

Единицы измерения

В прошлых уроках вы уже узнали о единицах измерения *px*. Можно заметить, что стоит указать где-то `20px` и это значение будет зафиксировано. Неважно что вокруг элемента с таким значением, какое разрешение экрана или какое устройство используется. 20 пикселей всегда будут 20 пикселей.

Пиксель является *абсолютной* единицей измерения. Как уже было сказано выше, такая единица измерения строго фиксирована и не меняется от внешних факторов. Изменить размер можно только переопределением.

В противовес *абсолютным* единицам измерения существуют *относительные* единицы измерения. По их названию видно, что они зависят «от чего-то». А вот от чего они зависят мы рассмотрим в конце урока.

В этом уроке будут рассмотрены все основные единицы измерения. На самом деле их достаточно много, но не все используются постоянно. Если у вас возникнет желание познакомиться со всеми доступными единицами измерения, то в конце урока будет ссылка на официальную спецификацию.

Абсолютные единицы

Абсолютные единицы измерения можно считать самыми простыми. Достаточно получить немного опыта, и вы уже на глаз сможете определять, какой размер имеет тот или иной элемент внутри документа.

Самой распространённой единицей является пиксель. С ней вы уже знакомы по другим урокам. Но что же такое пиксель? Если не вдаваться в электронику, то пиксель — это всего лишь минимальный элемент, из которого состоит изображение на экране. Значит, мы указываем, сколько таких элементов должно быть занято на экране устройства.

При этом стоит понимать, что 100 пикселей на одном экране могут быть не равны 100 пикселям на другом экране. Всё зависит от размера пикселей в мониторе. А это, в свою очередь, зависит от характеристик экрана и его разрешения.

По этим причинам можно назвать пиксель как относительной, так и абсолютной единицей измерения. Всё зависит от того, с какой точки зрения мы смотрим. Для нас важен пиксель в контексте CSS, где он является абсолютной единицей.

```
div {  
  height: 100px;  
  
  /* Блочный элемент занимает строго 100 пикселей. Реальный размер (в сантиметрах) может  
  отличаться на разных устройствах */  
}
```

Другие абсолютные единицы почти не используются в разработке для веб-устройств. Стоит их перечислить, но есть вероятность, что вы никогда не будете их использовать.

- *cm* — сантиметры
- *mm* — миллиметры
- *Q* — четверть миллиметра
- *in* — дюйм
- *pc* — пайка/пика. Термин из типографики
- *pt* — пункт

Заметьте, что среди этих единиц есть знакомые в реальной жизни. При вёрстке для печати это может быть полезно.

Относительные единицы

Самая запутанная часть урока. Относительные единицы берут своё значение от других значений. Самой универсальной такой единицей являются проценты. Они всегда берут свои значения от ближайшего родительского элемента. Остальные единицы можно разделить на две группы:

1. Относительные единицы, рассчитывающиеся от размера шрифта.
2. Относительные единицы, рассчитывающиеся от размера экрана. Это немного грубое сравнение — на самом деле расчёт ведётся от размера *viewport*. О том, что это такое, вы узнаете из курса по адаптивности.

Проценты

Как уже было сказано, проценты рассчитываются от значения того же свойства ближайшего родителя. Родитель — элемент, внутри которого находится текущий блок. Например:

```
<html>
  <body>
    <!-- body — дочерний элемент html. html — родительский элемент для body -->
    <main>
      <!-- main — дочерний элемент body. body — родительский элемент для main -->
      <section>
        <!-- section — дочерний элемент main. main — родительский элемент для section -->
        <p><!-- p — дочерний элемент section. section — родительский элемент для p --></p>
        <div><!-- div — дочерний элемент section. section — родительский элемент для div -->
        </div>
      </section>
    </main>
  </body>
</html>
```

Рассмотрим на примере ширины и размера шрифта.

```
<section class="w-400px size-50px bg-blue">
  <div class="w-50p bg-gray">
    <p class="size-70p">Параграф текста</p>
  </div>
</section>
.w-400px {
  width: 400px;
}
```

```
.w-50p {
  width: 50%;
}

.size-50px {
  font-size: 50px;
}

.size-70p {
  font-size: 70%;
}
```

Что же здесь происходит? Разберёмся по шагам:

- Для основного родительского `<section>` установлены размеры в абсолютных единицах измерения. Ширина 400 пикселей и размер шрифта 50 пикселей.
- Вложенный `<div>` имеет размер 50%. Этот размер относительный и высчитывается относительно ближайшего родителя. У этого родителя размер 400 пикселей. Следовательно, размер блока будет 200 пикселей.
- Параграф текста имеет размер шрифта 70%. Ближайший родитель с установленным размером шрифта — `<section>`. Параграфу установится размер шрифта в 70% от 50 пикселей. То есть 35px.

