**Единицы измерения: px, em, rem и другие**

**[Пиксели: px](https://learn.javascript.ru/css-units" \l "pikseli-px)**

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

Количество пикселей задаётся в настройках [разрешения экрана](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)), 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.

**Почему в сантиметре cm содержится ровно 38 пикселей?**

В реальной жизни сантиметр – это эталон длины, одна сотая метра. А [пиксель](http://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D0%BA%D1%81%D0%B5%D0%BB%D1%8C) 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](http://www.w3.org/TR/CSS2/syndata.html#length-units).

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

**[Относительно шрифта: em](https://learn.javascript.ru/css-units" \l "otnositelno-shrifta-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:

<div style="font-size:24px">

Страусы

<div style="font-size:24px">Живут также в Африке</div>

</div>

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

А вот аналогичный пример с em вместо px:

<div style="font-size:1.5em">

Страусы

<div style="font-size:1.5em">Живут также в Африке</div>

</div>

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.

**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 р, 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](http://www.w3.org/TR/css3-values/#ex-unit) и [ch](http://www.w3.org/TR/css3-values/" \l "ch-unit), 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 %](https://learn.javascript.ru/css-units" \l "protsenty)**

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](http://www.w3.org/TR/CSS2/visudet.html).

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

<div style="font-size:150%">

Страусы

<div style="font-size:150%">Живут также в Африке</div>

</div>

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](https://learn.javascript.ru/font-size-line-height).

**width/height**

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 %](https://learn.javascript.ru/height-percent).

**[Единица rem: смесь px и em](https://learn.javascript.ru/css-units" \l "edinitsa-rem-smes-px-i-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:

<ul>

<li>Собака

<ul>

<li>бывает

<ul>

<li>кусачей

<ul>

<li>только

<ul>

<li>от жизни

<ul>

<li>собачей</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

Пока это обычный вложенный список.

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

<style>

li {

font-size: 0.8em;

}

</style>

<ul>

<li>Собака

<ul>

<li>бывает

<ul>

<li>кусачей

<ul>

<li>только

<ul>

<li>от жизни

<ul>

<li>собачей</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

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:

<style>

html {

font-size: 14px;

}

li {

font-size: 0.8rem;

}

</style>

<div><button id="up">Кликните, чтобы увеличить размер шрифта</button></div>

<img src="https://js.cx/clipart/angry\_dog.png">

<ul>

<li>Собака

<ul>

<li>бывает

<ul>

<li>кусачей

<ul>

<li>только

<ul>

<li>от жизни

<ul>

<li>собачей</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

</ul>

<script>

let html = document.documentElement;

up.onclick = function() {

// при помощи JS увеличить размер шрифта html на 2px

html.style.fontSize = parseInt(getComputedStyle(html, '').fontSize) + 2 + 'px';

};

</script>

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

Элементы, размер которых задан в rem, 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 rem не поддерживается в IE8-.

**[Относительно экрана: vw, vh, vmin, vmax](https://learn.javascript.ru/css-units" \l "otnositelno-ekrana-vw-vh-vmin-vmax)**

Во всех современных браузерах, исключая IE8-, поддерживаются новые единицы из черновика стандарта [CSS Values and Units 3](http://dev.w3.org/csswg/css3-values/):

* vw – 1% ширины окна
* 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](https://learn.javascript.ru/css-units" \l "itogo)**

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.