CSS: Отступы, background, box-sizing

Мы уже познакомились с тем, что браузер интерпретирует (видит, воспринимает) тэги, как прямоугольники разного размера. Но пока нам показалось, что он вокруг каждого тга рисует только один прямоугольник, не так ли? На самом деле, он рисует целых 4(!), да-да, четыре прямоугольника вокруг каждого тэга. Зачем же ему так много лишней работы?

Давайте сначала посмотрим, что же за прямоугольники он рисует:

    Первый прямоугольник

        Итак, контент браузер должен

        вложить в три коробки: контент в

        паддинг, паддинг  в бордер, бордер в марджин.

        Браузер вкладывает контент каждого тэга в коробки,

        которые помогут нам расположить

         блоки относительно друг друга.

Graphical user interface, text, application

Description automatically generated

Давайте посмотрим, как дружище-браузер позаботился о счастье верстальщика.

A picture containing logo

Description automatically generated

            Допустим, нам нужно создать кнопку, прямоугольную,

            с белой границей и текстом,

            расположенным по центру кнопки.

            Кроме того, кнопка должна сама располагаться по центру черного блока.

В html создаем разметку, основой для кнопки нам послужит ссылка, вложенная в див:

Вот что получилось в редакторе (слева) и в браузере (справа)

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Пока в браузере не очень похоже, правда? Давайте добавим немного стилей для дива и для ссылки

Text

Description automatically generated

​

Пока в браузере не очень похоже, правда? Давайте добавим немного стилей для дива и для ссылки

Text

Description automatically generated

​

Shape, rectangle

Description automatically generated

        В браузере уже получилась что-то, больше похожее на нашу кнопку.

        Ей мы можем добавить еще стилей и выровнять текст по центру.

        Кроме того, мы зададим ей ширину.

 Кроме того, мы зададим ей ширину.

        Так как ссылка - это строчный элемент, ширина, высота, отступы на нее не действуют, поэтому мы изменим ее отображение на строчно-блочное, и тогда на нее будут действовать и свойства текста, и свойства блоков, например, мы сможем применить свойства ширины(свойство для блоков).

Text

Description automatically generated

Text

Description automatically generated with medium confidence

            Текст кнопки выровнялся

            по центру блока, но ей все

            еще не хватает **границы и отступов.**

Text

Description automatically generated

        Вот тут-то мы и воспользуемся

        коробками,припасенными браузером для

        нашей кнопки. Добавим ей свойство margin, например:

​запишем в стилях margin:20px. И вот, как по волшебству, кнопка отодвинулась от границ родительского блока.

Graphical user interface

Description automatically generated with low confidence

​     Если посмотреть ее в инспекторе, то видно, что

     кнопка (подсвечена голубым) имеет размер 150px

     в ширину (мы задали ее в стилях), текст расположен по центру,

     а вокруг кнопки с появился внешний отступ (margin) размером в 20

Logo

Description automatically generated

        Добавьте кнопке еще свойство border:

        2px solid #fff (граница: 2 пикселя сплошная белая)

        и полюбуйтесь результатом:

        кнопка почти как на дизайне!

        Сейчас только граница расположена слишком близко

к тексту, и нам бы добавить еще один отступ от текста до границы. Спасибо браузеру, он припас для нас паддинг - добавим в стилях свойство padding:10px;

Тройное УРА! Кнопка получилась совсем как на дизайне! Мы радуемся, но понимаем, что мы не за что бы не справились так быстро и просто, если бы не наш друг, который заранее положил коробку с кнопкой в три обертки: margin, border и padding.

Graphical user interface, diagram

Description automatically generated

        Теперь давайте запомним все три замечательные свойства:

**Паддинг** - самый ближайший к содержимому отступ.

        Отступ от содержимого до границы блока

**Бордер** - граница блока.

**Марджин**  - отступ от границы блока до других блоков.

Обратите внимание, что паддинг и граница увеличивают ширину и высоту блока, но у строчных элементов они не отодвигают от себя содержимое сверху и снизу. У строчных элементов нет свойств высоты, потому что их высота - это высота строки. Поэтому у строчных элементов паддинг отображается, но не отодвигает его от блоков выше или ниже.

**Background**

Свойство background задает фон элемента. Это очень богатое свойство, у него есть много возможностей.

* **background-color - цвет фона.**
* **background-image - картинка фона.**
* **background-position - позиция фоновой картинки.**
* **background-repeat  - повторение.**
* **background-size - размер фоновой картинки**

**Помимо вышеперечисленных, также иногда используются:**

* **background-origin определяет область позиционирования фонового рисунка**
* **background-clip определяет, как цвет фона или фоновая картинка должна выводиться под границами. Эффект заметен при прозрачных или пунктирных границах**
* **background-attachment определяет является ли фоновое изображение фиксированным или прокручивается вместе со страницей**

Почти все эти свойства можно задать одним свойством background:  
**background: #fc0 url(путь к картинке) 20px 50% no-repeat;**

Свойство background определяет цвет фона, путь к фоновой картинке, позиционирование фоновой картинки, и правило указывающее что фоновая картинка не должна повторяться

Помимо традиционных background-color, background-image, background-repeat, background-position, часто в верстке используется свойство background-size.

Background-size может иметь следующие значения:

background-size: 20px 20px -- задаёт размер в пикселях.

background-size: 100% задаёт размер фоновой картинки в процентах от ширины элемента.

**аuto** если задано одновременно для ширины и высоты (auto auto), размеры фона остаются исходными; если только для одной стороны картинки (100px auto), то размер вычисляется автоматически исходя из пропорций картинки.

**сover** - “заливает” весь элемент картинкой.

**сontain** уменьшает или увеличивает картинку, чтобы картинка целиком поместилась внутрь блока.

**box-sizing**

Данное css-правило применяется для изменения принципа расчета ширины и высоты элемента.

Согласно спецификации CSS ширина блока складывается из ширины контента (width), значений отступов (margin), полей (padding) и границ (border). Аналогично обстоит и с высотой блока. Свойство box-sizing сказать браузеру, что границы следует не прибавлять к ширине, а отнимать от нее.

**box-sizing может иметь следующие значения:**

**сontent-box**  свойства width и height задают ширину и высоту контента, но не включают в себя значения отступов, полей и границ, это свойство используется практически во всех браузерах по умолчанию.

**border-box** свойства width и height включают в себя значения полей (padding) и границ (border), но не отступов (margin).

Например, двум дивам  заданы следующие css правила:

  .box1 {  width: 300px;  padding: 10px;    border: 2px solid #000;   }

   .box2 {  width: 300px;  padding: 10px;    border: 2px solid #000;  box-sizing: border-box;  }

Див с классом  box1 будет иметь ширину 324px эта ширина складывается из значения width + по 10 пикселей паддинги справа и слева + по 2 пикселя ширина правой и левой границ т.е. 300+2\*10+2\*2 = 324 пикселя

Див с классом  box2 будет иметь ширину 300px так как для него задано значение  box-sizing: border-box;