Интересным примером является установка процентов для тега или класса, который предполагает вложение друг в друга. Взгляните на следующий пример:

```
<ul class="size-30px">
  <li>Элемент 1
    <ul>
      <li>Элемент 1.1</li>
      <li>Элемент 1.2
        <ul>
          <li>Элемент 1.2.1</li>
          <li>Элемент 1.2.2</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Элемент 2</li>
  <li>Элемент 3</li>
  <li>Элемент 4</li>
</ul>
.size-30px {
  font-size: 30px;
}

li {
  font-size: 70%;
}
```

Перед тем, как идти дальше, попробуйте проанализировать эту разметку и посчитать, какие размеры шрифта будут у каждого элемента списка. Не пугайтесь, если вы запутаетесь. Это может быть сложно.

Поразмыслили? Молодцы! Теперь посмотрим, как это выглядит:

Проанализируем результат:

- Размер шрифта самого первого элемента `` — 30 пикселей.

- Все элементы `` имеют размер шрифта 70%. Это значит, что чем глубже список, тем меньше размер шрифта:
 - Элементы 1, 2, 3, 4 будут иметь размер 21 пиксель.
 - Элементы 1.1 и 1.2 лежат внутри *элемент 1*. Размер шрифта этого элемента 21 пиксель. Это следует из прошлого пункта. Теперь размер шрифта считается именно от 21 пикселя.
 - Элементы 1.1 и 1.2 будут иметь размер 14.7 пикселя.
 - Элементы 1.2.1 и 1.2.2 лежат внутри *элемент 1.2*. Размер шрифта этого элемента 14.7 пикселей. Это следует из прошлого пункта.
 - Элементы 1.2.1 и 1.2.2 будут иметь размер шрифта 10.29 пикселя.

Относительные единицы, рассчитывающиеся от размера шрифта

Выдохните — самое сложное уже позади. Если вы внимательно изучили работу процентов, то другие относительные единицы не вызовут сложностей. Есть две основные относительные единицы измерения, которые зависят от размера шрифта: *em* и *rem*.

Значение *em* очень похоже на использование процентов. Оно показывает *во сколько раз* размер будет больше, чем у ближайшего вычисленного значения `font-size`. Если в прошлом примере со списками заменить 70% на *0.7em*, то результат будет тот же самый.

Тогда в чём смысл *em*? Эту единицу измерения можно использовать не только для шрифта, но и для любых других свойств, которые принимают числовое значение. Например, в следующем коде размер нижнего внешнего отступа будет 60px:

```
section {  
  font-size: 30px;  
}  
  
section p {  
  margin-bottom: 2em;  
}
```

Но если у параграфа выставить своё значение `font-size`, то единица *em* будет высчитываться именно от него. В этом и кроется смысл формулировки «чем у ближайшего вычисленного значения `font-size`».

Второй такой единицей является *rem*. Её способ работы точно такой же, как у *em*. Но есть одно громадное отличие: *rem* отсчитывается от размера шрифта *корневого элемента*. Корневым элементом является `<html>`. Именно размер шрифта у этого элемента и учитывается при расчёте единицы *rem*.

Вернёмся к примеру со списком. Заменяв *em* на *rem* получим совершенно другой результат. *0.7rem* высчитываются из одного и того же места, без учёта вложенности. Итого, каждый элемент будет иметь размер шрифта 11.2px. Это связано со стандартным значением размера шрифта в 16 пикселей у большинства браузеров.

Вы можете заметить, что размер шрифта в 30 пикселей у элемента `` не играет, в данном случае, никакой роли.

Относительные единицы, рассчитывающиеся от размера экрана

CSS позволяет задавать размеры относительно текущего размера *viewport* — размера окна браузера, доступного без прокрутки. Основными двумя единицами являются *vw* — ширина *viewport* и *vh* — высота *viewport*. Вы можете рассматривать это как процент от размера *viewport*.

Частый «кейс» с использованием таких единиц — создание секции на всю доступную область браузера.

```
<section class="one-screen-section"></section>
body {
  margin: 0;
  padding: 0;
}

.one-screen-section {
  width: 100vw;
  height: 100vh;
  background: #80deea;
}
```