前端：为浏览器内核webview写代码实现UI、实现交互

出于安全考虑，现在浏览器禁止直接访问磁盘文件

HTML/CSS

* + 浏览器
    - Chrome
    - FireFox
    - IE
    - Edge(Win10)
    - Safari(Mac)
    - Opera
  + 浏览器内核：渲染引擎（layout Engine + rendering Engine）+JS引擎
    - Trident(IE)
    - EdgeHTML(Edge)
    - webkit(Safari)
    - Chromium/Blink(Chrome，比webkit渲染更快)
    - Gecko(FireFox)
    - Presto (Opera 已经废弃 )
* HTML:超文本标记语言
* 标记：Markup,能被浏览器识别,无法直接显示,用于框定HTML文件结构
* HTML页面骨架：所有标记小写，html,body,title,head
* 标记的分类
* 注释 :<!--x-->或者Ctrl + /
* 单标签 :只有一个开始标签，结尾可以什么都没有也可以有一个/
* 双标签 :有一个开始标签和一个结束标签
* 标记的关系
* 父子关系
* 兄弟关系
* DOCTYPE:指明了HTML的标准,如：HTML5(<!DOCTYPE html> )
* HTML4.01
* HTML5
* XHTML
* 常用标记

表格标记：

table

<caption>表格标题</caption>

<thead></thead>

<tbody></tbody>

tr>td/th tr:行 td:列 th:小标题

建立4行3列的表格：(tr>td\*3)\*4

表格属性：border(设定表格边框)：外部是table边框，内部是cell边框

cellspacing: table与cell之间的间隙（缺省为2px）

cellpadding：cell与内容之间的距离（缺省为1px）

width, height:表格的宽高度

align:表格水平对齐方式

表格的合并：

1. 横行合并
2. 找到左上cell的行、列
3. 几个单元格合并
4. 左上cell 的td加属性colspan=”合并数”
5. 删除已经占位的cell对应的td
6. 纵向合并
7. 找到左上cell的行、列
8. 几个单元格合并

（3）左上cell 的td加属性rowspan=”合并数”

（4）删除已经占位的cell对应的td

3、多行多列合并

1. 找到左上cell的行、列
2. 几个行、列单元格合并

（3）左上cell 的td加属性rowspan=”列合并数” colspan=”行合并数”

（4）删除已经占位的cell对应的td

标题标记：hx 即h1~h6 是语义化的

段落标记：p

链接标记：a

href=”” 跳转地址 #表示空链接

target=”” 缺省为self(本页面打开) \_blank(新页面打开)

锚点定位 <a href=”#xxx”>zzz</a>

<p id=”xxx”>zzzzz</p>

水平线标记：hr

换行标记：br

布局标记：div （独立一行） span（不独立一行） 无语义

文本格式化标记：

b—strong（加粗） i—em（倾斜） s—del（删除线） u—ins（下划线）

b i s u:是HTML4标准中，不带有语义

strong em del ins:是XHTML标准中，兼容HTML5，带有语义，表示强调

图片标记 img

1. alt=” ” 用于图片无法加载时显示的文字
2. title=” ” 用于鼠标悬停时显示的文字
3. 在img里面设置宽度或高度中的一个，图片等比例缩放
4. border 边框宽度
5. src 图片地址

特殊字符：空格：&nbsp 小于：&lt 大于：&gt &:&amp

路径

绝对路径：域名+路径+参数

相对路径：路径+参数 不同文件夹下 . 表当前目录 .. 表上级目录

列表标记

无序列表：ul>li ul中只能有li 可以做菜单 如果非要加其他的可以放到li里面

(去除原点：list-style：none)

有序列表：ol>li

自定义列表：dl>dt dd dt:小标题

Input标记

html 改变input触发边框颜色

//去掉input点击后边框颜色

input { outline:none; }

//改变input触发后边框颜色

input{ outline-color:red; }

placeholder：隐式的显示文字，当输入内容后将消失。

type

text password radio checkbox button reset submit

radio:name相同只能选择其中一个

name、id:标识控件，id唯一

maxlength:可输入最大字符数

label标记

将文字与控件关联起来

（1）<label>文字+控件</label>

（2）<label for=”控件id”>文字</label>控件

textarea标记:用于输入多行文本，但不支持富文本

select标记:下拉选项option 默认显示在option中加selected

form

action:如果点击了submit按钮,浏览器会向服务器发送请求,action就是请求的页面,如果action为空就还是请求刚刚的页面

method:get/post

get:把form中控件的参数用?<name >=<value>&<name >=<value>

post: 把form中控件的参数放到form data中

CSS：层叠样式表（父代的样式若子代没有样式,则子代延续父代的样式,否则用自己的样式）

引入方式:

