017



CSS. Синтаксис

- Если HTML состоит из тегов, то в CSS основные элементы - это **селекторы**.
- **Основной синтаксис:**
селектор {
 свойство1: значение1;
 свойство2: значение2;

 свойствоN: значениеN;
}
- Пример:
div {
 background-color:red;
 width: 100px;
 height: 60px;
}
- Обязательными параметрами и символами в описании свойств являются:
селектор, пара фигурных скобок (открытая и закрытая), название свойства и двоеточие, значение свойства и точка с запятой. Точка с запятой означает конец описания свойства. Закрытая фигурная скобка означает конец описания стиля селектора.
- Комментарии в css можно делать с помощью синтаксиса /* */

CSS. Способы подключения в проект

- **1. Подключение внешних стилей в <head> с помощью <link>:**
- **Пример:** `<link rel="stylesheet" href="style.css">`
- `rel="отношение"` - атрибут `rel` определяет отношения между текущим документом и файлом, на который делается ссылка. Это необходимо, чтобы браузер знал, как использовать подключаемый документ.
- `href="адрес"` - в качестве значения принимается полный или относительный путь (адрес) к файлу `.css`.

CSS. Способы подключения в проект

- **2. Подключение внутренних стилей через тег <style>, который должен находиться в <head>:**

- **Пример:**

```
<head>
```

```
...
```

```
<style>
```

```
    селекторы со стилями CSS
```

```
</style>
```

```
...
```

```
</head>
```

- **3. Использование CSS внутри тегов через атрибут style=«":**

- **Пример:** `<p style=«стили CSS">Рецепт печенья</p>`

Стиль будет применен только к содержимому этого тега.

CSS. Элементы стилей для текста

- **color** - изменение цвета текста. Принимает значения: цвета на английском языке, rgb(r,g,b), rgba(r,g,b,a), #hex.
- Примеры:
p {color: blue;}
div {color: rgb(255, 0 ,0);}
span{color: rgba(0,0,0,0.5);}
h1{color: #abcdef;}
- Значение #aabbcc аналогично записи #abc;
- **background-color** - изменение цвета для фона элемента. Принимает такие же значения, как и color.
- Пример textStyle1

CSS. Элементы стилей для текста

- **font-family:** шрифт, шрифт, шрифт, ...; - Устанавливает шрифт текста. Основные шрифты: arial, sans-serif, times, verdana, helvetica. <https://www.w3.org/Style/Examples/007/fonts.ru.html>
- **font-size:** размер; - определяет размер шрифта.
- **font-weight:** насыщенность; - устанавливает насыщенность шрифта. Значение указывается в виде слов (normal, bold) или чисел кратных 100 (100-900)
- **font-style:** определяет начертание шрифта — normal (обычное), italic (курсивное).
- **text-align:** выравнивание; - определяет горизонтальное выравнивание текста в пределах элемента. Значения: left (по левому краю), right (по правому краю), center (по центру), justify (по ширине);
- **line-height:** размер; - устанавливает высоту строки текста.
- **text-decoration:** дополнительные эффекты для текста. Значения - underline, overline, line-through, none. Дополнительные параметры - тип линии (solid, dashed, dotted, wavy) и размер линии. Пример **text-decoration: underline solid 5px;**
- Пример textStyle2 (с использованием Chrome DevTools - F12)

CSS. Элементы стилей для текста

- Задача. Написать CSS, чтоб получить текст с картинки.

мой текст

- Задача. Написать CSS, чтоб получить ссылку со стилистикой как на картинке.

МОЯ ССЫЛКА

- Задача. Написать CSS, чтоб получить таблицу со стилистикой как на картинке.

