**HTML基础—基本概念**

**1.**什么是浏览器？

功能：（1）将网页渲染出来让用户查看（2）并让用户与网页交互的一种软件

主流浏览器：IE（内核：Trident） 火狐（内核：Gecko） 谷歌（内核：Webkit/Blink） Safari（内核：Webkit） 欧朋（内核：Presto）

因为内核不同，所以渲染的网页有所不同

**2.**什么是服务器？

超级计算机，24小时不断电不关机，专门用于存储数据

**3.**什么是URL?

我们在浏览器的地址栏输入的就是URL，eg：<http://127.0.0.1:80/index.html>

http:// URL协议类型

127.0.0.1 服务器IP地址

:80 服务器的端口号

/index.html 需要访问的资源名称

**4.HTML（超文本标记语言）**

**作用：专门用来描述文本语义的，也就是说告诉浏览器哪些是标题哪些是段落。**

**注意事项：HTML作用只有一个专门给文本添加语义的，不是用来修改文本样式的。**

**<h1>标签的作用是告诉浏览器哪些文字是标题，即给指定文字添加标题语义的**

.html是纯文本文件

**5.乱码问题：**（1）基本上都保存为UTF-8:<meta charset=”UTF-8”>（2）保存网页的字符集和网页中指定的字符集设置为一致

**6.**标签分类：

1. 双标签 <html></html> <title></title> <body></body>……

单标签 <meta> <br>……

1. 兄弟关系 <head></head><body></body>

嵌套关系 <head><title></title></head>

**7.**DTD文档声明：不是HTML标签，必须在第一行，不区分大小

<!DOCTYPE html>：HTML5的DTD文档声明是上下兼容的

**HTML相关标签**

**8.**<hn>标签：<h1>~<h6>

注意点：

（1）H标签是用来给文本添加标题语义的

（2）被H系列标签包裹的内容会独占一行

（3） 在企业开发中一般情况下一个界面只能出现一个H1标签

**9.**<p>标签：

作用：告诉浏览器哪些文字是一个段落

注意点：

在浏览器中会独占一行

**10．**<hr>标签：分割线

注意点：

在浏览器中会独占一行

**11.**img标签

（1）格式：<img src=””alt=””title=”” >

（2）注意点：

如果我们不指定img标签的宽度和高度，那么系统按照默认图片宽度和高度

如果我们手动指定了Img标签的宽度和高度，有可能导致图片变形。

如果想指定img标签的宽度和高度，又不想图片变形，我们只能只指定宽度或高度其中一个

（3）常用属性：

src：用来显示图片的路径

title：用来告诉浏览器，当鼠标悬停在图片上时，需要弹出的描述框中显示什么内容

alt:当需要显示的图片找不到的时候显示什么内容

**12.**<br>：用于换行

注意点：

多个br标签可以同时使用由于HTML是用于给文本添加语义的，而br标签的语义不是另起一个段落，所以在企业开发中很少使用它

**13.**路径问题

给src属性赋值有两种方式：

（1）相对路径:每次都是从.html文件所在的文件夹开始查找

同级：图片和.html文件存储在同一个文件夹中

格式：src=”xxx.xxx”

下级：存储图片的文件夹和.html文件在同一个文件夹中

格式：src=”xxx/xxx.xxx”

上级：存储图片的位置和存储代码的文件夹在同一个文件夹中

格式：src=”../xxx.xxx”

（2）绝对路径:每次都是从指定的盘符开始查找

注意点：

企业开发中如果需要编写路径，统一使用反斜杠/

企业开发中一般不使用绝对路径，因为可移植性不好

**14.**a标签：用于控制页面与页面之间的跳转

（1）格式：<a href=””></a>

（2）注意点：

a标签不仅可以让文字可以点击，也可以让图片能够被点击

a标签必须有一个href属性

如果通过a标签的href属性指定一个URL地址，那么必须在地址前加上http://或https://

a标签的href属性除了可以指定一个网址，还可以指定一个本地地址

（3）常用属性：

target（跳转方式）：\_self:在当前选项卡中跳转，默认。\_blank:在新的选项卡中跳转

title：用来告诉浏览器，当鼠标悬停在链接上时，需要弹出的描述框中显示什么内容

（4）假链接：点击之后不会跳转，企业开发前期，其他页面都没有写出来，就用假链接来代替

格式：<a href=”#”>会自动回到网页顶部

<a href=”javascript”>不会自动回到网页顶部

**15.**base标签：专门用来统一的指定当前网页中所有的超链接需要如何打开

格式：<base target=””>

注意点：如果base和a标签同时指定了打开方式，网页的打开方式由a标签决定

**16.**锚点：

(1)要想通过a标签跳转到当前界面的指定位置，那么必须告诉a标签一个独一无二的id，这样才能在当前界面找到需要跳转到的目标位置

格式：<a href=”zhongbu”><h2>我是顶部</h2></a>

<h2 id=”zhongbu”>我是中部</h2>

<a href=”dibu”> 跳转到底部</a>

<h2 id=”dibu”>我是底部</h2>

(2)想通过a标签跳转到其他界面的指定位置

格式： <a href=”新界面的网址#123”>跳转到xxx</a>

<h2 id=”123”>欢迎来到这里</h2>

**17.**列表标签

（1）无需列表：ul

作用：给一堆数据添加列表语义，并且这一堆数据没有先后之分

格式：<ul><li></li></ul>

应用场景：新闻列表 商品列表 导航条

（2）有序列表：ol

作用：给一堆数据添加列表语义，并且这一堆数据有先后之分

格式：<ol><li></li></ol>

（3）定义列表：dl

作用：给一堆数据添加列表语义，先通过dt标签定义列表中的所有标题，然后在通过dd标签给标题添加描述信息

格式：<dl><dt></dt><dd></dd></dl>

dt:定义列表的标题 dd:描述标题对应的含义

应用场景：网页的导航条 做网站尾部的相关信息 做图文混排

**18.**表格标签：<table></table>

（1）作用：给一堆数据添加表格语义

（2）格式：<table><caption></caption><tr><td></td></tr></table>

tr：表示一行 td:表示一行的一个单元格

th：专门用来存储每一列的标题，内容会自动加粗居中

caption:专门用来指定表格标题

（3）注意点：表格边框属性border默认为0

**常用属性：（1）**宽度和高度（table、td）

表格默认宽高度是根据内容的尺寸调整的，也可以通过table标签设置width/height属性来手动设置

如果给td标签设置width/height属性，只会改变单元格的宽度和高度，不会影响整个表格的宽度和高度

**（2）**水平对齐（table、td、tr）

给table标签设置align属性，可以控制表格在水平方向的对齐方式

给tr标签设置align属性，可以控制当前行中的所有单元格的水平方向对齐方式

给td标签设置align属性，可以控制当前行中的单元格内容的水平方向对齐方式

注意点：如果同时给tr、td设置align属性，那么单元格中的内容会安照td的设置来对齐

**（3）**垂直对齐（td、tr）

给tr标签设置设置valign属性,可以控制当前行中的所有单元格的垂直方向对齐方式

给td标签设置设置valign属性,可以控制当前行的单元格内容的垂直方向对齐方式

注意点：如果同时给tr、td设置valign属性，那么单元格中的内容会安照td的设置来对齐

**（4）**外边距：单元格与单元格之间的距离

默认情况下cellspacing=2px

**（5）**内边距:单元格的边框与内容之间的间隙

默认情况下cellpadding=1px

**（6）**水平方向上的单元格合并：（行上的单元格合并）

给td标签设置一个colspan属性，来指定把一个单元格当作多个单元格

**（7）**垂直方向上的单元格合并：（列上的单元格合并）

给td标签设置一个rowspan属性,来指定把一个单元格当作多个单元格

**细线表格制作方式：**

<table bgcolor="black" cellspacing="1px" >

<tr bgcolor="white" >

<td>1.1</td>

<td>1.2</td>

</tr>

<tr bgcolor="white" >

<td>2.1</td>

<td>2.2</td>

</tr>

表格完整格式：（仅作了解）

<table>

<caption></caption>

<thead>

<tr>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<td></td>

</tr>

</tbody>

<tfoot>

<tr>

<td></td>

</tr>

</tfoot>

</table>

**19.**表单标签： （1）作用：表单是专门用来收集用户信息的

表单标签比较特殊，在浏览器中有自己默认的外观和功能。

（2）格式：

<from><表单元素></from>

（3）常见表单元素：

**input标签（默认有边框）：**它有一个type属性，这个属性有很多类型的取值，取值的不同决定了input标签的功能和外观，除了按钮类型的input标签以外，其他标签都可以通过value属性来指定将来提交到服务器的值

《1》<input type=”text”>明文输入框

《2》<input type=”password”>暗文输入框

《3》<input type=”radio”>单选框

注意点：默认单选框不会互斥，要想选框互斥，要给每个单选框标签设置一个name属性，并且属性值必须相同。要想让单选框默认选中其中一个，那么就给他添加一个checked属性