行内样式表：style=”属性名：属性值；属性名：属性值”

内部样式表：

选择器：

标签选择器：标签名{}

类选择器：.类名{} 用于描述标签的特征

（1）一个标签使用多个类： <div class=”类1 类2”></div>类之间用空格隔开

冲突：同一个样式在不同的类里定义成不同的样式，定义在后面的样式可以覆盖定义在先的样式，在先在后指的是类的定义，而不是标签中类的使用

（2）一个类可以用于多个标签

Id选择器：一个标签只能有一个id,同一个id值只能有一个标签使用 #id{}

应用于页面的顶级结构

通配符选择器：\*：代表所有标签

后代选择器：E F { }访问到E下的所有的F元素（无论嵌套多少层）

子代选择器：E>F { }访问到E下的所有的F元素（只访问下一级，只有一层）

兄弟选择符：E~F选择E元素所有兄弟元素F。

相邻选择符：E+F选择紧贴在元素之后F元素。

交集选择器：直接把选择器拼接在一起（div.red{}）

并集选择器：用逗号分隔选择器（p,div）

属性选择器

E[att]：选择具有att属性的E元素。

E[a。tt=”val”]选择具有at属性且属性值等于val的E元素。

伪类选择器：a:hover{} 鼠标移入效果

a:link{} 没有访问过的标签

a:visited{} 访问过的标签

a:active{} 选定的链接

#taget\_0:target{}

E:not[s] 匹配不含有s选择符的元素E。

E:not(:first-chilid)

E:only-child 配父元素仅有的一个子元素E。

a:nth-child(n)

nth-child(an+b) a和b必须是整数， 且b比a小， <序号>除以a,余数是b（例如要选中6,11,16,21则用nth-child(5n+1)表示）

nth-child(even)选中偶数个

nth-child(odd)选中奇数个

样式

字体样式：

line-height:行高

font-size 字体大小（px em）

段首缩进2个字：(1)、p{font-size:16px;text-indent:2em}

(2)、p{font-size:16px;text-indent:32px}

font-family 字体集

字体:字符的编码(二进制数) <=> 图片

1，字体的名字里有中文，特殊字符，字体的名字需要用双引号包裹

2,在一个font -family中可以设定多个字体，用逗号分隔

font-weight：字体粗细 100~900 norma l:400 bold:700

font-style：normal italic oblique(后两个表斜体，倾斜，italic需要有设计的倾斜字体，oblique是人为倾斜字体)

font：连写

font: font-style font-weight font-size/line-height font-family

有些值是可以省略：font-style font -weight. line -height

font -size和font - family必须有，否则font不起作用

text-align:内容在盒子中的水平对齐方式

文字对齐方式：

水平方向：text-align:”center”（写在父盒子中）

垂直方向：line-height:”盒子高度（写在父盒子中）

text-decoration:修改文字的修饰

none:什么都没有

underline:下 划线

overline:上 划线

line - through:删除线

显示模式

display

1. block:块级元素 div hx li ol ul p

特点：独占一行 可设置宽高、内边距、外边距

块级元素的宽度是其容器(父节点)宽度的100%

块级元素内部可以容纳其他的块级元素或者内联元素

（例外：p和hx通常只放文本）

1. inline：行内元素: span, a,文本格式化标签

特点：

相邻的行内元素可以在一行上

行内元素设定宽度和高度无效

行内元素可以设定内边距，但是设定外边距时，只有水平方向有效

行内元素的宽度是其内部内容的宽度

行内元素通常只容纳文本或者其他行内元素

例外，a元素内部可以放块级元素

注意: a元素内部不能再放a元素

1. inline-block： 行内块级元素：img input td

特点：

相邻的行内块元素是可以在一行上

行内块元素可以设定宽度、高度、内边距和外边距

1. 模式转换

display:inline/block/inline-block

（inline和inline-block宽度能有多小是多小，block是能有多大是多大）

快速生成代码：div[class=box]>(a[href=#]{测试})\*5

生成结果：<div class="box">

<a href="#">测试</a>

<a href="#">测试</a>

<a href="#">测试</a>

<a href="#">测试</a>

<a href="#">测试</a>

</div>

显示

1. display:none（隐藏）display:block(显示)
2. visibility:visible(默认值元素是可见的)visibility:hidden(元素是不可见的)

