Блочная модель. Позиционирование элементов.

margin

Свойство margin позволяет установить пространство, которое окружает элемент. margin находятся за пределами любых границ и полностью прозрачны в цвете. Они могут использоваться для позиционирования элементов в конкретном месте на странице или добавить пустое пространство, сохраняя все другие элементы на безопасном расстоянии.

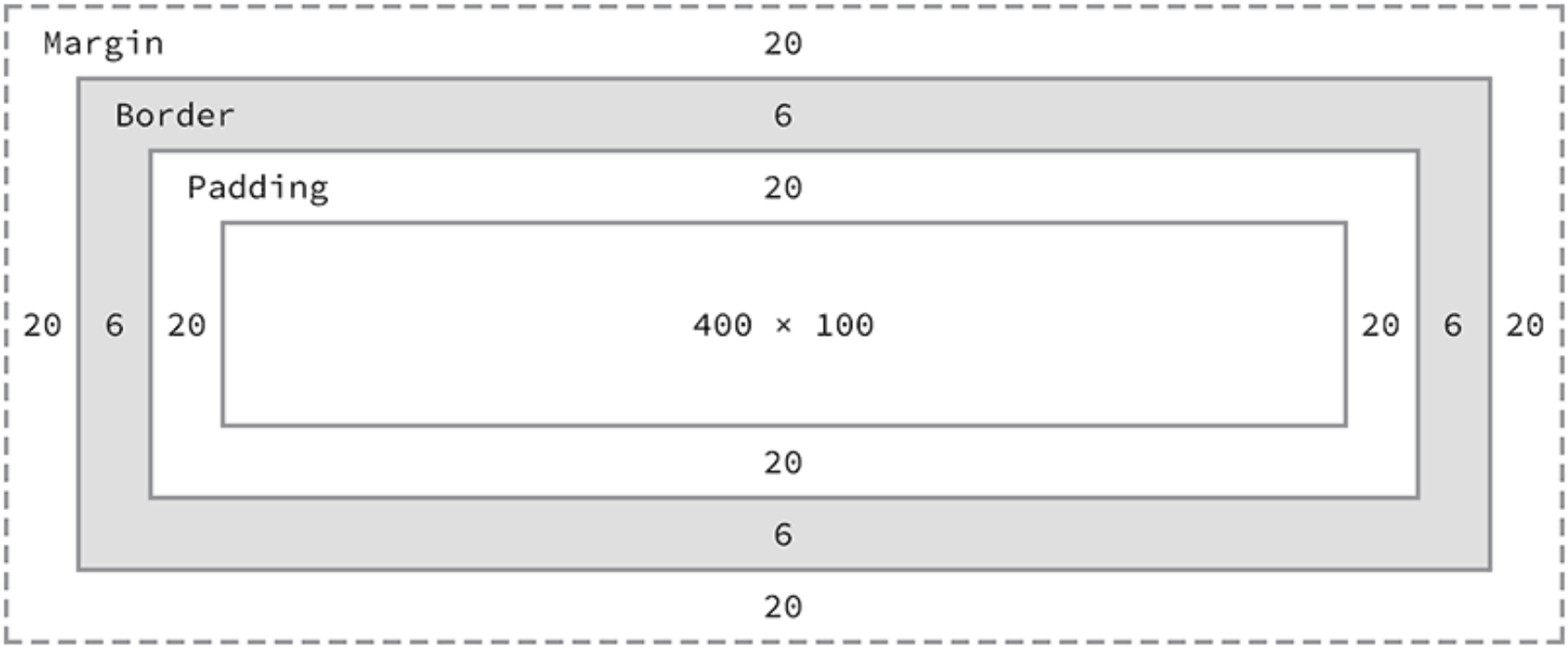
padding

Свойство padding очень похоже на свойство margin, однако располагается внутри границ элемента. Свойство padding используется, чтобы задать пространство непосредственно внутри элемента.

Границы

Для свойства border требуется три значения: ширина, стиль и цвет. В обычной записи эти три значения могут быть разбиты по свойствам border-width, border-style, border-color, border-top, border-right, border-bottom и border-left, border-radius. Обычная запись полезна для изменения или переписывания отдельного значения границы.