АвтоПарк

Usual	Luxury
Ford, Kia, Lada	BMW, Mercedes, Audi

CSS. Виды селекторов и их вес

- **Универсальный селектор:** выбирает все теги из документа. Синтаксис: `* { ...css... }`
- **Селектор тегов:** В качестве селектора может выступать любой элемент HTML.
- **Классы** - применяют, когда необходимо определить стиль для одного или нескольких элементов веб-страницы или задать разные стили для одного тега. Для указаний класса в HTML к тегу добавляется атрибут `class`. При описании стилей для класса вначале указывается символ точки (`.`), затем идёт имя класса. У одного элемента может быть сразу несколько классов.
- **Идентификаторы (ID)** - определяет уникальное имя элемента, которое используется для изменения его стиля. В отличие от классов идентификаторы должны быть **уникальны** для всего HTML документа. Обращение к идентификатору происходит через атрибут `id`, который может быть установлен в любой HTML тег. При описании стилей для идентификатора вначале указывается символ решётки (`#`), затем идёт имя идентификатора.
- **Веса селекторов (от меньшего к большему)** - универсальный -> тег -> класс -> id -> атрибут `style`, навешенный на элемент.
- **Пример selectors1**

CSS. Виды селекторов и их вес

- **Вложенные селекторы** - при создании веб-страницы часто приходится вкладывать одни теги внутрь других. Чтобы стили для этих тегов использовались корректно, помогут вложенные селекторы. Например, задать стиль для тега `` только когда он располагается внутри контейнера `<p>` можно через селектор:
`p b {`
- **Группированные селекторы** - используются, если одни и те же стили используются для нескольких селекторов сразу. Пример -
`p, div, .testClass, #testId {`
- Также селектор может включать в себя большую детализацию элемента, пример `div.class1.class2 {`
- **Пример selectors2**

CSS. Наследование и каскадирование селекторов

- **Наследование** - стили, присвоенные некоторому элементу, **наследуются** всеми потомками (вложенными элементами), если они не переопределены явно. В основном это работает только для стилей, относящихся к текстовому содержимому документа.
- **Каскадирование** - порядок следования правил в CSS имеет значение; когда применимы два правила, имеющие одинаковый вес, используется то, которое идёт в CSS последним. **Однако** это правило можно обойти, используя для свойства дополнительное значение **!important**.
Синтаксис: `p { color: red !important; }`
- **Пример priority1**

CSS. Виды селекторов и их вес

○ Задача.

1. Придумайте селектор, который выберет абзацы **<p>** внутри дивов **<div>**.
2. Придумайте селектор, который выберет все абзацы **<p>** из элемента с **id=test**.
3. Выберите все элементы с классом **bbb**.
4. Выберите все элементы с классом **bbb** из элемента с **id=test**.
5. Выберите все **<h2>** с классом **bbb**.
6. Выберите все абзацы **<p>** с классом **bbb** из элемента с **id=test**.
7. Выберите все абзацы **<p>** с классом **bbb** и **<h2>** с классом **xxx** одновременно.
8. Выберите все абзацы **<p>** с классом **bbb** из **id=test** и все абзацы **<p>** с классом **xxx** из **id=test** одновременно.

○ Ответы

1. `div p {`
2. `#test p {`
3. `.bbb {`
4. `#test .bbb {`
5. `h2.bbb {`
6. `#test p.bbb {`
7. `p.bbb, h2.xxx {`
8. `#test p.bbb, #test h2.xxx`

CSS. Виды селекторов и их вес

- Селектор **A > B** выбирает все элементы B, вложенные в элемент A только на **первом** уровне вложенности.
- Селектор **A ~ B** выбирает все элементы B, которые идут **после** A на том же уровне вложенности, что и A.
- Селектор **A + B** выбирает только первый элемент B, который идет после A.
- Селектор **[attr]** выбирает элементы, для которых задан этот атрибут. Например, `[border] {}` применит стили к HTML элементу `<table border=«1»>`.
- Селектор **[attr=value]** работает по имени и значению атрибута.
- Все указанные выше селекторы оцениваются по весу также, как и селектор класса.
- **Пример selectors3**

CSS. Единицы измерения

- px - измерение в пикселях экрана. Один пиксель равен одной точки на экране компьютера.
- % - устанавливает измерение в процентах относительно родительского элемента .
- em - относительное измерение высоты шрифта. 1em эквивалентна размеру установленного шрифта у родителя.
- rem - это размер относительно размера шрифта корневого элемента в документе.
- vh, vw - 1% от высоты\ширины экрана
- Также есть еще такие единицы как mm, cm, pt, pc, ex, ch, vmin, vmax, но из-за своей не сильной востребованности мы их рассматривать не будем.
- **Пример measures1.**

