

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика


Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub

 → [CSS для JavaScript-разработчика](#) 3-го июля 2019

Знаете ли вы селекторы?

CSS3-селекторы – фундаментально полезная вещь.

Даже если вы почему-то (старый IE?) не пользуетесь ими в CSS, есть много фреймворков для их кросс-браузерного использования CSS3 из JavaScript.

Поэтому эти селекторы необходимо знать.

Основные виды селекторов

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`



Отношения



В CSS3 предусмотрено четыре вида отношений между элементами.

Самые известные вы наверняка знаете:

- `div p` – элементы `p`, являющиеся потомками `div`.
- `div > p` – только непосредственные потомки

Есть и два более редких:

- `div ~ p` – правые соседи: все `p` на том же уровне вложенности, которые идут после `div`.
- `div + p` – первый правый сосед: `p` на том же уровне вложенности, который идёт сразу после `div` (если есть).

Посмотрим их на примере HTML:

```
1 <h3>Балтославянские языки</h3>
2
3 <ol id="languages">
4   ...Вложенный OL/LI список языков...
5 </ol>
```

CSS-селекторы:

```
1 /*+ no-beautify */
2 #languages li {
3   color: brown; /* потомки #languages, подходящие под
4 }
5
6 #languages > li {
7   color: black; /* первый уровень детей #languages по
8 }
9
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



```
10 #e-slavic { font-style: italic; }
11
12 #e-slavic ~ li { /* правые соседи #e-slavic с селектор
13     color: red;
14 }
15
16 #latvian {
17     font-style: italic;
18 }
19
20 #latvian * { /* потомки #latvian, подходяще под *
21     font-style: normal;
22 }
23
24 #latvian + li { /* первый правый сосед #latvian с селе
25     color: green;
26 }
```

Результат:

Балтославянские языки

1. Славянские языки

1. Славянские микроязыки
2. Праславянский язык
3. Восточнославянские языки `#e-slavic`
4. Западнославянские языки `#e-slavic ~ li`
5. Южнославянские языки `#e-slavic ~ li`
6. ... `#e-slavic ~ li`

2. Балтийские языки

1. Литовский язык `#latvian`
2. Латвийский язык `#latvian`
 1. Латгальский язык `#latvian *`
3. Прусский язык `#latvian + li`



Фильтр по месту среди соседей



При выборе элемента можно указать его место среди соседей.

Список псевдоклассов для этого:

- `:first-child` – первый потомок своего родителя.
- `:last-child` – последний потомок своего родителя.
- `:only-child` – единственный потомок своего родителя, соседних элементов нет.
- `:nth-child(a)` – потомок номер `a` своего родителя, например `:nth-child(2)` – второй потомок. Нумерация начинается с 1.
- `:nth-child(an+b)` – расширение предыдущего селектора через указание номера потомка формулой, где `a, b` – константы, а под `n` подразумевается любое целое число.

Этот псевдокласс будет фильтровать все элементы, которые попадают под формулу при каком-либо `n`. Например: `:nth-child(2n)` даст элементы номер 2, 4, 6 ..., то есть чётные.

- `:nth-child(2n+1)` даст элементы номер 1, 3 ..., то есть нечётные.
- `:nth-child(3n+2)` даст элементы номер 2, 5, 8 и так далее.

Пример использования для выделения в списке:

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



- Древнерусский язык
- Древненовгородский диалект `li:nth-child(2n)`
- **Западнорусский письменный язык** `li:nth-child(3)`
- Украинский язык `li:nth-child(2n)`
- Белорусский язык
- Другие языки `li:nth-child(2n)`

```
1 /*+ hide="CSS к примеру выше" no-beautify */
2 li:nth-child(2n) { /* чётные */
3     background: #eee;
4 }
5
6 li:nth-child(3) { /* 3-ий потомок */
7     color: red;
8 }
```

- `:nth-last-child(a)`, `:nth-last-child(an+b)` – то же самое, но отсчёт начинается с конца, например `:nth-last-child(2)` – второй элемент с конца.

Фильтр по месту среди соседей с тем же тегом

Есть аналогичные псевдоклассы, которые учитывают не всех соседей, а только с тем же тегом:

- `:first-of-type`
- `:last-of-type`
- `:only-of-type`
- `:nth-of-type`
- `:nth-last-of-type`

Они имеют в точности тот же смысл, что и обычные `:first-child`, `:last-child` и так далее, но во время подсчёта игнорируют элементы с другими тегами, чем тот, к которому применяется фильтр.

Пример использования для раскраски списка DT «через один» и предпоследнего DD:

```
Первый dt
  Описание dd
Второй dt dt:nth-of-type(2n)
  Описание dd
Третий dt
  Описание dd dd:nth-last-of-type(2)
Четвёртый dt dt:nth-of-type(2n)
  Описание dd
```

```
1 /*+ hide="CSS к примеру выше" no-beautify */
2 dt:nth-of-type(2n) {
3     /* чётные dt (соседи с другими тегами игнорируются) */
4     background: #eee;
5 }
6
7 dd:nth-last-of-type(2) {
8     /* второй dd снизу */
9 }
```