《4》<input type=”checkbox”>多选框

注意点：默认可以选择多个，要想让单选框默认选中其中一个，那么就给他添加一个checked属性

《5》 按钮

普通按钮：<input type=”button”value=”按钮”>，通过value属性给按钮添加标题

图片按钮：<input type=”img”src=””>，作用与普通按钮一样

重置按钮：<input type=”reset”>，用于清空表单填写的内容

注意点：默认按钮标题是重置，可以通过value属性修改按钮标题

提交按钮：<input type=”submit”>，将表单中填写好的内容提交到远程服务器

注意点：要想把表单中填写的内容提交给远程服务器中，需要做到以下两点：<form action=”网址”></form>；给需要提交的表单元素都添加name属性

《6》隐藏域：<input type=”hidden”>配合提交按钮将一些数据默默的提交到远程服务器

**label标签：**默认情况下，文字与输入框是没有关联关系的，也就是说点击文字输入框不会聚焦，如果想聚焦，需要这样：

方法一：<label for=”一样”>文字:</label><input type=”” id=”一样”>

方法二：<labrl>文字：<input type=””></label>

局限性是不能交叉绑定

datalist标签：给输入框绑定待选项（了解）

格式：

<input type="text" list="cities">

<datalist id=”cities”>

<option>待选项内容</option>

<option>待选项内容</option>

<option>待选项内容</option>

</datalist>

**select标签：**用于定义下拉列表

格式：

<select>

<option>列表数据</option>

<option>列表数据</option>

<option>列表数据</option>

</select>

注意点：下拉列表不能输入内容，但是可以选择内容。也可以设置默认值：<option selected="selected">列表数据</option>

给下拉列表分类：

<optgroup label="分组名称">

<option>列表数据</option>

</optgroup>

**textarea标签：**默认情况下可以无限换行，有默认的宽度和高度，默认情况下输入框是可以手动拉伸的

格式：

<textarea>内容</textarea>

可以通过这样给文本添加列数和行数：<textarea cols=”” rows=””>内容</textarea>

**fieldset标签**：给表单添加边框

**legend标签：**可以给边框指定一个标题

**20.video标签：**

（1）作用： 播放视频

（2）格式： 第一种格式：<video src=””></video>

第二种格式：<video><source src=”” type=”video/webm/ogg/mp4”></suorce></video>

由于视频数据非常的重要，所以五大浏览器厂商都不愿意支持别人的视频格式，所以导致了没有一种视频格式是所有浏览器都支持的，这个时候w3c为了解决这个问题，所以有了第二种格式

（3）常用属性：width/height

autoplay:用来告诉video标签是否需要自动播放视频

controls: 用来告诉video标签是否需要显示控制条

poster：用来告诉video标签视频没有播放之前显示的占用图片

loop：一般用于做广告视频，用于告诉video标签播放结束后是否需要循环播放

preload：预加载视频，与autoplay相冲，如果设置了autoplay属性那么preload就会失效

muted静音：

**21.audio标签：**

（1）作用：播放音频

（2）格式：第一种格式：<audio src=””></audio>

第二种格式：<audio><source src=”audio/mp3”></source></audio>

（3）常用属性：width/height

autoplay:用来告诉audio标签是否需要自动播放视频

controls: 用来告诉audio标签是否需要显示控制条

loop：用于告诉audio o标签播放结束后是否需要循环播放

preload：加载音频，与autoplay相冲，如果设置了autoplay属性那么preload就会失效

muted静音：

**22.**详情和概要标签：

（1）作用：利用summary标签来描述概要信息，利用dateils标签描述详情信息，默认情况下是折叠显示，想看详情信息必须点击

（2）格式：<details><summary>概要信息</summary>详情信息</details>

23.marquee标签：不是w3c推荐的标签，但各个浏览器对他的支持非常好

（1）作用：跑马灯效果，不仅可以让文字滚动也可以让图片滚动

（2）格式：<marquee >内容</marquee >

（3）常用属性：

direction：滚动方向 left/right/up/down

scrollamount:滚动速度

loop:滚动次数，默认是-1，无限次

behavior:滚动类型 slide滚动到边界就停止

alternate滚动到边界就弹回来

**24.**字符实体

&nbsp;空格 &lt;小于号 &gt;大于号 &copy;版权符号

**CSS基础概念**

**1.CSS的作用：**修改样式

格式：

<style type=”text/css”>

选择器名称{

属性名称：属性对应的值；

…

}

</style>

**2.属性**

**《1》规定文字的属性**：

（1）规定文字的样式的属性：

**font-style:**normal(正常的，默认) italic（倾斜）

快捷键：fsn→font-style:normal;

（2）规定文字的粗线的属性：

**font-weight:**lighter（细线，默认）bold（加粗）

取值：数字、单词

快捷键：fwb→font-weight:bold;

（3）规定文字的大小的属性：

**font-size:**30px;

取值：px（像素）

快捷键：fz30→font-size:30px;

（4）规定文字的字体的属性：

**font-family:**微软雅黑;

快捷键：ff→font-family:”微软雅黑”;

注意点：（1）如果取值是中文，需要用双引号或单引号括起来；

（2）设置的字体必须是用户电脑里已经安装的字体；

（3）如果设置的字体不存在，那么系统就会使用默认的字体来显示；

（4）如果不想用默认的字体来显示，可以给字体设置备选方案

font-family: “微软雅黑”，“宋体”…;

（5）如果想给中文和英文分别单独设置字体

但凡中文字体，里面都包含了英文；但凡英文字体，里面都不包含中文字体。

**补充：**企业开发中常用的是字体有：

中文：宋体 黑体 微软雅黑

英文：Times New Roman Arial

**简写：font: style weight size family;**

注意点：style和weight可以调换位置也可以省略，size和family不可以调换位置也不可以省略，size必须放在family后面，size和family必须放在所有属性后面

**《2》规定文本的属性：**

（1）文本装饰属性：

**text-decoration: ；**

取值：underline（下划线）/line-through（删除线）/overline（上划线）、none（什么都没有，常用于超链接去除下划线）

（2）文本水平对齐属性：

**text-align: ；**

取值：right（右）/center（居中）/left（左，默认）

（3）文本缩进属性：

**text-indent: ；**

取值：em

**《3》颜色控制属性：**

**color: ;**

取值：

（1）英文

（2）rgb（ , , ）：rgb其实就是三原色，其中r（red）g（green）b（blue）

rgb（0,0,0）表示黑色，rgb（255,255,255）表示白色，rgb（255,0,0）表示红色，rgb（0,255,0）表示绿色，rgb（0,0,255）表示蓝色

（3）rgba（ , , , ）：rgb其实就是三原色；a表示透明度，取值范围0-1，取值越小就越透明

（4）十六进制#：本质就是rgb，每两位表示一个颜色

（5）缩写十六进制：每个颜色的两位数都是一样的，那么就可以简写

**《4》规定背景属性：**

（1）规定背景颜色属性：

**background-color:**

取值：英文单词 、rgb、rgba、十六进制、缩写十六进制

注意点：同一个标签可以同时设置背景颜色和背景图片，如果颜色和图片同时存在，那么图片会覆盖颜色

（2）规定背景图片属性：

**background-color: url(“”);**

注意点：

图片地址必须放在url()中间，图片可以是本地地址也可以是网络上的地址

如果图片的大小没有背景大，那么它会自动的在水平和垂直方向平铺和填充

（3）规定背景图片平铺方式属性：

**background-repeat:**

取值：repeat:在水平和垂直方向上平铺，默认

repeat-x:水平方向上平铺

repeat-y:垂直方向上平铺

（4）背景定位属性：

**background-position:水平方向 垂直方向；**

取值：(1)具体的方位名词：

水平方向：left right center

垂直方向：top bottom center

(2)具体的像素： px

注意点：数字可以为负数

如何让一张很大的图片无论何时都居中显示？

background-position:center top;

