

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться

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

📅 25-го сентября 2019

Центрирование горизонтальное и вертикальное

В CSS есть всего несколько техник центрирования элементов. Если их знать, то большинство задач решаются просто.

Горизонтальное text-align

Для центрирования инлайновых элементов – достаточно поставить родителю `text-align: center`:

```
1 <style>
2   .outer {
3     text-align: center;
4     border: 1px solid blue;
5   }
6 </style>
7
8 <div class="outer">Текст</div>
```



Для центрирования блока это уже не подойдёт, свойство просто не подействует. Например:

```
1 <style>
2   .outer {
3     text-align: center;
4     border: 1px solid blue;
5   }
6   .inner {
7     width: 100px;
8     border: 1px solid red;
9   }
10 </style>
11
12 <div class="outer">
13   <div class="inner">Текст</div>
14 </div>
```



margin: auto

Блок по горизонтали центрируется `margin: auto`:

```
1 <style>
2   .outer {
3     border: 1px solid blue;
4   }
5   .inner {
6     width: 100px;
7     border: 1px solid red;
8     margin: auto;
9   }
10 </style>
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

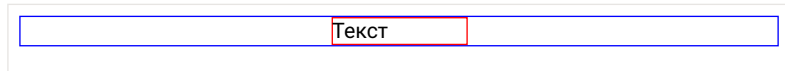
Поделиться



Редактировать на GitHub



```
11
12 <div class="outer">
13   <div class="inner">Текст</div>
14 </div>
```



В отличие от `width/height`, значение `auto` для `margin` само не появляется. Обычно `margin` равно конкретной величине для элемента, например `0` для `DIV`. Нужно поставить его явно.

Значение `margin-left:auto/margin-right:auto` заставляет браузер выделять под `margin` всё доступное сбоку пространство. А если и то и другое `auto`, то слева и справа будет одинаковый отступ, таким образом элемент окажется в середине. Детали вычислений описаны в разделе спецификации [Calculating widths and margins](#).

Вертикальное

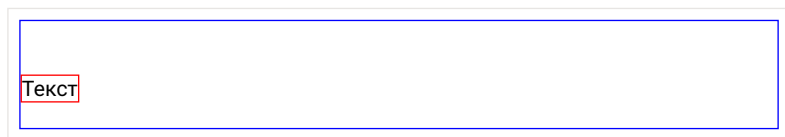
Для горизонтального центрирования всё просто. Вертикальное же изначально не было предусмотрено в спецификации CSS и по сей день вызывает ряд проблем.

Есть три основных решения.

`position:absolute + margin`

Центрируемый элемент позиционируем абсолютно и опускаем до середины по вертикали при помощи `top:50%`:

```
1 <style>
2   .outer {
3     position: relative;
4     height: 5em;
5     border: 1px solid blue;
6   }
7
8   .inner {
9     position: absolute;
10    top: 50%;
11    border: 1px solid red;
12  }
13 </style>
14
15 <div class="outer">
16   <div class="inner">Текст</div>
17 </div>
```



Это, конечно, не совсем центр. По центру находится верхняя граница. Нужно ещё приподнять элемент на половину своей высоты.

Высота центрируемого элемента должна быть известна. Родитель может иметь любую высоту.

Если мы знаем, что это ровно одна строка, то её высота равна `line-height`.

Приподнимем элемент на пол-высоты при помощи `margin-top`:

```
1 <style>
2   .outer {
3     position: relative;
4     height: 5em;
5     border: 1px solid blue;
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



```
6    }
7
8    .inner {
9        position: absolute;
10       top: 50%;
11       margin-top: -0.625em;
12       border: 1px solid red;
13    }
14 </style>
15
16 <div class="outer">
17   <div class="inner">Текст</div>
18 </div>
```

Текст

Почему -0.625em ?

При стандартных настройках браузера высота строки `line-height: 1.25`, если поделить на два $1.25em / 2 = 0.625em$.

Конечно, высота может быть и другой, главное чтобы мы её знали заранее.

Можно аналогично центрировать и по горизонтали, если известен горизонтальный размер, при помощи `left:50%` и отрицательного `margin-left`.