```
9   color: red;  
10 }
```



Как видим, селектор `dt:nth-of-type(2n)` выбрал каждый второй элемент `dt`, причём другие элементы (`dd`) в подсчётах не участвовали.



Селекторы атрибутов

На атрибут целиком:

- `[attr]` – атрибут установлен,
- `[attr="val"]` – атрибут равен `val`.

На начало атрибута:

- `[attr^="val"]` – атрибут начинается с `val`, например `"value"`.
- `[attr|="val"]` – атрибут равен `val` или начинается с `val-`, например равен `"val-1"`.

На содержание:

- `[attr*="val"]` – атрибут содержит подстроку `val`, например равен `"myvalue"`.
- `[attr~="val"]` – атрибут содержит `val` как одно из значений через пробел.

Например: `[attr~="delete"]` верно для `"edit delete"` и неверно для `"undelete"` или `"no-delete"`.

На конец атрибута:

- `[attr$="val"]` – атрибут заканчивается на `val`, например равен `"myval"`.

Другие псевдоклассы



- `:not(селектор)` – все, кроме подходящих под селектор.
- `:focus` – в фокусе.
- `:hover` – под мышью.
- `:empty` – без детей (даже без текстовых).
- `:checked`, `:disabled`, `:enabled` – состояния `INPUT`.
- `:target` – этот фильтр сработает для элемента, ID которого совпадает с анкором `#...` текущего URL.

Например, если на странице есть элемент с `id="intro"`, то правило `:target { color: red }` подсветит его в том случае, если текущий URL имеет вид `http://...#intro`.

Псевдоэлементы ::before, ::after

«Псевдоэлементы» – различные вспомогательные элементы, которые браузер записывает или может записать в документ.

При помощи псевдоэлементов `::before` и `::after` можно добавлять содержимое в начало и конец элемента:

```
1  <style>  
2    li::before {  
3      content: " [[ ";  
4    }  
5  
6    li::after {  
7      content: " ]] ";  
8    }  
9  </style>  
10  
11 Обратите внимание: содержимое добавляется <b>внутри</b>  
12
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



```
13 <ul>
14   <li>Первый элемент</li>
15   <li>Второй элемент</li>
16 </ul>
```

Обратите внимание: содержимое добавляется **внутри** LI.

- [[Первый элемент]]
- [[Второй элемент]]

Псевдоэлементы ::before / ::after добавили содержимое в начало и конец каждого LI.

:before или ::before?

Когда-то все браузеры реализовали эти псевдоэлементы с одним двоеточием: :after/:before.

Стандарт с тех пор изменился и сейчас все, кроме IE8, понимают также современную запись с двумя двоеточиями. А для IE8 нужно по-прежнему одно.

Поэтому если вам важна поддержка IE8, то имеет смысл использовать одно двоеточие.

Практика

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

Задачи



Выберите элементы селектором

важность: 5

HTML-документ:

```
1 <input type="checkbox">
2 <input type="checkbox" checked>
3 <input type="text" id="message">
4
5 <h3 id="widget-title">Сообщения:</h3>
6 <ul id="messages">
7   <li id="message-1">Сообщение 1</li>
8   <li id="message-2">Сообщение 2</li>
9   <li id="message-3" data-action="delete">Сообщение 3</li>
10  <li id="message-4" data-action="edit do-not-delete">Сообщение 4</li>
11  <li id="message-5" data-action="edit delete">Сообщение 5</li>
12  <li><a href="#">...</a></li>
13 </ul>
14
15 <a href="http://site.com/list.zip">Ссылка на архив</a>
16 <a href="http://site.com/list.pdf">...и на PDF</a>
```

Задания:

1. Выбрать input типа checkbox.
2. Выбрать input типа checkbox, НЕ отмеченный.
3. Найти все элементы с id=message или message-.*.
4. Найти все элементы с id=message-.*.
5. Найти все ссылки с расширением href="...zip".
6. Найти все элементы с атрибутом data-action, содержащим delete в списке (через пробел).
7. Найти все элементы, у которых ЕСТЬ атрибут data-action, но он НЕ содержит delete в списке (через пробел).



Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



8. Выбрать все чётные элементы списка #messages .

9. Выбрать один элемент сразу за заголовком h3#widget-title на том же уровне вложенности.

10. Выбрать все ссылки, следующие за заголовком h3#widget-title на том же уровне вложенности.

11. Выбрать ссылку внутри последнего элемента списка #messages .

[Открыть песочницу для задачи.](#)

решение

Отступ между элементами, размер одна строка

важность: 4

Есть список UL/LI .

```
1 Текст сверху без отступа от списка.
2 <ul>
3   <li>Маша</li>
4   <li>Паша</li>
5   <li>Даша</li>
6   <li>Женя</li>
7   <li>Саша</li>
8   <li>Гоша</li>
9 </ul>
10 Текст внизу без отступа от списка.
```

Размеры шрифта и строки заданы стилем:

```
1 body {
2   font: 14px/1.5 serif;
3 }
```

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.

Результат:

Текст сверху без отступа от списка.

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

Текст внизу без отступа от списка.

[Открыть песочницу для задачи.](#)

решение

Отступ между парами, размером со строку

важность: 4

Есть список UL/LI .

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Основные виды селекторов

Отношения

Фильтр по месту среди соседей

Фильтр по месту среди соседей с тем же тегом

Селекторы атрибутов

Другие псевдоклассы

Псевдоэлементы ::before, ::after

Практика

Задачи (3)

Комментарии

Поделиться



[Редактировать на GitHub](#)



```
1 Текст сверху без отступа от списка.
2 <ul>
3   <li>Маша</li>
4   <li>Паша</li>
5   <li>Даша</li>
6   <li>Женя</li>
7   <li>Саша</li>
8   <li>Гоша</li>
9 </ul>
10 Текст внизу без отступа от списка.
```

Размеры шрифта и строки заданы стилем:

```
1 body {
2   font: 14px/1.5 serif;
3 }
```

Сделайте, чтобы между каждой парой элементов был вертикальный отступ.

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

Результат:

Текст сверху без отступа от списка.

- Маша
- Паша

- Даша
- Женя

- Саша
- Гоша

Текст внизу без отступа от списка.

[Открыть песочницу для задачи.](#)

решение

Проводим [курсы по JavaScript и фреймворкам.](#)



Комментарии

перед тем как писать...