Классы, идентификаторы. Специфичность. Наследование свойств

План урока

1. Что такое class и id?
2. В чем отличия?
3. Использование классов и id
4. Как называть?
5. Комбинации селекторов по тегу и классу
6. Каскадирование стилей
7. Специфичность или приоритетность
8. Наследование CSS стилей
9. Не наследуемые свойства

Что такое классы и идентификаторы?

Классы и идентификаторы применяют, когда необходимо определить стиль для индивидуального элемента веб-страницы или задать разные стили для одного тега.

Как их использовать?

Любому тегу можно задать атрибут class

```
<p class="red-text">Lorem Iposum</p>
<p class="blue-text">Lorem Iposum</p>
```

Или атрибут id

```
<p id="red-text">Lorem Iposum</p>
<p id="blue-text">Lorem Iposum</p>
```

Как их использовать?

В CSS можем обращаться к таким элементам с помощью следующих селекторов

```
.[класс]{
    /* Стили */
}
#[идентификатор]{
    /* Стили */
}
```

Как их использовать?

Пример

```
.red-text{
    color: red;
}
```

```
#blue-text{
    color: blue;
}
```

Отличия классов и id

1. Классы применяются ко многим элементам, id к одному
2. По id можно создавать якорные ссылки к элементам
3. id более ориентированы на JS и логику, но это не значит что их нельзя применять для стилизации.
4. id имеет более высокий приоритет(об этом далее)

Как правильно называть классы и id

1. Название класса или id должно быть понятным
2. Не называйте класс или id по контенту который в нем находится, если контент может быть динамическим
3. Не злоупотребляйте названиями вроде: yellow-text, big-text.
4. Используйте ТОЛЬКО английские слова в наименованиях класса
5. Используйте нижний регистр в названиях классов

Примеры хороших названий классов:

```
.content, .order-block, .wrapper,
#clients, .strip, .orders-wrapper, ...
```

Примеры неудачных названий классов:

```
.TextAligner, .BigText, .fotografia, .input,
.pole-dlya-vvoda, .super-hard-reading_className ...
```

Избегайте сложности в названии классов

```
.button {
  /* Хорошо */
}
.dropdown-button {
  /* Всё ещё хорошо */
}
.dropdown-button-part-one {
  /* Хм, по-прежнему хорошо, но будет нечитаемым при
добавлении дочернего элемента, например: */
}
.dropdown-button-part-one__button-admin {
  /* Ой, всё!!! */
}
```

Когда применять class, а когда id?

Класс лучше применить, когда элементов с таким свойством может быть много.

Идентификатор идеально подходит для разных блоков на странице к которым нужна навигация, для элементов, к которым может понадобиться доступ через JS и т/д

Комбинированные селекторы

Мы можем комбинировать селекторы по тегу и классу

```
table.table {
  width: 100%;
  margin-bottom: 50px;
  font-size: 13px;
  font-family: "HelveticaNeue";
  position: relative;
}
```

Каскадность стилей

Какой стиль будет применен?

```
.content {
  width: 100%;
  margin-bottom: 50px;
  font-size: 13px;
  font-family: "HelveticaNeue";
  position: relative;
  color: red;
}
.content{
    color: blue;
}
.content{
    color: green;
}
```

Специфичность или приоритетность

Специфичность определяет приоритеты примененных стилей

Какой стиль применить если указаны разные стили для тега и для класса?

Правильно ли ВСЕГДА применять последний стиль?

Специфичность или приоритетность

Пример:

index.html

```
<div class="content" id="main">
    Какой-то контент...
</div>
```

Какой цвет текста будет у этого элемента?

styles.css

```
.content{
  color: blue;
}
.content#main{
  color: green;
}
div.content{
  color: red;
}
```

Специфичность или приоритетность

Откройте Chrome Dev Tools и попробуйте отключить стили, посмотреть какой из стилей главнее