（5）背景图片关联方式属性：

**background-attachment：**

取值：scroll 会随着滚动条而滚动，默认值

fixed 不会随着滚动条而滚动

（6）背景尺寸属性：用于设置背景图片的大小

**background-size：**

取值：默认

具体像素:background-size：px px;

第一个值宽度，第二个值高度

百分比：background-size：% %；

宽度的等比拉伸：background-size：auto px;

高度等比拉伸: background-size：px auto;

cover:告诉系统图片需要等比拉伸到宽度和高度都填满元素

contain：告诉系统图片需要等比拉伸到宽度或高度都填满元素

（7）规定背景图片的定位区域属性：告诉系统背景图片从哪个区域开始显示

**background-origin：**

取值：padding-box 默认，从padding区域开始显示

border-box从border区域开始显示

content-box从content区域开始显示

（8）规定背景的绘制区域属性：用于指定从哪个区域开始绘制背景

**background-clip：**

取值：border-box默认，从border开始显示

padding-box 默认，从padding开始显示

content-box从content区域开始显示

（9）多重背景图片：

**background:url(“”) no-repeat left top, url(“ ”)**

**no-repeat right top, url(“”) no-repeat left bottom, url(“”) no-repeat right bottom**

注意点：多张图片之间用逗号隔开，先添加的图片会盖住后添加的图片

编写多重背景图片时应分开来写 background-image:url("images/animal1.png"),url("images/animal2.png"),url("images/animal3.png");

background-repeat: no-repeat, no-repeat, no-repeat;

background-position: left top, right top, left bottom;

**简写：快捷键是bg+**

background：背景颜色 背景图片 平铺方式 关联方式 定位方式；

注意点：background的属性任何一个都可以省略

**《5》规定边框的属性：**

**（1.1）连写格式一：同时设置四条边框的宽度、样式、颜色**

**border:边框的宽度 边框的样式 边框的颜色；**

**（1.2）连写格式二：分别设置四条边框的宽度、样式、颜色**

**border-top: 边框的宽度 边框的样式 边框的颜色；**

**border-right:边框的宽度 边框的样式 边框的颜色；**

**border- bottom:边框的宽度 边框的样式 边框的颜色；**

**border- left:边框的宽度 边框的样式 边框的颜色；**

注意点：连写格式中颜色和宽度可以省略，但是样式不可以省略

**（2.1）分开格式一：**

（1）规定边框的样式的属性：

格式：（同时设置四条边的样式）border-style:上 右 下 左 ；

取值：dashed（虚线）solid（实线）

注意点：设一个值（上下左右都是一样的）

设两个值（上下）（左右）

设三个值（上）（左右）（下）

设四个值（上）（右）（下）（左）

（2）规定边框的宽度的属性：

格式：（同时设置四条边的宽度）：border-width:上 右 下 左 ；

（3）规定边框的颜色的属性：

格式：（同时设置四条边的颜色）border-color:上 右 下 左；

**（2.2）分开格式二：**

（1）规定边框的样式的属性：

格式：（分别设置四条边的样式）

border-top-style： ；

border-light-style: ；

border-bottom-style: ；

border-left-style: ；

（2）规定边框的宽度的属性：

格式：（分别设置四条边的宽度）

border-top-width： ；

border-light-width: ；

border-bottom-width: ；

border-left-width: ；

（3）规定边框的颜色的属性：

格式：（分别设置四条边的颜色）

border-top-color： ；

border-light-color: ；

border-bottom-color: ；

border-left-color: ；

**3.** **border-radius：**设置圆角属性

四个参数:左上,右上, 右下 左下, 省略值和它的对角一样

**分开写格式：** border-top-left-radius:

border-top-right-radius:

border-bottom-left-radius:

border-bottom-right-radius:

注意点：圆角的宽度=正方形的宽度一半就是圆

圆角的宽度=长方形宽一半/高一半就是椭圆

圆角的宽度=长方形高就是半圆

圆角的宽度=正方形的边框宽度+宽度的一半就是圆环

**4.border-image：**

**分开写格式：** border-image-source: 边框的图片的路径

border-image-slice: 设置裁剪的大小 如果是设置数字,则中间内容没有，想要中间也要有背景图片,则需要在最后添加一个参数为fill

border-image-width: 图片边框的宽度

border-image-outset: 边框图像区域超出边框的量

border-image-repeat: 图像边框是否应平铺(repeat)、自适应铺满(round)或拉伸(stretch)

**《6》 规定内边距的属性：就是边框和内容之间的距离**

**注意点：给标签设置内边距后，标签的宽度和高度会发生变化**

**内边距也会有背景颜色**

（1）连写格式：padding: 上 右 下 左;

（2）分别设置格式：

padding-top: ;

padding-right: ;

padding-bottom: ;

padding-left: ;

**《7》规定外边距的属性：就是标签和标签之间的距离**

**注意点：外边距那一部分是没有背景颜色的**

（1）连写格式：margin:上 右 下 左；

（2）分别设置格式：

margin-top: ;

margin-right: ;

margin-bottom: ;

margin-left: ;

**外边距合并的现象：**

在默认布局的水平方向上，默认情况外边距是会叠加的

在默认布局的垂直方向上，默认情况外边距是会合并的，谁的外边距比较大就听谁的

注意点：

1.如果两个盒子是嵌套关系，那么设置了里面有个盒子的顶部的外边距，外面一个盒子也会被顶下来

2.如果外面的盒子想保持不变，那么需要给外面的盒子添加一个边框属性

3.在企业开发中，一般情况下如果需要控制嵌套关系盒子之间的距离，应该首先考虑padding,其次再考虑margin,因为它本质上是控制兄弟关系之间的间隙的

4.在嵌套关系的盒子中，可以利用margin: 0 auto;方式让小盒子在大盒子中水平居中，并且margin: 0 auto;只对水平方向有效

**margin: 0 auto;和text-align:center;的区别：**

margin: 0 auto;让盒子中存储的文字和图片水平居中

text-align:center;让盒子水平居中

**清除默认边距（外边距和内边距）：**

**企业开发中为了更好的控制盒子的宽高和计算盒子的宽高，在编写代码前第一件事先清楚默认边距**

\*{

margin:0;