CSS. Элементы стилей для элемента

- **background-image:** url(путь к файлу) | none; Устанавливает фоновое изображение для элемента
- **background-repeat:** no-repeat | repeat | repeat-x | repeat-y | round; Если установлено фоновое изображение, то это свойство определяет будет ли оно повторяться.
- Для объединения этих свойств можно использовать одно свойство **background**.
- Пример background1

CSS. Элементы стилей для элемента

- **border:** толщина стиль цвет; - позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента. Толщина указывается в пикселях. Стиль границы может быть: **dotted (точками)**, **dashed (пунктиром)**, **solid (сплошной линией)**.
- Можно указывать свойства для толщины, стиля и цвета по отдельности, используя соответственно свойства **border-width**, **border-style** и **border-color**;
- Также для верхней, нижней, левой и правой границы можно использовать соответственно правила с приставкой **-top**, **-bottom**, **-left**, **-right** после слова **border**. Например: **border-top-color**, **border-bottom-width**, **border-left-width**.
- Для закругления границ у элемента используется **border-radius:** величина.
- Пример `border1`

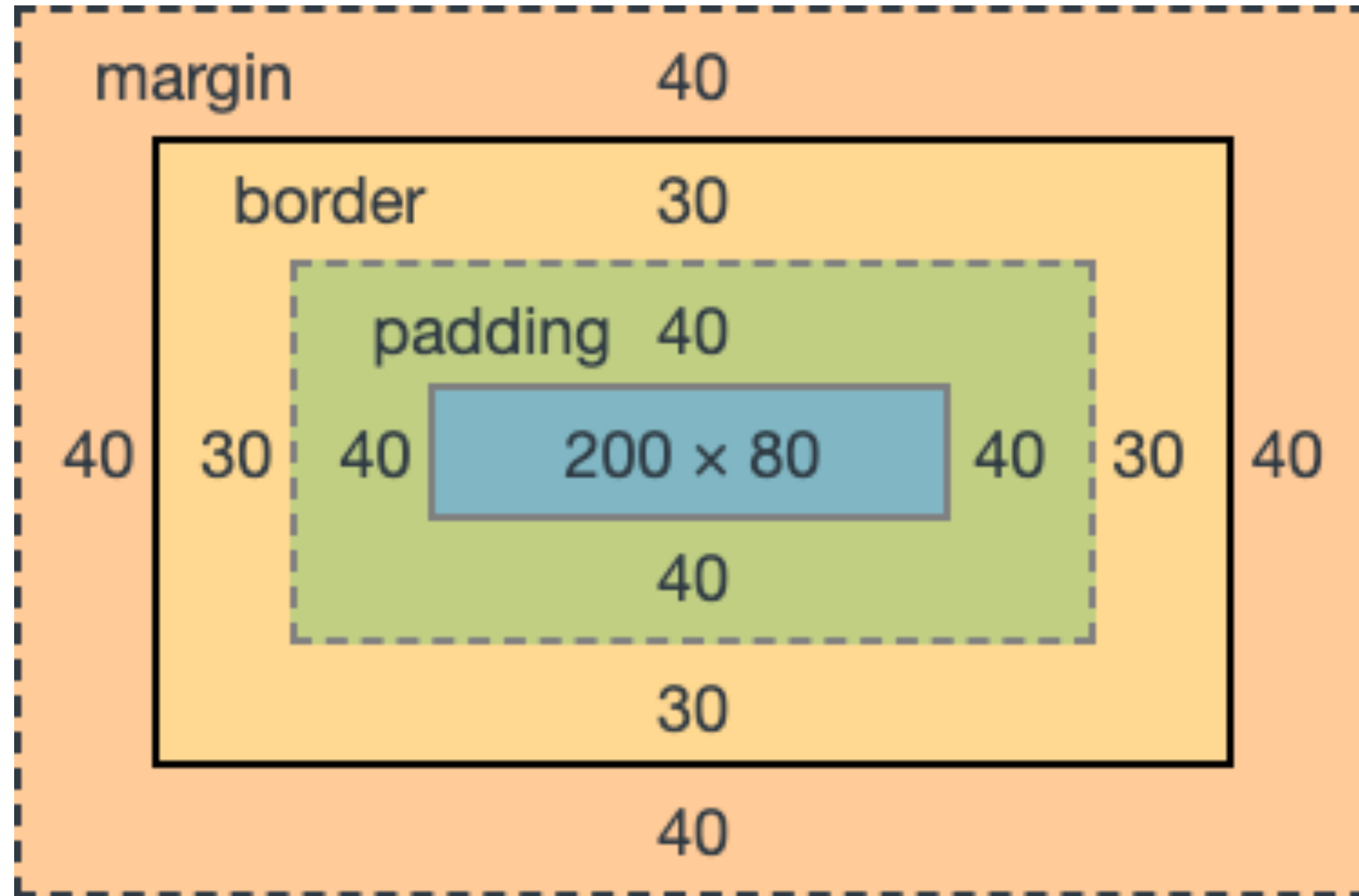
CSS. Элементы стилей для элемента

- **margin:** значение; - устанавливает для элемента величину отступа от элементов, которые его окружают.
- В margin можно записывать 1, 2, 3 или 4 значения.
 - 1- устанавливает отступы для всех направлений
 - 2- первое - для верха и низа, второе- для лево и право
 - 3- первое - для верха, второе- для лево и право, третье - для низа
 - 4- первое-для верха, второе- для право, третье - для низа, четвертое- для лево.
- Также для каждой из сторон можно указывать значение отдельным свойством, добавляя приставки -top, -right, -bottom, -left
- Пример margin1
- Для вертикальных margin есть отличительное поведение, которое называется **схлопыванием** маржинов. Маржин между двумя элементами А и В будет равняться максимальному из маржинов А и В.
- Пример margin2