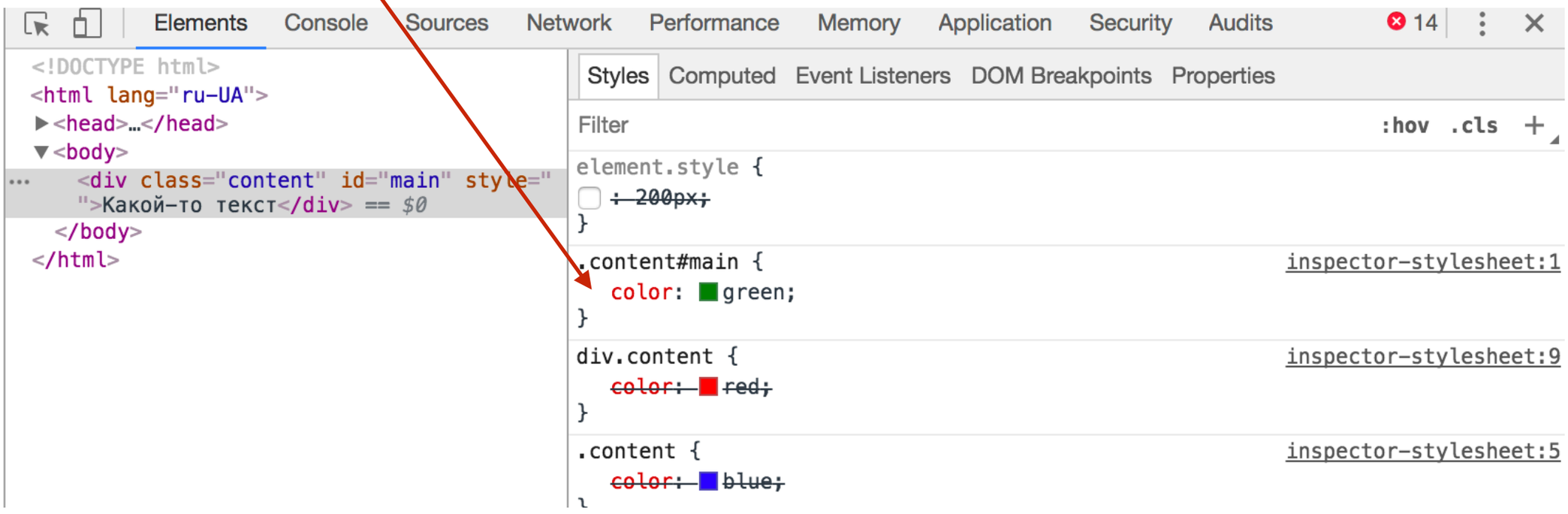

Таблица специфичности

Тэги(div, p, etc.)	Классы (.className)	Идентификаторы (#identifier)	Атрибут style=‘’
1 бал	10 баллов	100 баллов	1000 баллов

Пример:

```
.content#main
```

10 + 100 = 110

Чему равна приоритетность следующих селекторов?

1. .price
2. #select
3. span
4. .add-button.button
5. div#teams
6. div.some-class

Практическая задача

Оформить данный текст с помощью только CSS

Время выполнения ~10 минут

Я пришёл на курсы, чтобы изучить основы создания сайтов. От преподавателя я узнал, что семантика в HTML, это очень **важно.**
Есть теги, на которых *стоит акцентировать свое внимание.*
Я думаю будет не лишним их подчеркнуть. Это теги input и img
Они опасны тем, что ~~я их не знаю~~ их необязательно закрывать.
Особенно хочу выделить тег **<code>** с помощью него, мы можем вставлять примеры нашего кода на сайт, и стилизировать их.

Подсказка:

Используйте тэг span и классы для выделения нужного фрагмента.
Не забывайте о спецсимволах!

Понятие наследственности в CSS

Некоторые свойства в CSS могут применятся ко всем вложенным элементам:

```
<div class="content">
    <p>
        Абзац внутри которого текст
    </p>
</div>
```

```
.content{
    color: blue;
    text-transform: uppercase;
    text-decoration: underline;
}
```

Абзац внутри блока .content унаследует свойства текста

Какие свойства наследуются?

Прежде всего, это все свойства которые касаются оформления текста. (font-family, font-size, color, text-transform, text-decrotation и т/д)

Вложенные блоки так же наследуют некоторые свойства таблиц и списков, как например border-collapse, list-style и другие

Как явно задать наследование?

Мы можем явно указать наследование
используя свойство inherit.

Пример:

```
a{
   color: inherit;
}
```

Зачем это нужно?

Составные селекторы: родитель -> ребенок

Как выбрать с помощью css селектора
все теги span в каком-либо контейнере?

```
.content span{
    text-transform: uppercase;
}
```

С помощью ПРОБЕЛА – мы выбираем все элементы,
которые находятся внутри родительского селектора

Составные селекторы: прямой потомок

Чтобы выбрать только прямого потомка,
используем символ >

```
.content > span{
    text-transform: uppercase;
}
```

Просчет приоритетности составного селектора

Специфичность считается по тем же правилам:

.content > span 10 + 1 = 11

.content > span > b 10 + 2 = 12

.content #element 100 + 10 = 110

Селектор *

* – выбирает все элементы

Выберем все элементы на странице

```
*{
  color: inherit;
}
```

Выберем все элементы которые содержатся в блоке с классом .content

```
.content *{
  color: inherit;
}
```

Выберем всех прямых потомков которые содержатся в блоке с классом .content

```
.content > *{
  color: inherit;
}
```

Сброс стилей

```
*{
   margin: 0;
   padding: 0
}
```

1. normalize.css
2. screen.css
3. reset.css

Возможности Chrome dev tools для работы с классами

1. Мы можем добавлять класс к элементу или удалять его
2. Можем добавлять новые селекторы
3. Можем добавлять стили в атрибут style, тут нужно быть осторожнее, т.к такие стили получают самую высокую специфичность