

Цели занятия

- узнать про единицы измерения размеров, абсолютные и относительные величины;
- поговорить про то, что такое резиновая верстка;
- повторить про минимальные и максимальные размеры;

Полезные материалы

- [Справочник CSS](#)
- [Единицы измерения размеров в CSS](#)
- [Резиновый дизайн. Что и куда будем растягивать?](#)

Единицы измерения размеров

Пиксели — это самая базовая, абсолютная и окончательная единица измерения.

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.

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

Вот 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.

em, % — эти единицы измерения относительные и высчитываются на основе размеров родительского элемента, но в конечном итоге их значение всё равно будет переведено в пиксели.

Например, если родительскому элементу задать font-size: 10px, а внутри него разместить параграф, у которого будет указано font-size: 2em, то итоговый размер текста будет 20px, потому что оно будет рассчитано по формуле $10px * 2 = 20px$.

Причём при изменении размеров у родительского элемента размер текста внутри также будет автоматически изменяться.

rem — ещё одна единица измерения, которая является относительной и её поведение очень похоже на em или проценты. Есть лишь одна разница — значение рассчитывается не относительно родителя, а относительно корневого элемента страницы (т.е. html).

Это, например, позволяет пропорционально изменять размер сразу всех текстов на странице, которые были заданы через rem просто меняя начальный размер текста на html.

vw, vh — сравнительно новые, но уже хорошо поддерживаемые свойства, которые позволяют задавать размеры относительно ширины или высоты экрана пользователя. То есть 1vw равен 1% от ширины устройства, а 1vh равен 1% от высоты устройства и их

размеры автоматически пересчитываются, если размер устройства/окна браузера меняется.

Резиновая вёрстка

Веб-разработка существует уже более 20 лет и за это время подходы к вёрстке страниц сильно изменялись. Поначалу большинство страниц создавалось с помощью таблиц и так называемой табличной верстки, затем их начали вытеснять страницы с использованием `float`, но большинство страниц всё ещё имели фиксированные размеры блоков.

И с развитием мобильных устройств и просто устройств с разными экранами это начало создавать много проблем для разработчиков — им приходилось верстать несколько версий одной и той же страницы под разные экраны, чтобы сайт выглядел красиво и удобно. Именно тогда появился тренд вместо фиксированных размеров использовать проценты, чтобы блоки зависели от ширины родителя и это было названо “резиновой вёрсткой”

С появлением Flex Layout и единиц измерения `vh`, `vw` этот подход стал также уходить на второй план и на данный момент его не стоит изучать отдельно, лучше сфокусироваться на работе с флексами.

Но многие начинающие разработчики путают понятия резиновой и адаптивной вёрстки, поэтому, чтобы понять разницу, крайне рекомендую прочесть [данную статью](#). Более подробно принципы адаптивной вёрстки будут разобраны в отдельном занятии.

Свойства `min` и `max`

В современном мире пользователи уже привыкли к тому, что сайт должен одинаково хорошо отображаться как на самых широких экранах, так и на мобильных устройствах. И в этом разработчикам помогает Flexbox Layout, который позволяет гибко управлять

размерами элементов внутри контейнера, сжимая или растягивая их.

Но иногда этот процесс необходимо контролировать более гибко и для этого есть свойства, которые позволяют установить некоторые рамки сжатия/растягивания:

min-width — элемент ни при каких условиях не будет сжиматься по ширине меньше, чем указано в этом свойстве;

min-height — то же самое что и min-width, но для высоты элемента;

max-width — свойство позволяет задать максимальную ширину элемента, больше которой он не может растягиваться ни при каких условиях;

max-height — то же самое что и max-width, но для высоты элемента;