Размеры блока



CSS3 ввёл свойство box-sizing, которое позволяет нам точно менять, как блочная модель работает и как вычисляются размеры элемента. Это свойство принимает три основных значения — content-box, padding-box и border-box, каждое из которых оказывает несколько разное влияние на вычисление размера блока.

content-box

Значение content-box является значением по умолчанию, оставляя блочную модель в качестве аддитивной

padding-box

Значение padding-box изменяет блочную модель путём включения всех значений свойства padding внутри width и height элемента. При использовании значения padding-box, если у элемент width равно 400 пикселей и padding 20 пикселей вокруг всех сторон, фактическая ширина останется 400 пикселей. При увеличении padding на любое значение, размер содержимого внутри элемента сжимается пропорционально.

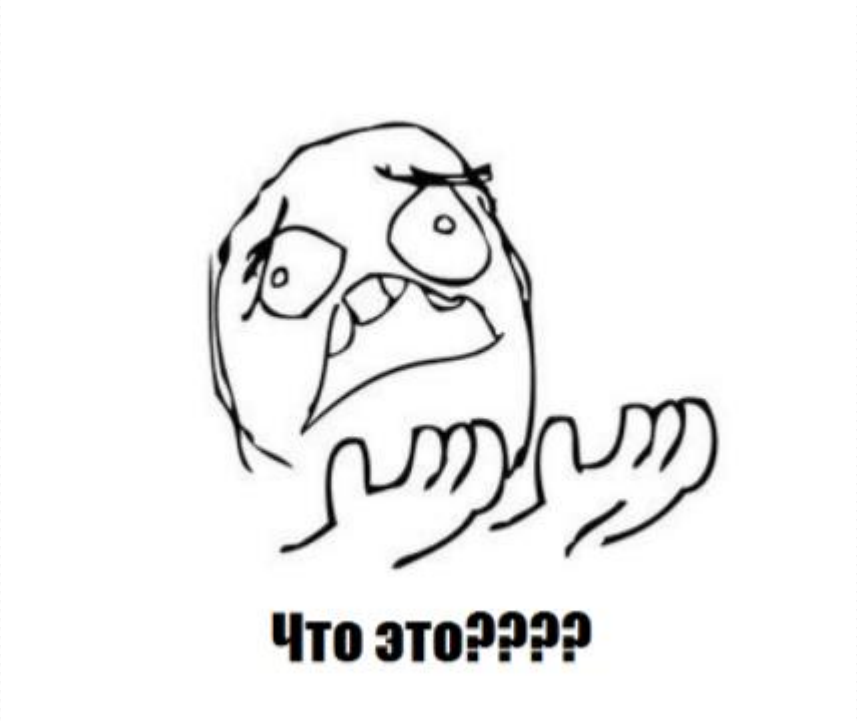
Если мы добавим border или margin, эти значения будут добавлены к свойствам width или height для вычисления полного размера блока. Например, если мы добавим border 10 пикселей и padding 20 пикселей вокруг всех сторон элемента, у которого width 400 пикселей, фактическая полная ширина будет 420 пикселей.

border-box

Наконец, значение border-box изменяет блочную модель так, чтобы любые значения свойств border или padding включались внутри width и height элемента. При использовании значения border-box, если для элемента указаны width 400 пикселей, padding 20 пикселей и border 10 пикселей вокруг всех сторон, фактическая ширина останется 400 пикселей.

Если мы добавим margin, эти значения должны учитываться для расчёта полного размера блока. Независимо от того, какое значение свойства box-sizing применяется, любые значения margin не будут добавляться в вычисления полного размера элемента.

Поток документа



Поток документа

Порядок отображения элементов на странице называется ***потоком документа***. Блочные элементы отображаются как прямоугольные области, идущие друг за другом сверху вниз, а строчные элементы располагаются сверху вниз и слева направо и при необходимости переносятся на новую строку.

Элементы можно вкладывать друг в друга. Чем раньше в коде расположен элемент, тем выше он расположен на странице.

Позиционирование через float

Один из способов позиционирования элементов на странице — через свойство float. Это свойство довольно универсально и может применяться разными путями.

По существу, свойство float берёт элемент, убирает его из обычного потока страницы и позиционирует слева или справа от родительского элемента. Все остальные элементы на странице будут обтекать такой элемент. Например, абзацы будут обтекать изображение, если для элемента <img> установлено свойство float.