padding:0;}

注意点：

通配符选择器会找到(遍历)当前界面中所有的标签，所以性能不好

企业开发中可以从这个网址中拷贝：http://yui.yahooapis.com/3.18.1/build/cssreset/cssreset-min.css

**《8》行高属性：**（行高可以撑起盒子的高度，有时可以省略盒子的高度）

格式： line-height: ；

要想一行文字在盒子中垂直居中那么只需要设置这行文字的行高等于盒子的高度即可

如果是多行文字那么只能通过设置padding来让文字居中

**CSS盒子模型：**

1.什么是盒子模型？

CSS盒子模型只是一种形象的比喻，HTML中所有的标签都是盒子

结论；

（1）在HTML中所有标签都可以设置宽度和高度

宽度/高度 == 存放内容的区域

内边距 ==内容与边框的距离

边框 ==

外边距 ==

（2）内容的宽度和高度：width/height

元素的宽度和高度：

宽度=左边框+左内边距+width+右内边距+右边框

高度同理可证

元素空间的宽度和高度：

宽度=左外边距+左边框+左内边距+width+右内边距+右边框+右外边距

高度同理可证

2.规律：1.增加了padding之后元素的宽高也会发生变化

2.如果增加了padding之后还想继续保持元素的宽高，那么就必须减去内容的宽高

1.增加了border之后元素的宽高也会发生变化

2.如果增加了border之后还想继续保持元素的宽高，那么就必须减去内容的宽高

**《9》盒子box-sizing属性：**

作用：保证盒子增加了padding和border后，盒子的宽度和高度不变

取值：content-box：默认值，元素的宽高=边距+内边距+内容宽高

border-box；元素的宽高==width属性

**3.选择器**

**（1）标签选择器：**

作用**：**根据指定的标签名称，在当前界面中找到所有该名称的标签，然后设置属性

格式：标签名称{

属性：值；

}

注意点：（1）标签选择器选中的是当前界面中的所有标签，而不能单独选中一个标签

（2）标签选择器无论标签藏得多深都能选中

（3）只要是HTML中是标签都可以做标签选择器

**（2）id选择器：**

作用：根据指定的id名称找到对应的标签，然后设置属性

格式：#id名称{

属性：值；

}

注意点：（1）每个HTML都有一个属性叫做id，也就是说每个标签都可以设置id

（2）在一个界面中id的名称是不可以重复的

（3）编写id选择器时一定要在id名称面前加上#

（4）id的名称是由一定规范的：只能有字母/数字/下划线组成，并且不能以数字开头，企业开发过程中一般都是以字母开头的。Id名称不能是HTML标签的名称。

（5）如果仅仅是为了设置样式，企业开发中一般不会使用id，因为在前端开发中id 是留给js使用的

**（3）类选择器：**

作用：根据指定的类名称找到对应的标签，然后设置属性

格式：.类名{

属性：值；

}

注意点：（1）每个HTML都有一个属性叫做class，也就是说每个标签都可以设置类名

（2）在一个界面中class的名称是可以重复的

（3）编写class选择器时一定要在类名称面前加上.

（4）类的名称是由一定规范的：只能有字母/数字/下划线组成，并且不能以数字开头，企业开发过程中一般都是以字母开头的，不能是HTML标签的名称

（5）类名是专门用来给某个特定的标签设置样式的

（6）在HTML中每个标签是可以同时绑定多个类名

格式：<标签名称 class=”类名1 类名2 ……”>

**id选择器与class选择器的区别：**

（1）id相当于人的身份证不可以重复，class相当于人的名称可以重复

（2）一个HTML标签名称只能绑定一个id名称，可以绑定多个class名称

（3）id选择器是以#开头，class是以.开头

（4）id一般是给js使用的，所有除非特殊情况，否则不要使用id去设置样式

**（4）后代选择器：**

作用：找到指定的标签的所有后代标签，然后设置属性

格式：标签名称1 标签名称2{

属性：值；

}

先找到名称叫做标签名称1的标签，再在这个标签下去查找所有名称叫做标签名称2的标签，然后设置属性

注意点：

（1）后代选择器必须用空格隔开

（2）后代不仅是儿子，也包括孙子等等，只要最终是放在指定标签的都是后代

（3）后代选择器不仅仅可以使用标签名称，还可以使用其他选择器

**（4）子元素选择器：**

作用：找到指定标签中所有特定的直接子元素，然后设置属性

格式：标签名称1>标签名称2{

属性：值；

}

先找到名称叫做标签名称1的标签，然后再在这个标签下去查找所有直接子元素名称叫做标签名称2的标签，然后设置属性

注意点：

（1）子元素选择器之间需要用>符号连接，并且不能有空格

（2）子元素选择器只会查找儿子，不会查找其他被嵌套的标签

（3）子元素选择器不仅仅可以使用标签名称，还可以使用其他选择器

**后代选择器与子元素选择器的区别与联系：**

（1）后代选择器用空格作为连接符号，子元素选择器用>作为连接符号

（2）后代选择器会选中指定标签中所有的特定后代标签，子元素选择器只会选中指定标签中的所有特定标签中的直接标签

（3）后代选择器和子元素选择器都可以使用标签名称/id名称/class名称作为选择器，都可以通过各自的连接符号一直延续下去

（4）在企业开发中如果想选中所有特定的标签，那么就使用后代选择器，如果只想选中指定标签中的所有特定儿子标签，那就使用子元素选择器

**（5）交集选择器：**

作用：给所有选择器选中的标签中，相交的那部分设置属性

格式：

选择器1选择器2{

属性：值；

}

注意点：（1）选择器和选择器之间没有任何的连接符号

（2）选择器可以使用标签名称/id名称/class名称

（3）企业开发中用的比较少

**（6）并集选择器：**

作用：给所有选择器选中的标签设置属性

格式：

选择器1，选择器2{

属性：值；

}

注意点：（1）选择器和选择器之间必须用，来连接

（2）选择器可以使用标签名称/id名称/class名称

**（7）兄弟选择器：**

1.相邻兄弟选择器：（CSS2）

作用：给指定选择器后面紧跟的那个选择器选中的标签设置属性

格式：

选择器1+选择器2{

属性：值；

}

注意点：（1）选择器和选择器之间必须用+来连接

（2）选择器可以使用标签名称/id名称/class名称

（3）相邻兄弟选择器只能选中紧跟其后的那个标签，不能选中隔开的标签

2.通用兄弟选择器：（CSS3）

作用：给指定选择器后面的所有选择器选中的所有标签设置属性

格式：

选择器1~选择器2{

属性：值；

}

注意点：（1）选择器和选择器之间必须用~来连接

（2）选择器可以使用标签名称/id名称/class名称

（3）相邻兄弟选择器选中的是指定选择器后面某个选择器选中的所有标签，无论有没有隔开

**（8）序选择器：**

1.同级别的第几个

（1）first-child:选中同级别的第一个标签

（2）last-child:选中同级别的最后一个标签

（3）nth-child（n）:选中同级别中的第n个标签

（4）nth-last-child（n）:选中同级别中的倒数第n个标签

（5）only-child：选中父元素中唯一的元素

（6）nth-child(odd):选中同级别中的奇数行

（7）nth-child(even):选中同级别中的偶数行

（8）nth-child(xn+y):x、y用户自定义，n是计数器，从0开始递增

注意点：不区别类型

2.同类型的第几个

（1）first-of-type:选中同级别中同类型的第一个标签

（2）first-of-type:选中同级别中同类型的最后一个标签

（3）nth-of-type（n）:选中同级别中同类型的第n个标签

（4）nth-last-of-type（n）:选中同级别中同类型的倒数第n个标签

（5）only-of-type：选中父元素中唯一类型的某个标签

**（9）属性选择器：**

作用：根据指定的属性名称找到对应标签，然后设置属性

（1）格式：[属性]

（2）格式：[属性=value]