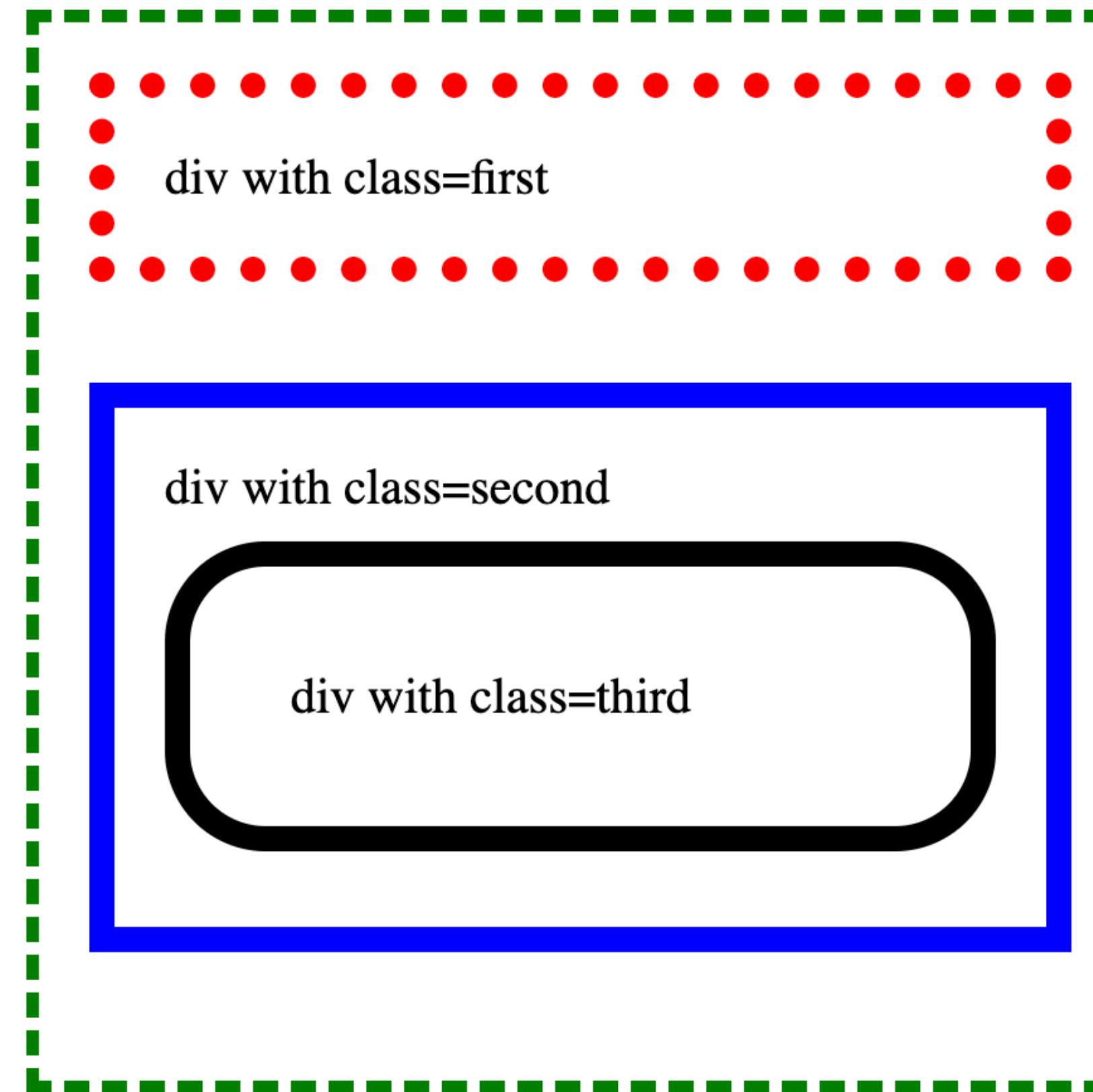
CSS. Элементы стилей для элемента

- **Padding:** значение; - устанавливает для элемента величину отступа от контента до его границы.
- В padding, аналогично margin, можно записывать 1, 2, 3 или 4 значения и также для каждой из сторон можно указывать значение отдельным свойством, добавляя приставки -top, -right, -bottom, -left.
- Свойство **box-sizing** применяется для изменения алгоритма расчёта ширины и высоты элемента. Два значения - content-box(в размеры входит только контент) и border-box(в размеры входят padding и border).
- Для изменения **ширины и высоты блочных** элементов (<p>, <div>, и тд) можно использовать соответственно свойства **width** и **height**. Единицы измерения - любые изученные.
- Пример padding1

CSS. Элементы стилей для блочного элемента



- Задача. Реализовать верстку со стилями как на примере ниже (mbpTask)