**display:none和visibility:hidden的区别：**

1. d….n把元素从Render tree取下，但是v….h不会

2. d….n的元素不参与 layout (不参与标准流的计算)，但是v….h参与

即1、display:none：元素隐藏，不占位置。

2、visibility:hidden：元素隐藏，占位置。

溢出：

overflow:

* visible不剪切内容，不出滚动条(缺省) .
* auto根据内容，决定使用hidden还是scroll
* hidden剪切内容，不出滚动条
* scroll剪切内容，无论如何都出滚动条，很丑

鼠标：

cursor:

default

pointer (手)

move (移动)

text (文本)

轮廓

outline

0/ none（将一些例如文本框之类的鼠标移入边框变色的效果去掉）

resize

none(对textarea实现无法改变框的大小)

如何消除img下的空白。

vertical-align: middle

display: block

文字标题过长的处理

white-space: nowrap 强制不折行， 除非遇到<br>标签

overflow: hidden 隐藏超出的文字部分

text-overflow: ellipsis 用省略号代替截掉的文字

CSS Sprite，CSS精灵，CSS雪碧图

原理:当网页需要多张背景的小图时，将多张背景图合并为一张大图，一

次请求拉到本地，然后不同元素使用CSS的background- position进行定位。

优点

减少请求次数，缩短页面加载时间

缺点

开发过程繁琐.

维护过程繁琐

当组合的图片过大且排列不当时，会消耗过多的内存

圆角文字边框原理

1.只有一个比较长的背景图片

2.用两个元素，一个表示左边，一个表示右边

3.长度自适应，撑开响应的宽度

CSS编码规范：

格式规范：

1.选择器和后面的"{"大括号之间要有一个空格

2.属性名和":"冒号之间没有空格，":"冒号和属性值之间有一个空格，结尾有";"分号， 且无空格分隔

选择器规范:

3.当选择器是并集选择器时，每个并集选择器独占一行

4.后代\子代选择器不建议超过5层:在能达到选择且层数可控的条件下，优先使用子代选择器

属性规范：

5.每个属性单起一行

CSS特性

CSS层叠性：

（1）两个选择器选到了同一个元素，并且对同一个CSS属性进行设定，那么后定义的CSS属性将覆盖先前定义的CSS属性

（2）如果是针对不同的CSS属性进行设定，互不影响

（3）如果有些CSS属性冲突，有些不冲突，那么冲突的层叠，不冲突的互不影响

CSS的继承性：

（1）一个元素的CSS属性可以继承自其父亲或者祖辈的CSS属性

（2）如果多个级别的祖辈元素对同一个CSS属性进行了设定，那么采取就近原则.

（3）只有一部分的CSS属性是可以继承的常用的可继承的CSS属性: 字体，color. line-height. text-align. text-indent

CSS优先级:

原则：

（1）如果优先级不同，高优先级的选择器将覆盖低优先级的选择器的CSS属性

（2）如果优先级相同，定义在后的选择器将覆盖定义在先的选择器的CSS属性

判断优先级：

\*，继承= 0,0,0,0

元素(标记) = 0,0,0,1

类，伪类= 0,0,1,0

ID = 0,1,0,0

行内= 1,0,0,0

!important =∞

/\*列出以下选择器的优先级\*/

div ul 1i {}

/\* 0,0,0,1 + 0,0,0,1 + 0,0,0,1 = 0,0,0,3 \*/

.nav ul>li {}

/\* 0,0,1,0 + 0,0,0,1 + 0,0,0,1 = 0,0,1,2 \*/

a :hover {}

/\* 0,0,1,1 \*/

nav a{}

/\* 0,0,1,1 \*/

#nav P {}

/\* 0,1,0,1 \*/

background

/\*背景颜色\*/

background-color: #0ff;

/\*背景图片\*/

background- image: ur1l( image/me . jpg);

/\*平铺模式\*/

/\* no-repeat: 不平铺\*/

/\* repeat-x: 延水平方向平铺\*/

/\* repeat-y: 延垂直方向平铺\*/

background- repeat: no- repeat ;

/\*平铺图片的位置\*/

/\*两种:1左中右 上中下left, right, center, top, bottom, center \*/

background- position: right bottom;

/\* 2. px先水平方向(x)，再垂直方向(y) \*/正右负左 正下负上

background- position: 15px 30px;

/\*问题，左右防线偏移30px,垂直方向居中?