（3）格式（属性的取值是以什么开头的）：

[属性|=value] （CSS2）

[属性^=value] （CSS3）

两者之间的区别：CSS2中的只能找到以value开头，并且value是被-和其他内容隔开的，CSS3中的只要是以value开头的都可以找到

（4）格式（属性的取值是以什么结尾的）：

[属性$=value] （CSS3）

（5）格式（属性的取值是否包含某个特定的值）：

[属性~=value] （CSS2）

[属性\*=value] （CSS3）

两者之间的区别：CSS2中的只能找到独立的单词，也就是包含value，并且是被空格隔开的，CSS3中的只要包含value就可以找到

**（10）通配符选择器 ：**

作用：给当前界面上所有的标签设置属性

格式：\*{ 属性：值；}

注意点：由于通配符选择器是给界面所有的标签设置属性，所以在设置之前会遍历所有的标签，如果标签比较多，那么性能就会比较差，所有一般在企业开发中不会使用它

**CSS三大特性**

**1.继承性：**

作用：给父元素设置一些属性，子元素也可以使用

注意点：（1）并不是所有的属性都可以继承，只有以color/font-/text-/line开头的属性可以继承。

（2）不仅是儿子可以继承，后代都可以继承。

（3）CSS继承性的特殊性：a标签的文字属性和下划线是不能继承的，h标签的文字大小是不能继承的

应用场景：在企业开发中，一般用于设置网页的一些共性信息，例如网页的文字颜色，字体，大小等

**2.层叠性：**

作用：层叠性是CSS处理冲突的一种能力

注意点：层叠性只有在多个选择器选中“同一个标签”，又设置的 “相同属性”，才会发生层叠性

**3.优先级：**

作用：当多个选择器选中同一个标签，并且给同一个标签设置相同的属性时，如何层叠就由优先级来确定

优先级的判断方式：（1）是否是直接选中（间接选中就是指继承）：

如果是间接选中那么就是谁离目标标签比较近就听谁的

（2）相同选中器：

如果都是直接选中并且是相同选中器那么谁写在后面就听谁的

（3）不同选择器：

如果都是直接选中并且不是相同选中器，那么就按照选择器的优先级来层叠：id>类>标签>通配符>继承>浏览器默认

**《10》!important属性：**

作用：用于提升某个直接选中标签的选择器中的某个属性的优先级，可以将被指定的属性的优先级被提升为最高

注意点：（1）只能用于直接选中，不能用于间接选中

（2）只能提升被指定的属性的优先级，其他属性的优先级不变

（3）通配符选择器选中的标签也是直接选中

**权重问题：**

作用：当多个选择器混合在一起使用时，我们可以通过计算权重来判断谁的优先级最高

权重的计算规则：（1）首先先计算选择器中有多少个id，id多的选择器优先级最高

（2）如果id的个数一样，那么再看类名的个数，类名个数多的优先级最高

（3）如果类名的个数一样，那么再看标签名称的个数，标签个数多的优先级最高

（4）如果id个数一样，类名个数一样，标签名称个数一样，那么此时谁写在后面就听谁的，即优先级一样的时候，谁写在后面就听谁的

注意点：只有选择器直接选中标签才需要计算权重

**CSS元素显示模式**

**1.div和span**

**1.div**

作用:一般配合CSS完成网页的基本布局

**2.span**

作用：一般配合CSS修改网页中的一些局部信息

两者区别：（1）div会单独占领一行，而span不会单独占领一行

（2）div是一个容器级的标签，而span是一个文本级的标签

容器级标签和文本级标签的区别：

容器级的标签中科院嵌套其他所有的吧，而文本级的标签只能嵌套文字/图片/超链接标签

容器级的标签：div h u lol li dt dd……

文本级的标签：span p buis e mind del……

2.块级元素、行内元素、行内块级元素

（1）块级元素：独占一行；如果没有设置宽度，那么默认和父元素一样宽；如果设置了宽度，就按照设置的宽度来显示

p div h ul oi li dt dd……

（2）行内元素：不会独占一行；如果没有设置宽度，那么默认和内容一样宽；行内元素是不可以设置宽度的

span buis strong em ins del……

（3）行内块级元素：为了能够让元素既能够不独占一行，又可以设置宽度和高度，那么就出现了块级行内元素

如何转换CSS元素的显示模式？

设置元素的display属性

取值：block 将元素转换为块级元素

inline 将元素转换为行内元素

inline-block 将元素转换为行内块级元素

**网页布局方式 （水平排版用浮动）**

1.网页的布局方式其实就是指浏览器是如何对网页的元素进行排版的

（1）标准流（文档流/普通流）排版方式

浏览器的默认排版方式，有两种排版方式：

垂直排版：如果元素是块级元素，那么就会垂直排版

水平排版：如果元素是行内元素/行内块级元素，那么就会水平排版

（2）浮动流排版方式

【1】浮动流是一种半脱离标准流的排版方式，只有水平排版方式，它只能设置某个元素向左对齐或向右对齐

注意点：（1）浮动流中是没有居中对齐的

（2）浮动流中是不可以使用margin:0 auto;

特点： （1）在浮动流中是不区分行内元素/块级元素/行内块级元素，无论是内元素/块级元素/行内块级元素都可以水平排版，都可以设置宽高

（2）浮动流中行内元素/块级元素/行内块级元素都可以设置宽高

【2】浮动元素脱标：

当某一个元素浮动之后，那么这个元素看上去就像被从标准流中删除一样

脱标后的影响：如果前面一个元素浮动了，而后面一个元素没有浮动，那么这个时候前一个元素就会盖住后面一个元素

【3】浮动元素排序规则：

（1）相同方向上的浮动元素，先浮动的元素会显示在前面，后浮动的元素会显示在后面

（2）不同方向上的浮动元素，左浮动找左浮动，有浮动找右浮动

（3）浮动元素浮动后的位置由浮动元素浮动之前在标准流中的位置来决定

【4】浮动元素贴靠方式：

（1） 当父元素足够放下所有浮动元素的时候，那么浮动元素就会并列显示

（2）当父元素不够放下所有浮动元素的时候，那么就会从最后一个元素开始向前一个元素贴靠，一直往前贴，直到贴到父元素的边框，这是不管父元素够不够都会贴靠在那里

【5】浮动元素字围现象：

【6】浮动元素高度问题：

（1）在标准流中内容的高度是可以撑起父元素的高度

（2）在浮动流中浮动元素是不可以撑起父元素的高度

【7】清除浮动：

《1》方式一：给前面一个元素设置高度

注意点：在企业开发中，我们能不写高度就不写高度，所以这种方式用的比较少

《2》方式二：给后面一个元素添加clear属性

取值：none:默认取值，按浮动元素的规则来排序(右浮动找右浮动,左浮动找左浮动)

left:不要找前面的左浮动

right:不要找前面的右浮动

both: 不要找前面的左浮动和右浮动

注意点：当给某个元素添加clear属性之后，那么这个属性的margin属性就会失效

《3》方式三：隔墙法

（1）外墙法：在两个盒子之间添加一个额外的块级元素，并且给这个块级元素添加clear:both;属性

注意点：外墙法可以让第二个盒子使用margin-top属性，但不可以让第一个盒子使用margin-bottom属性。一般在企业开发中，是直接给额外添加的块级元素添加一个高度

（2）内墙法：在第一个盒子中所有子元素最后添加一个额外的块级元素，并且给这个块级元素添加clear:both;属性

注意点：内墙法可以让第二个盒子使用margin-top属性，也可以让第一个盒子使用margin-bottom属性。

（3）区别：外墙法不能撑起第一个盒子的高度，而内墙法可以撑起第一个盒子的高度

《4》方式四：伪元素选择器：

作用：给指定标签的内容前面添加一个子元素或者给指定标签的内容后面添加一个子元素

（1）格式一:给指定标签的内容前面添加一个子元素

标签名称：：before{

属性名称：值；

content: ；

给指定添加的子元素中存储的内容

width: ;

height: ;

给指定添加的子元素的宽度和高度

display:block;

给指定添加的子元素的显示模式

visibility:hidden;

隐藏添加的子元素}

（2）格式二：给指定标签的内容后面添加一个子元素

标签名称::after{

属性名称：值；

content: ；

设置添加的子元素的内容为空

height:0 ;

设置添加的子元素的高度为0

display:block;

设置添加的子元素为块级元素

visibility:hidden;

隐藏添加的子元素

clear:both;

}

兼容IE6：给第一个盒子添加

.box1{\*zoom:1;}

《6》 方式六：overflow:hidden;