CSS. Особенности позиционирования элементов на странице.

- **Позиционированием** называется положение элемента в системе координат. Различают четыре типа позиционирования: нормальное, абсолютное, фиксированное и относительное. В зависимости от типа, который устанавливается через свойство **position**, изменяется и система координат.
- Свойства **left**, **top**, **right** и **bottom** означают смещение позиционированного элемента от соответствующего названию края и указываются в изученных единицах измерения.
- Благодаря комбинации свойств **position**, **left**, **top**, **right** и **bottom** элемент можно накладывать один на другой, выводить в точке с определёнными координатами, фиксировать в указанном месте, определить положение одного элемента относительно другого и др.

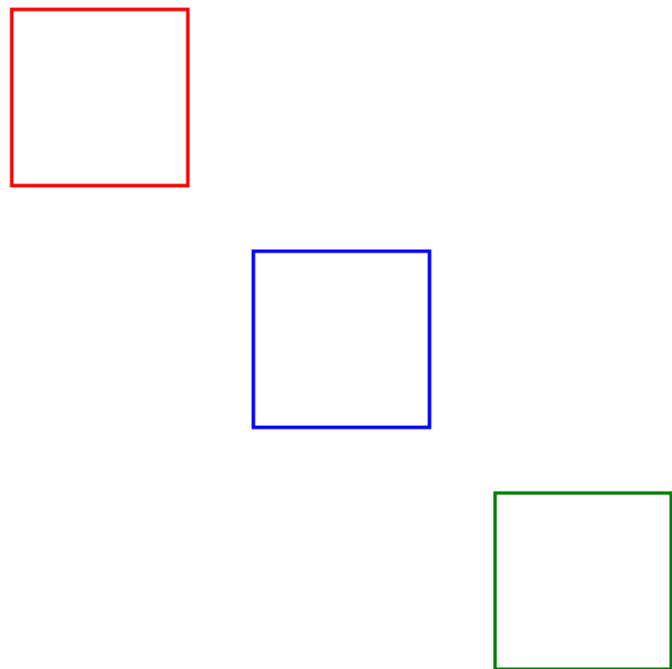
CSS. Особенности позиционирования элементов на странице.

