

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться

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

13-го января 2020

# Единицы измерения: px, em, rem и другие

В этом очерке я постараюсь не только рассказать о различных единицах измерения, но и построить общую картину – что и когда выбирать.

## Пиксели: px

Пиксель px – это самая базовая, абсолютная и окончательная единица измерения.

Количество пикселей задаётся в настройках [разрешения экрана](#), один px – это как раз один такой пиксель на экране. Все значения браузер в итоге пересчитает в пиксели.

Пиксели могут быть дробными, например размер можно задать в 16.5px. Это совершенно нормально, браузер сам использует дробные пиксели для внутренних вычислений. К примеру, есть элемент шириной в 100px, его нужно разделить на три части – волей-неволей появляются 33.333...px. При окончательном отображении дробные пиксели, конечно же, округляются и становятся целыми.

Для мобильных устройств, у которых много пикселей на экране, но сам экран маленький, чтобы обеспечить читаемость, браузер автоматически применяет масштабирование.

### Достоинства

- Главное достоинство пикселя – чёткость и понятность

### Недостатки

- Другие единицы измерения – в некотором смысле «мощнее», они являются относительными и позволяют устанавливать соотношения между различными размерами

### ⚠ Давно на свалке: mm, cm, pt, pc

Существуют также «производные» от пикселя единицы измерения: mm, cm, pt и pc, но они давно отправились на свалку истории.

Вот, если интересно, их значения:

- 1mm (мм) = 3.8px
- 1cm (см) = 38px
- 1pt (типографский пункт) = 4/3 px
- 1pc (типографская пика) = 16px

Так как браузер пересчитывает эти значения в пиксели, то смысла в их употреблении нет.

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



Редактировать на GitHub



### Почему в сантиметре см содержится ровно 38 пикселей?

В реальной жизни сантиметр – это эталон длины, одна сотая метра. А [пиксель](#) может быть разным, в зависимости от экрана.

Но в формулах выше под пикселем понимается «сферический пиксель в вакууме», точка на «стандартизованном экране», характеристики которого описаны в [спецификации](#).

Поэтому ни о каком соответствии см реальному сантиметру здесь нет и речи. Это полностью синтетическая и производная единица измерения, которая не нужна.

## Относительно шрифта: em

1em – текущий размер шрифта.

Можно брать любые пропорции от текущего шрифта: 2em, 0.5em и т.п.

**Размеры в em – относительные, они определяются по текущему контексту.**

Например, давайте сравним px с em на таком примере:

```
1 <div style="font-size:24px">
2   Страусы
3   <div style="font-size:24px">Живут также в Африке</div>
4 </div>
```

Страусы  
Живут также в Африке

24 пикселей – и в Африке 24 пикселей, поэтому размер шрифта в <div> одинаков.

А вот аналогичный пример с em вместо px :

```
1 <div style="font-size:1.5em">
2   Страусы
3   <div style="font-size:1.5em">Живут также в Африке</div>
4 </div>
```

Страусы  
Живут также в Африке

Так как значение в em высчитывается относительно *текущего шрифта*, то вложенная строка в 1.5 раза больше, чем первая.

Выходит, размеры, заданные в em, будут уменьшаться или увеличиваться вместе со шрифтом. С учётом того, что размер шрифта обычно определяется в родителе, и может быть изменён ровно в одном месте, это бывает очень удобно.

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



Редактировать на GitHub



### Что такое размер шрифта?

Что такое «размер шрифта»? Это вовсе не «размер самой большой буквы в нём», как можно было бы подумать.

Размер шрифта – это некоторая «условная единица», которая встроена в шрифт.

Она обычно чуть больше, чем расстояние от верха самой большой буквы до низа самой маленькой. То есть, предполагается, что в эту высоту помещается любая буква или их сочетание. Но при этом «хвосты» букв, таких как `p`, `g` могут заходить за это значение, то есть вылезать снизу. Поэтому обычно высоту строки делают чуть больше, чем размер шрифта.

### Единицы ex и ch

В спецификации указаны также единицы `ex` и `ch`, которые означают размер символа `"x"` и размер символа `"0"`.

Эти размеры присутствуют в шрифте всегда, даже если по коду этих символов в шрифте находятся другие значения, а не именно буква `"x"` и ноль `"0"`. В этом случае они носят более условный характер.