作用：（1）可以将超出标签范围的内容裁剪掉

（2）可以清处浮动

（3）可以通过overflow:hidden;让里面的盒子设置margin-top之后，外面的盒子不被顶下来

注意点：可以让第二个盒子使用margin-top属性，也可以让第一个盒子使用margin-bottom属性

（3）定位流排版方式

《1》相对定位:相对于自己以前在标准流中的位置来移动

格式：position:relative

注意点：

不会脱离标准流的，会继续在标准流占用一定的空间

需要配合 top right left bottom 来使用，但同一方向定位属性只能使用一个

由于相对定位是不脱离标准流的，所以是区分块级元素/行内元素/行内块级元素

由于相对定位是不脱离标准流的，并且相对定位的元素会占用标准流中的位置，所以当给相对定位元素设置margin/padding等属性会影响到标准流的布局

应用场景：用于对元素进行微调（想让多个盒子对齐时可以使用【想让多个盒子顶部对齐也可以使用浮动定位】}）

配合绝对定位来使用

《2》绝对定位：相对于body来定位

格式：position：absolate

定位的参考点规律：

默认情况下所有的绝对定位元素是相对于body来定位

如果绝对定位的元素有祖先元素，并且祖先元素也是定位流，那么这个绝对定位元素就会以定位流（相对定位/绝对定位/固定定位）的那个祖先元素作为参考点

如果绝对定位的元素有多个定位流的祖先元素，那么这个绝对定位流元素就会以离它最近的那个祖先元素作为参考点

注意点：绝对定位是脱离标准流的，不区分块级元素/行内元素/行内块级元素

需要配合 top right left bottom 来使用，但同一方向定位属性只能使用一个

如果一个绝对定位的元素是以body作为参考点，那么其实是以网页首屏的宽度和高度作为参考点，而不是整个网页的宽度和高度作为参考点的

一个绝对定位元素是会忽略祖先元素的padding

应用场景：用于对元素的微调

配合相对定位来使用

**子绝父相（子元素用绝对定位，父元素用相对定位）**

相对定位的弊端：不脱离标准流，会继续占用一定的空间，不利于布局界面

绝对定位的弊端：默认情况下绝对定位是以body作为参考点，不会随着浏览器的宽度高度的变化而变化

应用场景：当某一个元素覆盖在另一个元素上面时，需要用到定位流

**绝对定位的水平居中问题：**

只需设置绝对定位元素的left:50%;

然后在设置绝对定位元素的margin-left:-元素宽度的一半px

《3》固定定位：固定定位和背景关联方式很像，背景定位可以让背景图片不随着滚动条而滚动，而固定定位可以让某个元素不随着滚动条而滚动

格式：position:fixed;

注意点：固定定位的元素是脱离标准流的，不会占用标准流的空间

固定定位和绝对定位一样，不区分块级元素/行内元素/行内块级元素

应用场景：导航条在网页顶部的固定位置

网页两侧的广告

底部返回顶部的按钮

《4》静态定位（默认状态）

格式：position:static

**定位流z-index:专门用于控制定位流的覆盖关系**

规律：默认情况下所有的元素都有一个z-index属性，取值为0

默认情况下定位流元素覆盖标准流的元素

默认情况下定位流的元素后遍写的覆盖先编写的，如果定位流元素中设置了z-index属性，那么谁的z-index比较大，谁就显示在上面

注意点：从父现象

（1）如果两个元素父元素都没有设置z-index属性，那么谁的z-index比较大，谁就显示在上面

（2）如果两个元素父元素都设置了z-index属性，那么子元素的z-index就会失效，谁的父元素z-index属性比较大，那么谁就会显示在上面

**a标签伪类选择器：**

（1）作用：专门用来修改a标签不同状态的样式

（2）状态：从未被访问过（默认状态）、被访问过状态、鼠标长按状态、鼠标悬停在上面状态

（3）格式：

a:link{color： ;}修改从未被访问过状态下的样式

a:visited{color: ;}修改被访问过状态下的样式

a:active{color: ;}修改鼠标长按状态下的样式

a:hover{color: ;}修改鼠标悬停a标签上状态下的样式

（4）注意点：

a标签伪类选择器可以单独初中，也可以同时出现。同时出现需要严格准守爱恨原则（love hatd）：a:link、a:visited、 a:hover、a:active

如果默认状态和被访问过状态是一样的，可以简写a{color:}

在企业开发中编写a标签伪类选择器最好写在标签选择器后面

在企业开发中和a标签盒子相关属性都写在标签选择器中（显示模式/宽度/高度/padding/margin）

在企业开发中和a标签文字/背景相关属性都写在伪类选择器中

（5）悬停在链接上背景颜色的改变：

http://localhost:63342/HTML5学习代码/A标签伪类选择器.html

**过渡模块**

1.格式：选择器{

transition-property:属性；

transition-duration:时间

}

标签:hover{

属性：值；

}

2.过渡三要素

：hover可以使用在任何标签上

transition-property: 告诉系统哪个属性需要执行过渡效果

transition-duration:告诉系统过渡效果持续的时长

注意点：当多个属性需要同时执行过渡效果，用逗号隔开即可

3.其他属性；

transition-delay:告诉系统延迟多久后再执行过渡动画

transition-timing-function:告诉系统过渡动画的运动速度

取值：linear匀速

ease逐渐慢下来

ease-in加速

ease-out减速

ease-in-out先加速再减速

4.连写：transition：property duration timing-function delay；

注意点：如果多个属性需要同时执行过渡效果，用逗号隔开即可

连写的时候可以省略后面两个参数，因为只有编写了前面两个参数就已经满足了过渡三要素

如果多个属性运动的速度/延迟的时间/持续的时间都一样，那么可以简写为transition：all 值；

5.编写过渡套路：

先编写基本界面，再修改我们认为需要修改的属性，最后给被修改属性的那个元素添加过渡即可

6.弹性效果

http://localhost:63342 /HTML5学习代码/弹性效果.html

7.手风琴效果

http://localhost:63342/ HTML5学习代码/手风琴效果.html

**2D转换模式**

1.格式：

选择器：hover{transform: ；}

取值：rotate(数字deg) 旋转多少度

translate（ px, px） 平移

第一个值代表水平方向，第二个值代表垂直方向

scale（ , ） 缩放

第一个值代表水平方向，第二个值代表垂直方向，取值1代表不变，大于1 代表放大，小鱼1代表缩小。如果水平垂直方向缩放都一样，可以简写为一个参数

注意点：如果父元素被缩放了，那么子元素也会被缩放

2.综合效果：

transform: rotate(数字deg) translate（ px） scale（ , ）；

注意点：需要用空格隔开。2D的转换会修改元素坐标系，所以旋转后再平移就不是水平平移了

3.形变中心点：

格式：

选择器{：transform-origin： px px；}

第一个参数代表水平方向，第二个参数代表垂直方向

取值：像素、百分比、单词

默认情况下所有的元素都是以自己的中心点作为参考来旋转

4.旋转轴点：

（1）默认情况下所有的元素都是围绕Z轴进行旋转：

transform: rotateZ(数字deg)

（2）围绕X轴进行旋转：transform: rotateX(数字deg)

（3）围绕Y轴进行旋转：transform: rotateY(数字deg)

注意点：屏幕的原点在左上角 左上角transform: rotate(0% 0%)

中心点transform: rotate(50% 50%)

5.透视属性：近大远小

perspective: px;

注意点：想看到近大远小的效果越明显那么像素值就设置越大，想看到近大远小的效果越模糊那么就像素设置越大

必须添加到需要呈现近大远小效果的元素的父元素上面

**盒子阴影和文字阴影**

1.盒子阴影：box-shadow:h-shadow v-shadow blur spread color inset;

