**CSS基础（12-31）**

**css语法**

CSS 规则集由选择器和声明块组成；选择器指向你要设置样式目标元素。声明块由 { 开始，} 结束。声明块包含一条或多条声明, 声明之间使用 ;分隔。每条声明都包含一个 CSS 属性名称和一个值，以 : 分隔。

**css引入方式**

* **外部 CSS**

要使用外部 CSS ，首先要创建 CSS 文件。一个 CSS 文件要以 .css 扩展名保存；

eg：<link rel="stylesheet" href="styles.css" />

<link> 标签里面，我们用属性 rel，让浏览器知道有 CSS 文档存在（所以需要遵守 CSS 样式的规定），并利用属性 href 指定，寻找 CSS 文件的位置。

* **内部 CSS**

内部 CSS 样式，是在 <head> 内使用 <style> 标签定义。

* **行内 CSS**

行内样式（也称内联样式）可用于为单个元素应用唯一的样式。它的使用方法是直接将 style 作为属性，添加到目标元素上。

**如果我们同时使用多种 CSS ，它们的优先级为：**

1. 行内样式；
2. 外部和内部样式表（优先级取决于所在位置，后面的覆盖前面的)；
3. 不引入任何 CSS 时，浏览器默认的样式。

**css选择器**

* **基本选择器**

1. 通用选择器

选择页面上所有元素。语法：\*

1. 元素选择器

按照给定的元素名称，选择所有匹配的元素。语法：elementname

eg：/\* 匹配所有 <h1> 元素 \*/

h1 {

}

1. 类选择器

类选择器就是按照给定的 class 属性的值，选择所有匹配的元素。类选择器以 . 开头，后面跟类名。语法：.classname

/\* 匹配所有 class 属性中含有 "one" 类的元素 \*/

.one {

}

1. ID 选择器

按照 id 属性选择一个与之匹配的元素。需要注意的是，一个文档中，每个 ID 属性都应当是唯一的。id 选择器以 # 开头，后面跟 id 值。语法：#idname

/\* 匹配 id 为 "one" 的元素 \*/

#one {

}

1. 属性选择器

按照给定的属性，选择所有匹配的元素。属性选择器，使用 [ ] 选取带有指定属性的元素。语法：[attr]

* **分组选择器**

1. 选择器列表

选择器列表是将不同的选择器组合在一起的方法，常被称为并集选择器或并集组合器。选择器列表使用 , 分隔的列表来对选择器进行分组。语法：A, B

/\* 同时匹配 <div> 元素和 <span> 元素。 \*/

div, span {

}

* **组合器**

1. 后代组合器

选择前一个元素的后代节点。语法：A B

/\* 匹配所有位于任意 <div> 元素之内的 <span> 元素。 \*/

div span {

}

1. 直接子代组合器

选择前一个元素的直接子代的节点。语法：A > B

/\* 匹配直接嵌套在 <ul> 元素内的所有 <li> 元素。 \*/

ul > li {

}

1. 一般兄弟组合器

选择兄弟元素，也就是说，后一个节点在前一个节点后面的任意位置，并且共享同一个父节点。语法：A ~ B

/\* 匹配同一父元素下，<p> 元素后的所有 <span> 元素。 \*/

p ~ span {

}

1. 紧邻兄弟组合器

选择相邻元素，即后一个元素紧跟在前一个之后，并且共享同一个父节点。语法：A + B

/\* 匹配所有紧邻在 <p> 元素后的 <span> 元素。 \*/

p + span {  
}

* **伪选择器**

1. 伪类

CSS 伪类是添加到选择器的关键字，指定要选择的元素的特殊状态。

/\* 匹配选择具有焦点的 <input> 元素。 \*/

input:focus {

}

1. 伪元素

伪元素是一个附加至选择器末的关键词，允许你对被选择元素的特定部分修改样式。

/\* 匹配所有 <p> 元素的第一行。 \*/

p::first-line {

}

**css 盒模型**

CSS 中组成一个块级盒子需要：

1. Content box: 这个区域是用来显示内容，大小可以通过设置 width 和 height.
2. Padding box: 包围在内容区域外部的空白区域；大小通过 padding 相关属性设置。
3. Border box: 边框盒包裹内容和内边距。大小通过 border 相关属性设置。
4. Margin box: 这是最外面的区域，是盒子和其他元素之间的空白区域。大小通过 margin 相关属性设置。

* **标准盒模型**

在标准模型中，如果你给盒设置 width 和 height，实际设置的是 content box。padding 和 border 再加上设置的宽高一起决定整个盒子的大小。

.box {

width: 350px;

height: 150px;

margin: 25px;

padding: 25px;

border: 5px solid black;

}

（margin 不计入实际大小 ，盒子的范围到边框为止 —— 不会延伸到 margin。）

* **替代（IE）盒模型**

使用这个模型，所有宽度都是可见宽度，所以内容宽度是该宽度减去边框和填充部分。也就是说在替代（IE）盒模型下，你设置 width 属性，已经不仅仅是 content 的宽度，还包含了 padding 和 border 的大小。

* **改变盒模型**

如果你希望所有元素都使用替代模式，而且确实很常用，设置 box-sizing 在 元素上，然后设置所有元素继承该属性。

html {

box-sizing: border-box;