Эти единицы используются чрезвычайно редко, так как «размер шрифта» `em` обычно вполне подходит.

## Проценты %

Проценты `%`, как и `em` – относительные единицы.

Когда мы говорим «процент», то возникает вопрос – «Процент от чего?»

Как правило, процент будет от значения свойства родителя с тем же названием, но не всегда.

Это очень важная особенность процентов, про которую, увы, часто забывают.

Отличный источник информации по этой теме – стандарт, [Visual formatting model details](#).

Вот пример с `%`, он выглядит в точности так же, как с `em`:

```
1 <div style="font-size:150%">
2   Страусы
3   <div style="font-size:150%">Живут также в Африке</div>
4 </div>
```

Страусы

Живут также в Африке

В примере выше процент берётся от размера шрифта родителя.

А вот примеры-исключения, в которых `%` берётся не так:

#### margin-left

При установке свойства `margin-left` в `%`, процент берётся от *ширины* родительского блока, а вовсе не от его `margin-left`.

#### line-height

При установке свойства `line-height` в `%`, процент берётся от текущего *размера шрифта*, а вовсе не от `line-height` родителя. Детали по `line-height` и размеру шрифта вы также можете найти в статье [Свойства font-size и line-height](#).

#### width/height

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



Редактировать на GitHub



Для width/height обычно процент от ширины/высоты родителя, но при position: fixed, процент берётся от ширины/высоты окна (а не родителя и не документа). Кроме того, иногда % требует соблюдения дополнительных условий, за примером – обратитесь к главе [Особенности свойства height в %](#).

## Единица rem: смесь px и em

Итак, мы рассмотрели:

- px – абсолютные, чёткие, понятные, не зависящие ни от чего.
- em – относительно размера шрифта.
- % – относительно такого же свойства родителя (а может и не родителя, а может и не такого же – см. примеры выше).

Может быть, пора уже остановиться, может этого достаточно?

Э-э, нет! Не все вещи делаются удобно.

Вернёмся к теме шрифтов. Бывают задачи, когда мы хотим сделать на странице большие кнопки «Шрифт больше» и «Шрифт меньше». При нажатии на них будет срабатывать JavaScript, который будет увеличивать или уменьшать шрифт.

Вообще-то это можно сделать без JavaScript, в браузере обычно есть горячие клавиши для масштабирования вроде Ctrl+**+**, но они работают слишком тупо – берут и увеличивают всю страницу, вместе с изображениями и другими элементами, которые масштабировать как раз не надо. А если надо увеличить только шрифт, потому что посетитель хочет комфортнее читать?

Какую единицу использовать для задания шрифтов? Наверно не px, ведь значения в px абсолютны, если менять, то во всех стилевых правилах. Вполне возможна ситуация, когда мы в одном правиле размер поменяли, а другое забыли.

Следующие кандидаты – em и %.

Разницы между ними здесь нет, так как при задании font-size в процентах, эти проценты берутся от font-size родителя, то есть ведут себя так же, как и em.

Вроде бы, использовать можно, однако есть проблема.

Попробуем использовать этот подход для <li>.

Протестируем на таком списке:

```
1 <ul>
2 <li>Собака
3   <ul>
4     <li>бывает
5       <ul>
6         <li>кусачей
7         <ul>
8         <li>только
9         <ul>
10        <li>от жизни
11        <ul>
12        <li>собачей</li>
13        </ul>
14        </li>
15        </ul>
16        </li>
17        </ul>
18      </li>
19    </ul>
20  </li>
21 </ul>
22 </li>
23 </ul>
```



Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



Редактировать на GitHub



- Собака
  - бывает
    - кусачей
      - только
        - от жизни
          - собачей

Пока это обычный вложенный список.

Теперь уменьшим размер шрифта до 0.8em, вот что получится:

```
1 <style>
2   li {
3     font-size: 0.8em;
4   }
5 </style>
6
7 <ul>
8 <li>Собака
9   <ul>
10    <li>бывает
11      <ul>
12        <li>кусачей
13          <ul>
14            <li>только
15              <ul>
16                <li>от жизни
17                  <ul>
18                    <li>собачей</li>
19                  </ul>
20                </li>
21              </ul>
22            </li>
23          </ul>
24        </li>
25      </ul>
26    </li>
27  </ul>
28 </li>
29 </ul>
```

- Собака
  - бывает
    - кусачей
      - только
        - от жизни
          - собачей

Проблема очевидна. Хотели, как лучше, а получилось... Мелковато. Каждый вложенный `<li>` получил размер шрифта 0.8 от родителя, в итоге уменьшившись до нечитаемого состояния. Это не совсем то, чего мы бы здесь хотели.