h-shadow水平偏移

v-shadow垂直偏移

blur模糊度

spread阴影扩展

color阴影颜色

inset内外阴影

注意点：盒子阴影分内外阴影，默认是外阴影

快速添加盒子阴影：box-shadow:h-shadow v-shadow blur，阴影颜色由盒子内容颜色决定

2.文字阴影：text-shadow:h-shadow v-shadow blur color

注意点：快速添加文字阴影test-shadow:h-shadow v-shadow blur，阴影颜色由文字颜色决定

3.翻转菜单

http://localhost:63342/HTML5学习代码/翻转菜单.html

**动画模块**

1.动画与过渡的异同：

不同点：过渡必须人为的触发才会执行动画

动画不需要人为的触发就会执行动画

相同点：过渡与动画都是用来给元素添加动画的，都是系统新增的一些属性，都需要满足三要素才会有动画效果

2.动画三要素：

（1）告诉系统需要执行哪个动画

（2）告诉系统我们需要自己创建一个名称叫做xxx的动画

（3）告诉系统动画持续的时长

3.格式一：（起始动画）

选择器{

animation-name: ；

animation-duration: ；

}

@keyframes 动画名称{

form{

}

to{

}

}

格式二：（百分比）

选择器{

animation-name: ；

animation-duration: ；

}

@keyframes 动画名称{

0%{

}

25%{

}

50%{

}

75%{

}

100%{

}

}

4.其他属性：

<1>animation-delay:告诉系统延迟多久后再执行动画

<2>animation-timing-function:告诉系统动画的执行的速度

取值：linear匀速

ease逐渐慢下来

ease-in加速

ease-out减速

ease-in-out先加速再减速

<3>animation-iteration-count:告诉系统动画被播放的次数

<4>animation-direction:告诉系统是否需要执行往返动画

取值：normal:默认，执行完一次之后回到起点再执行下一次

alternate：执行完一次之后往回执行下一次

<5>animation-play-state: : 告诉系统当前动画是否需要暂停

取值:running: 执行动画

paused: 暂停动画

<6>动画的状态：等待状态 、开始状态 、结束状态

animation-fill-mode:指定动画的等待状态和结束状态的样式

取值：none:不做任何改变

forwords：让元素结束状态保持最后一帧的样式

backwordes：让元素等待状态时候显示动画的第一帧样式

both：让元素等待状态时候显示动画的第一帧样式，让元素结束状态保持最后一帧的样式

5.注意点：（1）动画中如果有和默认样式中同名的属性，会覆盖覆盖默认样式中的同名属性

（2）在编写动画的时，固定不变的值写在前面，需要变化的值写在后面

6.连写格式：

animation：动画名称 动画时长 动画运动速度 延迟时间 执行次数 往返动画；

简写模式：animation：动画名称 动画时长 动画运动速度；

7.云层效果

http://localhost:63342/HTML5学习代码/云层效果.html

8.无限循环的根本是末尾额外添加两张与第一张第二张的图片

<http://localhost:63342/HTML5学习代码/无限滚动.html>

**3D转换模块**

1.2D和3D：

2D就是一个平面，只有看到和高度，没有厚度

3D就是一个立体，有宽度和高度，还有厚度

默认情况下，所有元素都呈现2D展现的

2.如何让元素呈3D展现？

和透视效果一样，想看到某个元素的3D效果，只需要给它的父元素添加一个transform-style属性，然后设置perspective-3d即可

3.正方体

编写顺序：（先旋转再平移）上 后 下 前 左 右

4.长方体：先编写正方体然后再对上后下前四个面进行拉伸

**Vertical-align:**

设置图片与盒子底部对齐 ：vertical-align: bottom;

设置图片与盒子顶部对齐： vertical-align: top;

设置图片与文字基线对齐：vertical-align: baseline;

设置图片顶部与文字顶部对齐：vertical-align: text-top;

设置图片底部与文字底部对齐：vertical-align: text-bottom;

设置图片与文字中线对齐： vertical-align: middle;

**渐变：**

1.线性渐变：

直接设置渐变：background: linear-gradient(red,blue);

添加渐变方向：background: linear-gradient(to right, red,blue);

可以根据角度设置渐变方向：background: linear-gradient(45deg,red,blue);

设置渐变到下一颜色的位置：background: linear-gradient(to right,red 30%,green 80%,blue);

2.径向渐变：

默认从中点开始扩散：background: radial-gradient(red,green,blue);

大小范围,位置,颜色：background: radial-gradient(100px at center center,red,blue);

可以通过像素进行设置位置：background: radial-gradient(100px at 50px 50px,red,blue);

3.重复渐变：

重复线性渐变：background: repeating-linear-gradient(to right,red 10%,blue 20%);

重复径向渐变：background: repeating-radial-gradient(red 10%, blue 20%, yellow

公共的项目类写在第一个

小范围调整时，使用相对定位；大范围调整使用绝对定位

谷歌浏览最小字体是12px，如果想字体再小一些，可以使用缩放属性（inline-block）

在一个盒子中，如果一个元素添加了浮动，那么其他元素也要进行浮动，为了页面的稳定性

.webpictuer网页图片，谷歌独有图片，图片被压缩处理过，但图片清晰度不会改变，加载速度快

1.如果图片的宽度小于父元素的宽度, 那么可以使用text-align: center;来居中图片  
2.如果图片的宽度大于父元素的宽度, 但是图片的宽度是已知的, 那么可以使用定位流 left:50%; margin-left: -图片宽度的一半;  
3.如果图片的宽度大于父元素的宽度, 但是不知道图片的宽度, 那么可以使用给父元素添加text-align: center; 给图片添加margin: 0 -100%;

除了C语言，其他语言都是面向对象的，C语言是面向过程

单引号包括双引号，双引号包括单引号

https加密后的http

需求驱动开发

**JS基础**

**一、JS的认识**

（1）js的作用：html是从语法意义上来构造网页；css是从视觉上美化界面；js是从交互上提升用户体验

（2）js是出现：1995年出现，网景公司设计

（3）js的发展：原本叫做livescript，后来改名为javascript

**二、注释的认识**

（1）注释作用：用来说明代码，提升代码的可读性

（2）注释种类：单行注释（一般写在代码的上面或右面）：使用// 快捷键：ctrl + /

多行注释（一般写在代码的上面）：/\* 快捷键：ctrl + shift + /

\*

\*

\*/

**三、变量**

（1）定义：用来存储数字、字符以及可以表示这些东西的东西

（2）变量的使用：1.声明变量 2.赋值变量 3.使用变量

Eg: var: a;

a = 10;

alter(a);

（3）特点：声明变量和赋值变量可以写在一起，叫做变量的定义

eg：var a = 10;

通过控制台打印输出

eg；console.log（a）;

多个变量定义可以写在一行

eg: var a = 10,b = 20, c = 30;

（4）变量的命名规则：

从语法意义上的规范：只能以数字、字符、下划线以及$符号组成，不能以数字开头，不能和保留字重名

从形式上的规范：以它实际意思的英文命名，而且采用驼峰命名法。如果是由多个单词组成，那么从第二个单词开始的每个单词的第一个字母要大写

eg:lastName

（5）变量的种类：

数值

字符 ：但凡用单引号引起来的都是字符

布尔 ：就是判断值，只有两个值，一个ture,一个是false

对象

未定义:凡是系统无法识别的类型就是未定义类型

检查变量的种类：console.log(typeof 变量名)

变量如果未赋值会输出为未定义类型

（6）变量的赋值：用来赋值的值不会发生变化，被赋值的值会发生变化

（7）其他类型与布尔类型的转换：

对于数值类型，只要不是0，都是真；

对于字符类型，只要不是空，都是真；

对于对象类型，只要不是null，都是真

对于未定义类型，都是假

（8）布尔转化成数值类型：true是1；false是0

（9）变量根据位置的类型：全局变量 局部变量，一般在函数体中的变量就是局部变量，在函数外面的变量就是全局变量

《1》变量的作用域：局部变量只能在函数内部起作用，出了函数没有作用

《2》变量访问的原则：就近原则，就是访问在函数中定义了这个变量，就访问函数内部的变量，如果没有，就去访问全局变量，如果还没有就返回未定义变量（错误）

《3》变量的声明的提升：如果在函数中访问一个在访问后定义的变量，那么就会把这个变量的声明提升到函数的最前面。注意：只是提升声明

**四、运算符**

（1）定义：用来连接变量的符号就得运算符

（2）种类：算术运算符 关系运算符 逻辑运算符

《1》算术运算符：+ - \* / %

特点：\* / %的优先级高于+ -

使用小括号可以提升优先级

+=和-=表示赋值的时候，如果用自己赋值给自己就可以使用

++以及—的认识：一般++和—放在变量中使用，如果对于++，放在变量后面，表示先使用变量再让变量的值加一，赋给原来的变量；如果放在变量的前面，表示先让变量的值加一，然后赋给原来的变量

《2》关系运算符：> < >= <= == === != !==

特点：>=以及<=只有满足一个条件就可以

==只需要比较内容就可以

===既要比较类型，又要比较内容，两个都符合才可以

!=和!==是== ===的相反

关系运算符的结果都是布尔值

关系运算符不可以连续

《3》逻辑运算符：&& || !

