

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Значения box-sizing

Пример: подстроить ширину к родителю

Комментарии

Поделиться



Редактировать на GitHub

[🏠](#) → [CSS для JavaScript-разработчика](#)

📅 8-го сентября 2019

Свойство box-sizing

Свойство `box-sizing` может принимать одно из двух значений – `border-box` или `content-box`. В зависимости от выбранного значения браузер по-разному трактует значение свойств `width/height`.

Значения box-sizing

content-box

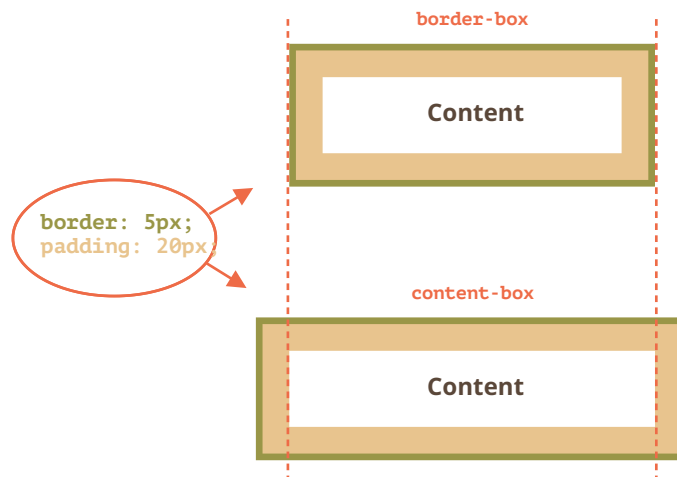
Это значение по умолчанию. В этом случае свойства `width/height` обозначают то, что находится *внутри* `padding`.

border-box

Значения `width/height` задают высоту/ширину *всего* элемента.

Для большей наглядности посмотрим на картинку этого `div` в зависимости от `box-sizing`:

```
1 /*+ no-beautify */
2 div {
3   width: 200px;
4   height: 100px;
5   box-sizing: border-box (вверху) | content-box (внизу)
6
7   padding: 20px;
8   border: 5px solid brown;
9 }
```



В верхнем случае браузер нарисовал весь элемент размером в `width` x `height`, в нижнем – интерпретировал `width/height` как размеры внутренней области.

Исторически сложилось так, что по умолчанию принят `content-box`, а `border-box` некоторые браузеры используют если не указан `DOCTYPE`, в режиме совместимости.

Но есть как минимум один случай, когда явное указание `border-box` может быть полезно: растягивание элемента до ширины родителя.

Пример: подстроить ширину к родителю

Задача: подогнать элемент по ширине внешнего элемента, чтобы он заполнял всё его пространство. Без привязки к конкретному размеру

Раздел

[CSS для JavaScript-разработчика](#)

[Навигация по уроку](#)

[Значения box-sizing](#)

[Пример: подстроить ширину к родителю](#)

[Комментарии](#)

[Поделиться](#)



[Редактировать на GitHub](#)



элемента в пикселях.

Например, мы хотим, чтобы элементы формы ниже были одинакового размера:

```
1 <style>
2   form {
3     width: 200px;
4     border: 2px solid green;
5   }
6
7   form input,
8   form select {
9     display: block;
10    padding-left: 5px;
11    /* padding для красоты */
12  }
13 </style>
14
15 <form>
16   <input>
17   <input>
18   <select>
19     <option>опции</option>
20   </select>
21 </form>
```



Как сделать, чтобы элементы растянулись чётко по ширине FORM ?
Попробуйте добиться этого самостоятельно, перед тем как читать дальше.



Попытка width:100%



Первое, что приходит в голову – поставить всем элементам INPUT ширину width: 100%.

Попробуем:

```
1 <style>
2   form {
3     width: 200px;
4     border: 2px solid green;
5   }
6
7   form input, form select {
8     display: block;
9     padding-left: 5px;
10    width: 100%;
11  }
12 </style>
13
14 <form>
15   <input>
16   <input>
17   <select><option>опции</option></select>
18 </form>
```



Как видно, не получается. Элементы вылезают за пределы родителя.

Причина – ширина элемента 100% по умолчанию относится к внутренней области, не включающей padding и border. То есть, внутренняя область

Раздел

[CSS для JavaScript-разработчика](#)

[Навигация по уроку](#)

[Значения box-sizing](#)

[Пример: подстроить ширину к родителю](#)

[Комментарии](#)

[Поделиться](#)



[Редактировать на GitHub](#)



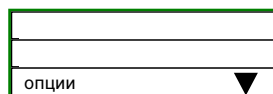
растягивается до 100% родителя, и к ней снаружи прибавляются padding/border, которые и вылезают.

Есть два решения этой проблемы.

Решение: дополнительный элемент

Можно убрать padding/border у элементов INPUT/SELECT и завернуть каждый из них в дополнительный DIV, который будет обеспечивать дизайн:

```
1 <style>
2   form {
3     width: 200px;
4     border: 2px solid green;
5   }
6   /* убрать padding/border */
7
8   form input,
9   form select {
10    padding: 0;
11    border: 0;
12    width: 100%;
13  }
14  /* внешний div даст дизайн */
15
16  form div {
17    padding-left: 5px;
18    border: 1px solid black;
19  }
20 </style>
21
22 <form>
23   <div>
24     <input>
25   </div>
26   <div>
27     <input>
28   </div>
29   <div>
30     <select>
31       <option>опции</option>
32     </select>
33   </div>
34 </form>
```



В принципе, это работает. Но нужны дополнительные элементы. А если мы делаем дерево или большую редактируемую таблицу, да и вообще – любой интерфейс, где элементов так много, то лишние нам точно не нужны.

Кроме того, такое решение заставляет пожертвовать встроенным в браузер дизайном элементов INPUT/SELECT.

Решение: box-sizing

Существует другой способ, гораздо более естественный, чем предыдущий.

При помощи box-sizing: border-box мы можем сказать браузеру, что ширина, которую мы ставим, относится к элементу полностью, включая border и padding:

```
1 <style>
2   form {
3     width: 200px;
```

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Значения box-sizing

Пример: подстроить ширину к родителю

Комментарии

Поделиться



[Редактировать на GitHub](#)



```
4     border: 2px solid green;
5   }
6
7   form input, form select {
8     display: block;
9     padding-left: 5px;
10    -moz-box-sizing: border-box; /* в Firefox нужен пре
11    box-sizing: border-box;
12    width: 100%;
13  }
14 </style>
15
16 <form>
17   <input>
18   <input>
19   <select><option>опции</option></select>
20 </form>
```



Мы сохранили «родную» рамку вокруг INPUT/SELECT и не добавили лишних элементов. Всё замечательно.

Свойство box-sizing поддерживается в IE начиная с версии 8.

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

Комментарии

перед тем как писать...