Одна строка: `line-height`



Вертикально отцентрировать одну строку в элементе с известной высотой `height` можно, указав эту высоту в свойстве `line-height`:



```
1 <style>
2   .outer {
3       height: 5em;
4       line-height: 5em;
5       border: 1px solid blue;
6   }
7 </style>
8
9 <div class="outer">
10  <span style="border:1px solid red">Текст</span>
11 </div>
```

Текст

Это работает, но лишь до тех пор, пока строка одна, а если содержимое вдруг переносится на другую строку, то начинает выглядеть довольно уродливо.

Таблица с `vertical-align`

У свойства `vertical-align`, которое управляет вертикальным расположением элемента, есть два режима работы.

В таблицах свойство `vertical-align` указывает расположение содержимого ячейки.

Его возможные значения:

baseline

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



Значение по умолчанию.

middle, top, bottom

Располагать содержимое посередине, вверху, внизу ячейки.

Например, ниже есть таблица со всеми 3-мя значениями:

```
1 <style>
2   table { border-collapse: collapse; }
3   td {
4       border: 1px solid blue;
5       height: 100px;
6   }
7 </style>
8
9 <table>
10 <tr>
11   <td style="vertical-align: top">top</td>
12   <td style="vertical-align: middle">middle</td>
13   <td style="vertical-align: bottom">bottom</td>
14 </tr>
15 </table>
```

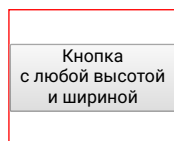
top		
	middle	
		bottom

Обратим внимание, что в ячейке с `vertical-align: middle` содержимое находится по центру. Таким образом, можно обернуть нужный элемент в таблицу размера `width: 100%; height: 100%` с одной ячейкой, у которой указать `vertical-align: middle`, и он будет отцентрирован.

Но мы рассмотрим более красивый способ, который поддерживается во всех современных браузерах, и в IE8+. В них не обязательно делать таблицу, так как доступно значение `display: table-cell`. Для элемента с таким `display` используются те же алгоритмы вычисления ширины и центрирования, что и в `TD`. И, в том числе, работает `vertical-align`:

Пример центрирования:

```
1 <div style="display: table-cell; vertical-align: middle
2   <button>Кнопка<br>с любой высотой<br>и шириной</button>
3 </div>
```



Этот способ замечателен тем, что он не требует знания высоты элементов.

Однако у него есть особенность. Вместе с `vertical-align` родительский блок получает табличный алгоритм вычисления ширины и начинает подстраиваться под содержимое. Это не всегда желательно.

Чтобы его растянуть, нужно указать `width` явно, например: `300px`:

```
1 <div style="display: table-cell; vertical-align: middle
2   <button>Кнопка<br>с любой высотой<br>и шириной</button>
3 </div>
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



Кнопка
с любой высотой
и шириной

Можно и в процентах, но в примере выше они не сработают, потому что структура таблицы «сломана» – ячейка есть, а собственно таблицы-то нет.

Это можно починить, завернув «псевдоячейку» в элемент с `display: table`, которому и поставим ширину:

```
1 <div style="display: table; width: 100%">
2 <div style="display: table-cell; vertical-align: middle
3   <button>Кнопка<br>с любой высотой<br>и шириной</butto
4 </div>
5 </div>
```

Кнопка
с любой высотой
и шириной

Если дополнительно нужно горизонтальное центрирование – оно обеспечивается другими средствами, например `margin: 0 auto` для блочных элементов или `text-align: center` на родителе – для других.

Центрирование в строке с `vertical-align`

Для инлайновых элементов (`display: inline/inline-block`), включая картинки, свойство `vertical-align` центрирует сам инлайн-элемент в окружающем его тексте.

В этом случае набор значений несколько другой:

Картинка размером в 30px, значения `vertical-align`:

baseline(по умолчанию) 30
middle(по середине) 30
subуровень с <sub> 30
superуровень с <sup> 30
text-top(верхняя граница вровень с текстом) 30
text-bottom(нижняя граница вровень с текстом) 30


Это можно использовать и для центрирования, если высота родителя известна, а центрируемого элемента – нет.

Допустим, высота внешнего элемента 120px. Укажем её в свойстве `line-height`:

```
1 <style>
2   .outer {
3     line-height: 120px;
4   }
5   .inner {
6     display: inline-block; /* центрировать..*/
7     vertical-align: middle; /* ..по вертикали */
8     line-height: 1.25; /* переопределить высоту строки
9     border: 1px solid red;
10  }
11 </style>
12 <div class="outer" style="height: 120px;border: 1px sol
13
```



```
14   <span class="inner">Центрирован<br>вертикально</span>
    </div>
```



Центрирован
вертикально

Работает во всех браузерах и IE8+.

Свойство `line-height` наследуется, поэтому надо знать «правильную» высоту строки и переопределять её для `inner`.


Центрирование с `vertical-align` без таблиц

Если центрирование должно работать для любой высоты родителя и центрируемого элемента, то обычно используют таблицы или `display: table-cell` с `vertical-align`.

Если центрируются не-блочные элементы, например `inline` или `inline-block`, то `vertical-align` может решить задачу без всяких таблиц. Правда, понадобится вспомогательный элемент (можно через `:before`).

Пример:

```
1  <style>
2  .before {
3    display: inline-block;
4    height: 100%;
5    vertical-align: middle;
6  }
7
8  .inner {
9    display: inline-block;
10   vertical-align: middle;
11 }
12 </style>
13
14 <div class="outer" style="height:100px;border:1px solid
15   <span class="before"></span>
16   <span class="inner" style="border:1px solid red">
17     Центрированный<br>Элемент
18   </span>
19 </div>
```



Центрированный
Элемент

- Перед центрируемым элементом помещается вспомогательный инлайн-блок `before`, занимающий всю возможную высоту.
- Центрируемый блок выровнен по его середине.

Для всех современных браузеров и IE8 можно добавить вспомогательный элемент через `:before`:

```
1  <style>
2  .outer:before {
3    content: '';
4    display: inline-block;
5    height: 100%;
6    vertical-align: middle;
7  }
8
9  .inner {
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



```
10 display: inline-block;
11 vertical-align: middle;
12 }
13
14 /* добавим горизонтальное центрирование */
15 .outer {
16   text-align: center;
17 }
18 </style>
19
20 <div class="outer" style="height:100px; width: 100%; bo
21   <span class="inner" style="border:1px solid red">
22     Центрированный<br>Элемент
23   </span>
24 </div>
```

Центрированный
Элемент

В пример выше добавлено также горизонтальное центрирование `text-align: center`. Но вы можете видеть, что на самом деле внутренний элемент не центрирован горизонтально, он немного сдвинут вправо.

Это происходит потому, что центрируется весь текст, а перед `inner` находится пробел, который занимает место.

Варианта два:

1. Убрать лишний пробел между `div` и началом `inner`, будет `<div class="outer">...`
2. Оставить пробел, но сделать отрицательный `margin-left` у `inner`, равный размеру пробела, чтобы `inner` сместился левее.



Второе решение:



```
1 <style>
2 .outer:before {
3   content: '';
4   display: inline-block;
5   height: 100%;
6   vertical-align: middle;
7 }
8
9 .inner {
10  display: inline-block;
11  vertical-align: middle;
12  margin-left: -0.35em;
13 }
14
15 .outer {
16   text-align: center;
17 }
18 </style>
19
20 <div class="outer" style="height:100px; width: 100%; bo
21   <span class="inner" style="border:1px solid red">
22     Центрированный<br>Элемент
23   </span>
24 </div>
```

Центрированный
Элемент

Центрирование с использованием модели flexbox

Данный метод поддерживается всеми современными браузерами.

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



[Редактировать на GitHub](#)



```
1 <style>
2 .outer {
3     display: flex;
4     justify-content: center; /*Центрирование по горизон
5     align-items: center;    /*Центрирование по вертика
6 }
7 </style>
8
9 <div class="outer" style="height:100px; width: 100%; bo
10 <span class="inner" style="border:1px solid red">
11     Центрированный<br>Элемент
12 </span>
13 </div>
```

Центрированный
Элемент

Плюсы:

- Не требуется знания высоты центрируемого элемента.
- CSS чистый, короткий и не требует дополнительных элементов.