Когда свойство float применяется к нескольким элементам одновременно, это даёт возможность создать макет с обтекаемыми элементами расположенными рядом или напротив друг друга

Для справки, обтекаемые элементы располагаются по краю родительского элемента. Если нет родителя, обтекаемый элемент будет располагаться по краю страницы.

Когда мы устанавливаем элемент обтекаемым, то убираем его из обычного потока HTML-документа. Это приводит к тому, что ширина этого элемента по умолчанию становится шириной его содержимого. Иногда, например, когда мы создаём колонки для многократно используемого макета, такое поведение нежелательно. Это можно исправить путём добавления свойства width с фиксированным значением для каждой колонки. Кроме того, чтобы обтекаемые элементы не соприкасались друг с другом, в результате чего содержимое одного элемента располагается рядом с другим, мы можем использовать свойство margin, чтобы установить пространство между элементами.

float могут изменить значение display у элемента!!!

Для обтекаемого элемента также важно понимать, что элемент удаляется из обычного потока страницы и что у элемента может измениться значение display, заданное по умолчанию. Свойство float опирается на то, что у элемента значение display задано как block и может изменить значение display у элемента по умолчанию, если он ещё не отображается как блочный элемент.

Например, элемент, у которого display указан как inline, такой как строчный , игнорирует любые свойства height или width. Однако, если для строчного элемента указать float, значение display изменится на block и тогда элемент уже может принимать свойства height или width.

Когда мы применяем float для элемента, то должны следить за тем, как это влияет на значение свойства display.

Позиционирование с помощью inline-block

Пример_4

Уникальное позиционирование элементов

Свойство position определяет, как элемент позиционируется на странице и будет ли он отображаться в обычном потоке документа. Оно применяется в сочетании со свойствами смещения блока — top, right, bottom и left, которые точно определяют, где элемент будет расположен путём перемещения элемента в разных направлениях.

По умолчанию у каждого элемента значение position установлено как static, это означает, что элемент существует в обычном потоке документа и не принимает какие-либо свойства для его смещения. Значение static наиболее часто переписывается значением relative или absolute

Относительное позиционирование

Значение relative для свойства position позволяет элементам отображаться в обычном потоке страницы, резервируя место для элемента как предполагалось и не позволяя другим элементам его обтекать. Однако, оно также позволяет модифицировать положение элемента с помощью свойств смещения.

Пример_7

Абсолютное позиционирование

Значение absolute для свойства position отличается от значения relative тем, что элемент с абсолютным позиционированием не появляется в обычном потоке документа, исходное пространство и положение абсолютно позиционируемого элемента не резервируется.

Кроме того, абсолютно позиционируемые элементы перемещаются относительно их ***ближайшего относительно позиционированного родительского*** элемента. Если относительно позиционированного родителя не существует, то абсолютно позиционированный элемент будет позиционироваться относительно элемента <***body***>

Абсолютное позиционирование

possition: fixed

По своему действию это значение близко к absolute, но в отличие от него привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы.

Пример_9

Работа с типографикой

```
p {
    font-family: 'Myriad Pro Regular', sans-serif;
    font-size: 34px;
    font-style: italic;
    font-weight: bold;
    line-height: 40px;
    text-align: right;
    text-decoration: underline;
    text-transform: uppercase;
    letter-spacing: 10px;
    word-spacing: 20px;
}
```

Медиа-запросы

Есть несколько разных способов применения медиа-запросов — с помощью правила @media в существующей таблице стилей, импорта новой таблицы стилей через правило @import и путём ссылки на отдельную таблицу стилей внутри HTML-документа. Вообще говоря, рекомендуем использовать правило @media внутри существующей таблицы стилей, чтобы избежать каких-либо дополнительных HTTP-запросов.

```
<!-- Отдельный CSS-файл -->
<link href="styles.css" rel="stylesheet" media="all and (max-width: 1024px)">
/* Правило @media */
@media all and (max-width: 1024px) {...}
/* Правило @import */
@import url(styles.css) all and (max-width: 1024px) {...}
```

Каждый медиа-запрос может включать в себя тип носителя, за которым следует одно или несколько выражений. Основные типы носителей включают в себя all, screen, print, tv и braille. Спецификация HTML5 содержит новые типы носителей, включая даже 3d-glasses. Если тип носителя не может быть указан, медиа-запросом по умолчанию будет screen.