}

/\* 使所有元素继承该属性 \*/

\*,

\*::before,

\*::after {

box-sizing: inherit;

}

* **外边距（margin）**

margin 属性接受 1~4 个值 。每个值可以是 <length>，<percentage>，或 auto。

1. 指定四个值时，依次（顺时针方向）作为上边，右边，下边，和左边的外边距。
2. length 以固定值为外边距。
3. percentage 相对于包含块的宽度，以百分比值为外边距。
4. auto 让浏览器自己选择一个合适的外边距。有时，在一些特殊情况下，该值可以使元素居中。

* **内边距（padding）**

padding 属性接受 1~4 个值。每个值可以是 <length> 或 <percentage>。取值不能为负。

1. 指定四个值时，依次（顺时针方向）作为上边，右边，下边，和左边的内边距。
2. length 以固定值为内边距。
3. percentage 相对于包含块的宽度，以百分比值为内边距。

**css背景和边框**

* **背景颜色**

background-color 属性用于定义元素的背景颜色，接受有效的色彩值。 背景色扩展到元素的内容（content）和内边距（padding）

* **背景图像**

background-image 属性用于定义元素的背景图像，默认情况下图像是平铺重复显示，覆盖元素背景

.bgi {

background-image: url(./img/wtflogo.png);

}

* **背景平铺**

background-repeat 属性用于控制图像的平铺行为。常用的属性值包含以下几个：

1. repeat：图像在水平方向与垂直方向重复(默认)
2. repeat-x：图像在水平方向重复
3. repeat-y：图像在垂直方向重复
4. no-repeat：图像仅平铺一次

* **背景大小**

background-size ，它可以设置长度或百分比值，使背景的大小以适应被填充背景的元素尺寸，常用的属性值包含以下几个：

1. 指定长度或百分比值：按指定大小渲染
2. auto：以图像的比例缩放作为背景，图像可能会重复展示
3. cover：图像扩展至足够大，使其完全覆盖整个区域，同时保持其高宽比，图像可能会被裁剪
4. contain：图像尽可能地缩放并保持宽高比例以适应整个背景区域。缩放会导致背景可能出现部分空白，空白区域会显示由background-color 设置的背景颜色。

* **背景定位**

background-position 用于设置背景图像的起始位置。框的左上角是 (0,0)，框沿着水平 (x) 和垂直 (y) 轴定位。常见的设置方法如下：

1. 可用任何长度单位，第二个位置(即Y轴方向)如果不声明，默认是50%(两个位置都不设置的话默认0% 0%)
2. 位置关键字（left/right/top/bottom/center），可单双使用（第二个关键字如果不声明默认是center）
3. 混合使用

* **背景附加**

background-attachment属性用于设置背景图像是否固定或者随着页面的其余部分滚动。 常见的属性值如下：

1. scroll：背景图片随网页滚动而移动(默认)
2. fixed：背景图片不会随网页滚动而移动
3. local： 背景图片会随着元素内容的滚动而滚动。

* **背景简写**

background属性可以实现简写，比单个子属性声明要简洁得多，可少写很多代码。

background: color image repeat attachment position/size

* **边框**

border 可以为元素的所有四个边设置边框。也可以将border拆分为border-top, border-bottom, border-left, border-right，单独设置一条边。

border由width、style、color三个属性组成，其中width表示边框宽度，style表示边框样式（可以是实线、圆点、虚线等样式），color表示边框颜色。

.solid {

border: 3px solid red;

// 等价于

border-width: 3px;

border-style: solid;

border-color: red;

}

* **圆角**

border-radius 属性用于设置方框的圆角。可以使用两个长度或百分比作为值，第一个值定义水平半径，第二个值定义垂直半径

.radius {

border: 3px solid blue;

border-radius: 20px;

}

**文本方向和内容溢出**

* **书写模式**

CSS 中的书写模式是指文本的排列方向是横向还是纵向的，使用writing-mode设置

writing-mode的三个值分别是：

1. horizontal-tb: 块流向从上至下。对应的文本方向是横向的。
2. vertical-rl: 块流向从右向左。对应的文本方向是纵向的。
3. vertical-lr: 块流向从左向右。对应的文本方向是纵向的。

* **文本方向**

除了书写模式，我们还可以设置文本方向。有些语言（如阿拉伯语）是横向书写的，但是是从右向左。可以使用 direction 属性设置文本方向。

direction的常用属性值如下：

1. ltr：默认，left to right，从左到右
2. rtl：right to left，从右到左

* **溢出内容**

overflow属性是控制元素溢出的方式，overflow的默认值为visible，所以内容溢出的时候，我们在默认情况下可以看到它们。

1. visible：默认值。内容不会被修剪，会呈现在元素框之外
2. hidden：内容修剪，其余内容是不可见的
3. scroll：内容修剪，浏览器会显示滚动条
4. auto：如果内容被修剪，则浏览器会显示滚动条

.overflow > div {

height: 10em;

width: 15em;

border: 2px solid blue;

}

.hidden {

overflow: hidden;

}

.scroll {

overflow: scroll;

}

.auto {

overflow: auto;

}

*以上内容是总结/摘选自学习网站：https://github.com/WTFAcademy/WTF-CSS。*