/\*. 是网站显示图标的一般办法\*/

background- position: 30px center;

用background属性将上面所有的设置项放在一起

background: #00FF00 url(bgimage.gif) no-repeat 10px center;

样式

border:

border-redius:（左上 右上 右下 左下）与 （上 右 下 左）对应

border-style: none, solid（实线）, dashed（虚线）, dotted（圆点虚线）

border-color:边框颜色

border-width:边框宽度

* border-width:上 右 下 左（四条边宽度都不同）
* border-width:上下 右左（上下边宽度相同、左右边相同）
* border-width:上 左右 下（左右边相同）
* border-width:上下左右（四条边都相同）

该方法适用于border-color

统一写法：

border：宽度 样式 颜色

border-top/right/bottom/left

padding:内边距

什么情况下，padding不会撑开盒子：

1. 当宽度没有设定的时候，padding不会撑开宽度
2. 当高度没有设定的时候，padding不会撑开高度

margin：外边距（可以为负数，margin-top: -10px 向上移动）

外边距的合并（外边距塌陷）：块级元素的上下外边距有时候会合并，合并后的外边距等于合并前两个外边距中的较大值

两个盒子的真实距离，不是两个盒子的margin之和，而是两个margin中最大的那个;不要给两个盒子同时设定margin, 把margin的值设定到一个盒子上。

详解：[外边距塌陷](https://www.jianshu.com/p/cac24427d537)

/\* 解决塌陷 \*/

（代码：Hbuilder中web学习/外边距塌陷）

/\* 1、给父div添加padding-top 子div不需要再margin-top同时调整height \*/

/\* padding-top: 50px; \*/

/\* 2、给父div添加boder-top 同时调整height 子div需要margin-top \*/

/\* border-top: 1px solid #FF0000; \*/

/\* 3、给父div添加overflow: hidden防止塌陷 子div需要margin-top \*/

/\* overflow: hidden; \*/

<!-- 4、添加行内元素 -->

居中

块级元素

水平：margin:0 auto;

行内元素

水平：text-align:center;

垂直：line-height:

去除内外边距:不同浏览器的margin padding缺省不同(第二种效率更高)

\* { body,div（写所有块级元素） {

margin:0; margin:0;

padding:0 padding:0;

} }

盒子阴影：

box-shadow:水平阴影(必选) 垂直阴影(必选) 模糊距离 阴影尺寸 阴影颜色

水平阴影：正为右 垂直阴影：正为下

布局：

1. 标准流
2. 浮动：就是使元素脱离标准流，移动到其父元素指定的位置的过程

（1）浮动的子元素无法超出父元素范围，如果父元素有padding,那么子元素无法超出padding

（2）在工程中，一个父元素下，尽量让所有的子元素都浮动

（3）一个块级元素浮动以后，会产生类似inline-block的特性， 宽度是根据自己的内容的最小宽度，而不是父元素的最大宽度。

（4）一个行内元素浮动以后，会产生类似inline-block的特性， 可以设定宽度和高度|

（5）一行内有多个div前面的都是float:left最后一个float:right

页面布局：页头header导航栏nav内容content 页脚footer

快速生成：#header+ #nav+# content - main+#content - 2+#footer

<div id="header">k/div>

<div id="nav"></div>

<div id= " content- main"></div>

<div id="content-2"></div>

<div id=" footer"></div>

一个块中有几个小块，已知总宽度，margin的宽度:

1.求块的宽度:(总宽度-margin的宽度\*(n-1))/n向下取整

我们知道:n\*块宽度+(n-1)\*margin的宽度<总宽度:有一个差

2.求出差

3.最后一个块float right

有一个问题，就是最后一个块和倒数第二个块之间的margin = margin +差，且这个差小于n

4.调整一部分块的margin-right

得到了一张图片( windows截图)如何确定尺寸

液晶屏的像素

DPI (每英尺液晶像素的个数，是液晶屏的物理参数)

window本身的缩放，Chrome的缩放

2.使用画笔，用选择框选中，像素数/window缩放比例

3.使用PS，用切片工具，像素数/window缩放比例

如何清除浮动：

1. 为什么要清除浮动：在某些情况下，我们不能设定父元素的高度，但是所有的子元素都是浮动的(脱标)，父元素的高度为0，父元素后面的元素会上提。
2. 清除浮动的四种方法