Минусы:

- Не поддерживается IE9-, IE10 поддерживает предыдущую версию flexbox.

Итого

Обобщим решения, которые обсуждались в этой статье.



Для горизонтального центрирования:

- `text-align: center` – центрирует инлайн-элементы в блоке.
- `margin: 0 auto` – центрирует блок внутри родителя. У блока должна быть указана ширина.



Для вертикального центрирования одного блока внутри другого:

Если размер центрируемого элемента известен, а родителя – нет

Родителю `position: relative`, потомку `position: absolute; top: 50% и margin-top: -<половина-высоты-потомка>`. Аналогично можно отцентрировать и по горизонтали.

Если нужно отцентрировать одну строку в блоке, высота которого известна

Поставить блоку `line-height: <высота>`. Нужны конкретные единицы высоты (px, em ...). Значение `line-height: 100%` не будет работать, т.к. проценты берутся не от высоты блока, а от текущей `line-height`.

Высота родителя известна, а центрируемого элемента – нет.

Поставить `line-height` родителю во всю его высоту, а потомку поставить `display: inline-block`.

Высота обоих элементов неизвестна.

Три варианта:

1. Сделать элемент-родитель ячейкой таблицы при помощи `display: table-cell` (IE8) или реальной таблицы, и поставить ему `vertical-align: middle`. Отлично работает, но мы имеем дело с таблицей вместо обычного блока.
2. Решение со вспомогательным элементом `outer: before` и инлайн-блоками. Вполне универсально и не создаёт таблицу.
3. Решение с использованием flexbox.

✓ **Задачи**

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



[Редактировать на GitHub](#)

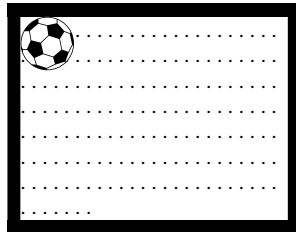


Поместите мяч в центр поля (CSS)

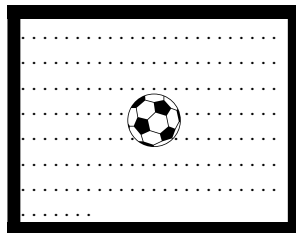
важность: 5

Поместите мяч в центр поля при помощи CSS.

Исходный код:



Используйте CSS, чтобы поместить мяч в центр:



- CSS для центрирования может использовать размеры мяча.
- CSS для центрирования не должен опираться на конкретный размер поля.

[Открыть песочницу для задачи.](#)



решение



Форма + модальное окно

важность: 5

Создайте при помощи HTML/CSS форму для логина в модальном окне.

Текст Текст Текст

Эта кнопка не работает

Текст Текст Те

Добро пожаловать!

Логин

Пароль

Войти

Требования:

- Кнопки окна вне формы не работают (даже на левый край нажать нельзя).
- Полупрозрачный голубой «экран» отстоит от границ на 20px .
- Форма центрирована вертикально и горизонтально, её размеры фиксированы.
- Посетитель может менять размер окна браузера, геометрия должна сохраняться.
- Не ломается при прокрутке.

Браузеры: все основные, IE8+.

Открыть песочницу для задачи.

решение

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Горизонтальное

Вертикальное

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



vertical-align + table-cell + position = ?

важность: 5

В коде ниже используется вертикальное центрирование при помощи `table-cell + vertical-align`.

Почему оно не работает? Нажмите на просмотр, чтобы увидеть (стрелка должна быть в центре по вертикали).

```
1 <style>
2   .arrow {
3     position: absolute;
4     height: 60px;
5     border: 1px solid black;
6     font-size: 28px;
7
8     display: table-cell;
9     vertical-align: middle;
10  }
11 </style>
12
13 <div class="arrow"><</div>
```

Как починить центрирование при помощи CSS? Свойства `position/height` менять нельзя.

решение



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

Комментарии

перед тем как писать...