Можно уменьшить размер шрифта только на одном «корневом элементе»... Или воспользоваться единицей `rem`, которая, можно сказать, специально придумана для таких случаев!

**Единица `rem` задаёт размер относительно размера шрифта элемента `<html>`.**

Как правило, браузеры ставят этому элементу некоторый «разумный» (reasonable) размер по умолчанию, который мы, конечно, можем переопределить и использовать `rem` для задания шрифтов внутри относительно него:

```
1 <style>
2   html {
3     font-size: 14px;
4   }
5   li {
6     font-size: 0.8rem;
7   }
```

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица gem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
8 </style>
9
10 <div><button id="up">Кликните, чтобы увеличить размер ш
11
12 
13
14 <ul>
15 <li>Собака
16   <ul>
17     <li>бывает
18       <ul>
19         <li>кусачей
20           <ul>
21             <li>только
22               <ul>
23                 <li>от жизни
24                   <ul>
25                     <li>собачей</li>
26                   </ul>
27                 </li>
28               </ul>
29             </li>
30           </ul>
31         </li>
32       </ul>
33     </li>
34   </ul>
35 </li>
36 </ul>
37
38 <script>
39 let html = document.documentElement;
40 up.onclick = function() {
41   // при помощи JS увеличить размер шрифта html на 2px
42   html.style.fontSize = parseInt(getComputedStyle(html,
43   });
44 </script>
```

Кликните, чтобы увеличить размер шрифта



- Собака
  - бывает
    - кусачей
      - только
        - от жизни
          - собачей

Получилось удобное масштабирование для шрифтов, не влияющее на другие элементы.

Элементы, размер которых задан в `gem`, не зависят друг от друга и от контекста – и этим похожи на `px`, а с другой стороны они все заданы относительно размера шрифта `<html>`.

Единица `gem` не поддерживается в IE8-.

## Относительно экрана: vw, vh, vmin, vmax

Во всех современных браузерах, исключая IE8-, поддерживаются новые единицы из черновика стандарта [CSS Values and Units 3](#):

- `vw` – 1% ширины окна

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Пиксели: px

Относительно шрифта: em

Проценты %

Единица rem: смесь px и em

Относительно экрана: vw, vh, vmin, vmax

Итого

Комментарии

Поделиться



[Редактировать на GitHub](#)



- `vh` – 1% высоты окна
- `vmin` – наименьшее из (`vw`, `vh`), в IE9 обозначается `vm`
- `vmax` – наибольшее из (`vw`, `vh`)

Эти значения были созданы, в первую очередь, для поддержки мобильных устройств.

Их основное преимущество – в том, что любые размеры, которые в них заданы, автоматически масштабируются при изменении размеров окна.

# Этот текст написан с размером '5vh'.

Вы сможете легко увидеть, как работает `vh`, если поменяете высоту окна браузера. Текст выше будет расти/уменьшаться.

## Итого

Мы рассмотрели единицы измерения:

- `px` – абсолютные пиксели, к которым привязаны и потому не нужны `mm`, `cm`, `pt` и `pc`. Используется для максимально конкретного и точного задания размеров.
- `em` – задаёт размер относительно шрифта родителя, можно относительно конкретных символов: `"x"` (`ex`) и `"0"` (`ch`), используется там, где нужно упростить масштабирование компоненты.
- `rem` – задаёт размер относительно шрифта `<html>`, используется для удобства глобального масштабирования: элементы которые планируется масштабировать, задаются в `rem`, а JS меняет шрифт у `<html>`.
- `%` – относительно такого же свойства родителя (как правило, но не всегда), используется для ширин, высот и так далее, без него никуда, но надо знать, относительно чего он считает проценты.
- `vw`, `vh`, `vmin`, `vmax` – относительно размера экрана.

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



## Комментарии

перед тем как писать...