Выражение медиа-запроса, которое следует за типом носителя может включать в себя различные мультимедийные функции и значения, которые затем считаются истиной или ложью. Когда медиа-функция и значение истинны, стили применяются. Если медиа-функция и значение ложны, то стили игнорируются.

Логические операторы в медиа-запросах

Логические операторы в медиа-запросах помогают построить мощные выражения. Существуют три разных логических оператора, доступных для использования в медиа-запросах: and, not и only.

Использование логического оператора and в медиа-запросе позволяет добавить дополнительное условие и убедиться, что браузер или устройство одновременно выполняет а, б, в и т. д. Несколько отдельных медиа-запросов могут быть разделены запятой, действуя как негласный оператор or (или). В приведённом ниже примере выбираются все типы носителей с шириной между 800 и 1024 пикселей.

```
@media all and (min-width: 800px) and (max-width: 1024px) {...}
```

Логический оператор not отрицает запрос с указанием любого запроса, но только одного. В приведённом ниже примере выражение применяется к любому устройству, у которого нет цветного экрана. Чёрно-белые или монохромные экраны, к примеру, применяться будут.

```
@media not screen and (color) {...}
```

Логический оператор only представляет собой новый оператор и не распознаётся браузерами, использующих алгоритм HTML4, что позволяет скрыть стили от устройств или браузеров, которые не поддерживают медиа-запросы. Выражение ниже выбирает только экраны в портретной ориентации на устройствах, способными работать с медиа-запросами.

```
@media only screen and (orientation: portrait) {...}
```

Медиа-функции в медиа-запросах

Медиа-функции определяют, какие атрибуты или свойства будут направлены в выражение медиа-запроса.

height и width

Одна из наиболее распространённых медиа-функций вращается вокруг определения высоты или ширины области просмотра устройства или браузера. Высота и ширина могут быть найдены с помощью медиа-функций height, width, device-height и device-width. Каждая из этих медиа-функций также может быть использована с префиксом min или max, вроде min-width или max-device-width.

Функции height и width базируются на высоте и ширине области визуализации, окна браузера, к примеру. Функции device-height и device-width с другой стороны основаны на высоте и ширине выходного устройства, которое может быть больше, чем фактическая область визуализации. Значением этих медиа-функций может быть любая единица длины, относительная или абсолютная.

```
@media all and (min-width: 320px) and (max-width: 780px) {...}
```

orientation

Медиа-функция orientation определяет, находится ли устройство в альбомной (landscape) или портретной (portrait) ориентации. Режим landscape срабатывает, когда дисплей шире, чем высота, режим portrait срабатывает, когда высота дисплея больше ширины. Эта медиа-функция играют роль в основном с мобильными устройствами.

```
@media all and (orientation: landscape) {...}
```

aspect-ratio

Функции aspect-ratio и device-aspect-ratio определяют соотношение ширины к высоте в пикселях целевой области визуализации или устройства вывода. Префиксы min и max доступны для использования с различными функциями, выявляя соотношения выше или ниже того, о котором говорится.

Значение функции состоит из двух положительных целых чисел, разделённых косой чертой. Первое число задаёт ширину в пикселях, а второе число задаёт высоту в пикселях.

```
@media all and (min-device-aspect-ratio: 16/9) {...}
```

pixel-ratio

Кроме медиа-функции aspect-ratio есть также функция pixel-ratio. Она включает функцию device-pixel-ratio, также с префиксами min и max. В частности, эта функция отлично подходит для определения устройств высокой чёткости, в том числе дисплеев ретина. Медиа-запрос для этого выглядит следующим образом.

```
@media only screen and (-webkit-min-device-pixel-ratio: 1.3), only screen and (min-device-pixel-ratio: 1.3) {...}
```

resolution

Медиа-функция resolution определяет разрешение устройства вывода в виде плотности пикселей, также известной как число точек на дюйм или DPI. Функция resolution также принимает префиксы min и max. Кроме того, функция resolution принимает число точек на пиксель (1.3dppx), точек на сантиметр (118dpcm) и другие размеры на основе значений разрешения.

```
@media print and (min-resolution: 300dpi) {...}
```

Другие медиа-функции

Другие функции включают в себя определение доступных выходных цветов с помощью color, color-index и monochrome, выявление растровых устройств через функцию grid, а также определение процесса сканирования телевизора функцией scan. Эти функции менее распространены, но столь же полезны при необходимости.

Пример_11