- `position=static` - стандартное значение этого свойства, означает что элемент не позиционирован и выводится в потоке документа как обычно. Для такого значения свойства **left, top, right, bottom** игнорируются.
- `position=relative` - элемент позиционируется относительно его текущего положения в документе.
- `position=fixed` - элемент позиционируется относительно экрана и имеет фиксированную позицию. При этом сам элемент исчезает из потока элементов.
- `position=absolute` - элемент позиционируется относительно ближайшего позиционированного родителя (элемент, у которого `position` не `static`; если такого элемента нет, то выводится относительно элемента `body`). При этом свойстве элемент тоже исчезает из потока элементов.
- При позиционировании некоторые элементы могут перекрывать другие, и для контроля этого состояния есть свойство **z-index: значение**, которое работает по правилу «чем больше значение, тем выше элемент над всеми другими». Изначально равен 0 для всех элементов.
- Пример `position1.html`

CSS. Особенности позиционирования элементов на странице.

- Задачи. Повторить следующие позиционирования элементов:

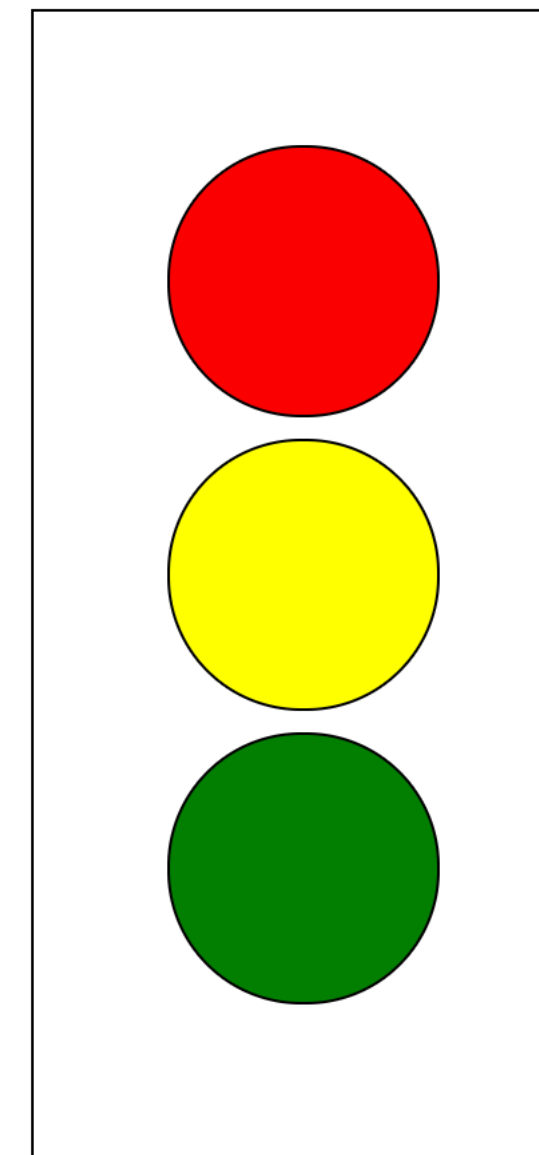
1



2



3



Для реализации круга можно использовать свойство `border-radius: 50%;`

CSS. Псевдоклассы.