特点：关系运算符进行运算符的都是布尔值

&&只有两边都是真的时候结果才是真

||只要有一个是真的返回结果就是真

！表示取反

**五、逻辑运算的特点**

（1）对于&&运算符，如果第一个表达式是真的，那么第二个表达式还需要计算；如果第一个表达式是假的，那么第二个表达式就不需要计算

（2）对于||运算符，如果第一个表达式是真的，那么第二个表达式就不需要计算；如果第一个表达式是假的，那么第二个表达式还需要计算

（3）运算符的优先级：

<1>算数运算符高于关系运算符，关系运算符高于逻辑运算符

<2>一般单目运算符高于双目运算符，逻辑！高于逻辑&&高于逻辑||

<3>赋值运算符的等级最低

**六、if语句的认识**

（1）基本if语句形式：

if（判断条件）{如果条件成立执行代码}else{如果条件不成立执行代码}

（2）多重If语句的形式：

if（判断条件1）{如果条件成立执行代码}else if（判断条件2）{如果条件2不成立执行代码}else{如果条件都不成立执行代码}

（3）嵌套if语句形式：if(if(判断条件){如果条件成立执行代码}else{如果条件不成立执行代码})else{if(判断条件){如果条件成立执行代码}else{如果条件不成立执行代码} }

（4）多重if语句的跳楼（阻断）现象：就是对于多重if语句，如果第一条件成立了，后面的所有条件都不再判断

（5）阻止跳楼（阻断）现象：就是写代码时候，把else去掉就可以了

**七、循环语句的认识**

（1）for循环：for（初始化条件；终止条件；变化值）{ }

（2）for循环本质：根据初始化条件，然后和终止条件进行判断，如果成立，就执行括号中的代码，然后让值进行变化，然后继续和终止条件进行判断，如果成立继续执行代码，直到不符合条件为止

（3）死循环：就是让终止条件无法终止，我们可以直接设置为true，一般不能设置死循环

（4）while循环：whlie(判断条件){如果判断条件成立就执行代码}

（5）whlie和for的区别：一般如果固定次数的我们就使用for循环

**八、函数的认识**

（1）函数的定义：函数可以通过 function 函数名称（ ）{ } 定义或者采用 var sum=function（）{ }来定义，一般都是采用第一种方式

（2）函数的参数：函数参数分为实际参数和形式参数两种，在函数定义时传入小括号中的参数是形式参数，在函数调用的时候传入的参数是实际参数，形式参数可以是符合变量命名规范的任何值

（3）函数的返回值：通过return来进行返回

注意点：函数中使用return后会结束函数，后面的代码不会执行了

（4）函数参数传递的特点：函数中实际参数可以和形式参数的个数不一样，编译器会根据具体的情况返回对应的值，获取函数的实际参数的个数可以通过argument.length，获取函数形式参数的个数可以通过 函数名称.length 获取

**九、常见数学公式的使用：**

（1）取出一个数值的正的平方根：Math.sqart( );

（2）设置一个数的几次方：Math.pow( , );

（3） 输入框的认识和使用：prompt( ),返回值都是字符

（4）转化整数的公式：parselnt() ，可以直接把数值字符转化为整数，可以截断字符，截取出数值，但是必须第一个字母是数字

（5）四舍五入函数：Math.round( );

**十、js的具体作用**

（1）js可以操作样式（css）：直接修改某一个盒子的高度，获取标签的方法是getelementByID();

（2）js可以操作结构（html）:设置一个双标签的内容是.innerHTML，包含了标签的中间的所有的东西

（3）js放置位置：可以放置行内样式，可以放置页内样式，可以放置外联样式，一般我们使用页内样式和外联样式

**十一、事件的基本认识**

（1）事件的三要素：

事件源：就是谁被触发了

事件属性：就是你做了什么导致了结果

事件指令：就是事件触发的结果或者影响或者事件触发后要去做的事情，形式是 function () { }

（2）事件三要素的组合：事件源.事件属性 = 事件指令

**补充：** 《1》不能通过设置行高来让图片垂直居中

《2》display和visiblity都可以隐藏标签，但是display不会占用位置，visiblity会占用位置

《3》NaN表示一个特殊的数值字符，自己和自己也是不相等的。isNaN，如果结果是true,表示不是一个数字，而且可以判断数值和字符

《4》switch语句的应用:使用selet标签显示不同的语句1.当select发生改变的时候,我们直接使用onchange来监听发生了改变;2.当需要获取选中的值得时候,使用select标签的value属性来获取，每个option中不要设置value值

《5》代码抽取的原则:1.相同的东西放在一起 2.不同的东西作为参数然后形成一个函数就可以

《6》transform属性的使用以及应用:1. Transform使用的时候是相对于第一次的位置为基准值 2. Transform旋转后会改变坐标系3.不同的浏览器的属性值是不一样的.4.transform如果想要设置同时改变，不要写在一起，不能独立写，否则会覆盖

《7》window.onlaod的认识:,window.onload表示当窗口加载完毕了,这个时候我们在js代码中获取标签才可以获取到,所以如果我们把js写在js标签的上面我们就需要先当窗口加载完毕后来获取对应的标签

**十二、数组的认识**

（1）数组的定义：数组可以用来存贮大量的数据，而且可以存贮不同类型的数据

（2）数组的特点：数组是有序的，数组中的元素可以通过序号来访问，形式：

数组名称[序号]；

（3）数组的长度：数组名称.length

（4）数组的遍历可以通过for循环来实现

**(5)数组的基本方法:**

-push(),在数组的后面添加一个元素,返回值是数组的长度

-pop(),在数组的后面删除一个元素,返回值是删除的那个元素

-shift(),在数组的前面删除一个元素,返回值是删除的那个元素

-unshift(),在数组的前面添加一个元素,返回的是数组的长度

-contact(),连接连个元素,原来的数组不会发生变化

-sort(),数组转化成字符后排序,但是如果要按数值排序,需要使用匿名函数参数,函数参数会自动执行

-join(),数组转化字符,可以使用传入的参数来连接每一个数组中的元素，形成一个字符串

-split(),字符转化数组,将有规律的字符转化成数组,使用参数中传入的值进行分割

**补充：**三目运算符：形式： 判断条件？代码1：代码2；如果判断条件成立就执行代码1，如果判断条件不成立就执行代码2。主要用于全选以及反选中

排他思想：所谓排他就是让其他的所有的设置一种样式，但是自己设置自己独立的属性，一般我们实现需要两步：1.设置所有的为其他样式 2.设置自己为自己的样式

排他思想的应用：给不同的行设置不同的颜色，通过排他思想或者monse方法的对立性（mouse有两个对立的方法）

this的认识:一般放在事件指令中，表示谁被触发了，一般是事件源

变量和属性以及方法和函数的区别：属性需要隶属某一个对象，变量不需要，函数属于某一个对象就是方法

属性的扩展:可以给某一个对象扩展属性,用来记录可能销毁的值

**十二、字符串的常用方法**

（1）转化成字符的方法：1.使用连接符号+ 2.使用string方法

（2）获取某一位置的字符以及unicode编码的认识：charCodeAt();

(3)获取某一个位置的字符：charAt();

(4)获取一个字符的位置的方法：indexof(),如果从后往前去：lastindexof(),注意排序的位置还是从左往右开始的

（5）获取一个字符的真正的占字节的方法的获取：一般如果是汉字unicode编码就是大于127，在0和127内是普通的字母数字

（6）转化大小写：转化大写：toUpperCase 转化小写：toLowerCase()

(7)截取字符的方法：

slice（），如果传入一个参数表示从这个参数开始截取后面所有的字符；如果传入两个参数，表示截取从第一个参数的位置到第二个参数的位置的所有字符，但不包括第二个参数所在位置的字符

subStr(),如果传入一个参数和slic()一样，如果传入两个参数表示，第二个参数表示从第一个参数开始截取字符的长度

**十三、定时器**

**（1）**定义：可以用来做到每隔一段时间自动执行某个操作的东西

（2）种类：一次定时器（setTimeout）和多次定时器（setInerval）

(3)定时器的清除：clearTimeout()清除一次定时器 clearInerval()清除多次定时器

（4）定时器的本质：多次定时器：执行的时候会把任务放在一个队列中，如果每一个任务是俩秒种，每间隔1秒钟。多次定时器执行时间就是两秒钟，具体是每隔一秒会把任务放在队列中，当上一个任务执行完成后，下一个任务立即执行；

如果是一次定时器，就是在一秒后执行执行两秒的任务，所以是三秒