（1）额外标签法：在最后增加一个额外的div，用clear: both去 掉浮动属性 W3C推荐的一.种方法 增加了一个莫名其妙的标签

（2）overflow:hidden:强制父元素计算内部高度

（3）after伪对象:相当于（1）

.clearfix:after {

content :””;

display: block;

height: 0;

clear: both;

visibility: hidden;

}

（4）before、after伪元素:相当于在子div之前加一个额外的div

.clearfix:after ,

.clearfix:before {

content :””;

display: table;

}

.clearfix:after {

clear: both;

}

对于IE6、IE7（3）（4）不能识别，需要添加.clearfix {\*zoom: 1;}

且这个Chrome不能识别，不会对其造成影响

1. 定位

（1）static

所有不设定定位的元素，定位方式都是static (缺省)

1.不脱离标准流

2.偏移不起作用: top（正向下移动） right（正想左移动） bottom（正向上移动） left（正向右移动）

3.作用：消除定位

（2）relative：相对定位

1.不脱离标准流

2.偏移起作用: top right bottom left，起始点为元素本来应该所在的位置的左上点

3.不破坏标准流，把元素移动到需要的地方

relative只是移动了元素，但是标准流不受影响

relative+偏移vs static+ margin的区别?

margin虽然也满足移动元素，但是会影响标准

4. relative作为absolute的基准

5.如果同时设定了left和right，只有left有效: 如果同时设定了top和bottom，只有top有效。

（3）absolute:绝对定位

1.脱离标准流，跟着滚动条一起滚动

2.偏移起作用: top right bottom left

没有任何一一层的祖先节点有定位(relative，absolute, fixed) ，以浏览器的可视区（屏幕）为起始点

父元素有定位( relative，absolute, fixed) ，以父元素的内角( border和padding的分界线)为起始点;

margin对子元素定位有影响，padding，border对子元素定位没有影响

如果有一个或者多个祖先元素有定位( relative，absolute， fixed) ，以最近的祖先元素的内角( border和padding的分界线)为起始点，

设定了position: absolute，没有设定left，top

（1）如果没有设定left和top，所有对元素能产生所用的因素全都起作用:注意，与top:0; left:0;不等价，不设定left和top的情况，left和top被 初始化为"应该在”的位置。

（2）如果left被设定，top没有: left按照设定，top按照“应该在”的位置。

（3）如果top被设定，left没有: top按照设定，left按照“应该在”的位置。

3. relative作为absolute的基准（父relative子absolute）

absolute元素如何居中:

第一步，设定top，left为50%

第二步，通过margin-left， margin-top设定 负值

（4）固定定位：fixed

1.脱离标准流，不和滚动条一起滚动

面试题:能脱离标准流的:浮动，absolute, fixed

2.偏移起作用: top right bottom left

3.不受父元素的影响，只依据浏览器的可视区定位。

4.当一个元素被设定为固定定位时，会产生模式转换=》行内块元素，所以如果开始没有设定宽度，设定后宽度为最小

**注：所有脱离标准流的动作，都会触发模式转换，对应元素变为行内块元素（浮动、position:absolute、position:fixed）如果一个类中既有浮动又有relative,那么还是脱离标准流**

1. 叠放次序 z-index

用于设定重叠元素的覆盖关系的

设定为一个数值，没有单位

数值越大，越靠上，可以覆盖数值小的元素

如果没有预先设定z-index， 那么缺省是0， 后定义的元素在新定义的元素之上。

设定z-index只对定位设定为relative, absolute, fixed三种元素有效，对标准流，浮动无效。

CSS的压缩

原因:调试代码和运行代码时需求不一-样，调试代码时希望代码可读性足够的好，运行时希望代码文件足够的小

压缩的工作:

* 删除注释、空格和换行
* 删除多余的分号
* 删除空的CSS样式

压缩的工具

命令行的工具 YUI Compressor

命令行的工具

YUI Compressor

1.安装jdk的环境

2.配置jdk环境，确定其已经装好

3.下载并配置yuicompressor

4.运行

压缩步骤：先找到要压缩的文件地址



网页的在线工具 Clean-Css(http://refresh-sf.com)

CSS选择器的使用哲学：

标签选择器:范围太大，用于定义全局的缺省样式

ID选择器:排他的，只用于设定body的子元素，用于布局:组件不可以使用ID选择器

类选择器:有比较好的灵活性和适应性的:不要在过多的元素上添加类

爬DOM树的方法是经常的使用方法

伪类选择器: hover; 从集合中选择其中的一部分

伪对象选择器:消除浮动:响应式布局

分组选择器: 用于合并相同的样式|