приоритеты селекторов

Таблицы стилей, добавляемые нами на страницу, не единственные, к которым обращается браузер. Существуют разные источники этих таблиц:

Авторские - созданные разработчиком страницы;

Пользовательские - заданные пользователями браузера;

Браузерные - установленные браузером по умолчанию.



* Тип данных - то, в чём задаются/выражаются значения свойств. То есть, тип данных определяет допустимые значения и формат, которые могут быть применены к определенным свойствам. Этих типов довольно много. Они могут выражаться в тексте(строках), числах, единицах измерения, функциях, процентах и т.д. Вот вкратце основные типы данных (жирным выделены наиболее важные для нас):
* <length>: Определяет значения длины, такие как пиксели, проценты, относительные единицы и т. д.
* <number>: Задает числовые значения.
* <keywords>: Задаёт значения в виде ключевых слов.
* <string>: Задает строковые значения, такие как текстовые данные или имена.
* <color>: Задает цветовые значения, такие как названия цветов, шестнадцатеричные коды, функции цвета и т. д.
* <angle>: Используется для задания угловых значений, таких как повороты и наклоны элементов.
* <position>: Определяет позиционные значения, такие как координаты, отступы и выравнивание.
* <percentage>: Используется для задания процентных значений относительно других свойств или контекста.
* <image>: Позволяет задавать изображения или ссылки на изображения.
* <time>: Задает временные значения, например, для анимаций или задержек.
* <flex>: Используется для задания гибкого макета элементов с использованием модуля CSSGrid.
* <url>: Используется для задания URL-адресов или ссылок на ресурсы.
* <blend-mode>: Определяет режим смешивания цветов при наложении элементов друг на друга.
* <custom-ident>: Позволяет создавать пользовательские идентификаторы (имена), которые можно использовать в CSS. Именно к этому типу относятся названия классов и идентификаторов. Чувствительны к регистру.
* <filter-function>: Описывает функции фильтров, которые применяются к элементам для изменения их внешнего вида, такие как размытие, насыщенность и т. д.
* <frequency>: Задает значения частоты, например, для анимаций или аудио.
* <gradient>: Определяет градиентные значения, такие как линейные и радиальные градиенты.
* <ratio>: Задает соотношение сторон для элементов, например, для видео или изображений.
* <single-transition-timing-function>: Определяет функции времени для анимаций и переходов.
* <transform-function>: Описывает функции трансформации, такие как перемещение, масштабирование и поворот элементов.

Пусть вас не смущает то, что названия взяты в угловые скобки - это не html-элементы, а сделано для обозначения того, что это является именем типа данных в спецификации CSS или в контексте разговора о CSS.

Разные типы данных могут выражаться другими типами данных вроде - <url> задаётся с помощью <string>, проценты (<percentage>)выражаются с помощью <number> и знака "%".



Адаптивный дизайн - это настройка реакции таблицы стилей на изменение размера окна браузера и/или на разные устройства, в которых открывается сайт - монитор настольного ПК, ноутбук, планшет, смартфон.

transform: rotateZ(1turn);

Вообще абсолютные единицы: px, s, ms, deg, pt ...

Относительные:

Проценты могут применяться к огромному списку свойств, но зачастую используются в свойствах width, height, margin, padding, font-size, position и др.

Псевдокласс **:root** в CSS используется для обращения к самому главному родительскому элементу документа. Его также называют корневым элементом.

**em**

* Как ни странно, мера em тоже пришла в веб-вёрстку из полиграфии. В CSS em равняется размеру шрифта текущего элемента. Вот некоторые особенности данной единицы измерения:
* Относительность: значение em зависит от размера шрифта текущего элемента или его родительского элемента. Если единица применяется для размера шрифта, то она зависит от унаследованного от родителя размера. Если единица применяется для других свойств, то она зависит от размера шрифта текущего элемента.
* Множители: можно использовать множители для определения размера, отличного от размера текущего шрифта. Например, font-size: 1.5em; увеличит размер шрифта на 50% от размера текущего шрифта.
* Каскадирование вложенных элементов: значение em наследуется от непосредственного родительского элемента. Поэтому, чем больше уровень вложенности, тем больше будет изменяться шрифт. В примере ниже каждый вложенный элемент <div> умножает размер шрифта предыдущего блока на 2.

**rem**

**Единица измерения rem (от "root em") в CSS**

**Viewport (область просмотра) -** ограниченная область в окне браузера, где отображается веб-страница, за исключением адресной строки, панели инструментов и панели текущего состояния.

Существуют также единицы измерения, относящиеся к размеру экрана (вернее, к размеру области просмотра)

1vh = 1/100 высоты области просмотра;

1vw = 1/100 ширины области просмотра;

1vmin | 1vmax = 1/100 наименьшего из двух предыдущих измерений;

Функции

calc()

Функция саlс() позволяет выполнять базовые арифметические вычисления с двумя и более значениями. Это особенно полезно для объединения значений, измеряемых в разных единицах. Функция поддерживает сложение (+), вычитание (-), умножение (\*) и деление (/). Операторы сложения и вычитания должны быть окружены пробелами, поэтому стоит выработать привычку всегда ставить пробел до и после каждого оператора, например: calc(1em + 10рх).



В примере ниже функция применяется для шрифта font-size: calc(0.5em + 1vw)