- В дополнение к селекторам тегов, классов и идентификаторов нам доступны селекторы **псевдоклассов**, которые несут дополнительные возможности по выбору нужных элементов и добавлению стилей при определенном поведении элементов.
Синтаксис: селектор:**псевдокласс** {
- **:hover** - представляет элемент, на который пользователь навел указатель мыши.
- **:focus** - представляет элемент, который получает фокус, то есть когда пользователь нажимает клавишу табуляции или нажимает кнопкой мыши на поле ввода (например, текстовое поле).
- **:link** - применяется к ссылкам и представляет ссылку в обычном состоянии, по которой еще не совершен переход
- **:visited** - применяется к ссылкам и представляет ссылку, по которой пользователь уже переходил
- **:active** - применяется к ссылкам и представляет ссылку в тот момент, когда пользователь осуществляет по ней переход
- Пример pseudo1

CSS. Псевдоклассы.

- Особую группу псевдоклассов образуют псевдоклассы, которые позволяют выбрать определенные дочерние элементы.
- **:first-child** - представляет элемент, который является первым дочерним элементом
- **:last-child** - представляет элемент, который является последним дочерним элементом
- **:only-child** - представляет элемент, который является единственным дочерним элементом в каком-нибудь контейнере
- **:nth-child(n)** - представляет дочерний элемент, который имеет определенный номер n. Можно передать odd или even для выборки нечетных или четных дочерних элементов соответственно
- **:nth-last-child(n)** - представляет дочерний элемент, который имеет определенный номер n, начиная с конца.
- Пример pseudo2

CSS. Псевдоклассы форм.

- **:disabled** - выбирает элемент, если у него установлен атрибут disabled (например для кнопок)
- **:enabled** - выбирает элемент, если у него **не** установлен атрибут disabled
- **:checked** - выбирает элемент, если у него установлен атрибут checked (для флажков и радиокнопок)
- **:required** - выбирает элемент, если у него установлен атрибут required
- **:optional** - выбирает элемент, если у него **не** установлен атрибут required
- Пример pseudo3

CSS. Псевдоэлементы.

- **Псевдоэлементы** обладают рядом дополнительных возможностей по выбору элементов веб-страницы и похожи на псевдоклассы.
Синтаксис: селектор::**псевдоэлемент**{
- **::first-letter** - позволяет выбрать первую букву из текста
- **::first-line** - стилизует первую строку текста
- **::before** - добавляет сообщение до определенного элемента
- **::after** - добавляет сообщение после определенного элемента
- Для установки сообщения для псевдоэлементов after и before используется свойство **content**
- Пример pseudo4

CSS. Псевдоклассы и псевдоэлементы

- Задача1. Реализовать верстку, указанную ниже

<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>
<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>
<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>	<input type="text" value="текст"/>

- Задача2. Реализовать верстку, указанную ниже

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

- Задача3. Реализовать верстку, указанную ниже

☒ Первый чекбокс

☐ Второй чекбокс

☒ Радио1 ☐ Радио2

Итого

- Изучили основы CSS и способы подключения к HTML.
- Поговорили про селекторы и их вес.
- Узнали, какие есть единицы измерения в CSS.
- Разобрали принципы наследования и каскадирование стилей.
- Посмотрели особенности позиционирования элементов на странице с использованием свойств margin, border, padding, а также с использованием свойств position, left, right, top, bottom.
- Просмотрели список псевдоклассов и псевдоэлементов CSS.

Задачи

<https://webref.ru/practice>

Домашнее задание

Написать сайт-визитку (кто я, чем занимаюсь, мои увлечения и достижения)

Использовать:

- 1 HTML изображения
- 2 HTML ссылки (пр. Навигация)
- 3 HTML форма (пр. Оставить отзыв)
- 4 HTML список (пр. Мои хобби)
- 5 HTML таблица (пр. Оценки в школе)
- 6 CSS цвета и фон
- 7 CSS различные селекторы
- 8 CSS margin+border+padding
- 9 CSS позиционирование
- 10 CSS

псевдоклассы\псевдоэлементы

Отправлять на minzer.andrei@gmail.com

Спасибо за внимание!

Ресурсы для изучения:

- <http://htmlbook.ru/samcss>
- <https://webref.ru/practice>