**DAY 1**

常用浏览器——谷歌（Chrome）、火狐（Firefox）、Safari、IE、Opera

浏览器内核分成两部分：渲染引擎和JS引擎

网页的构成

结构：Html

表现：CSS

行为：JavaScript

HTML介绍

超文本标记语言（Hyper Text Markup Language），超文本就是指超链接

编辑工具——vscode

新建文件

保存文件——ctrl + s

撤销操作——ctrl + z

页面骨架——! + 回车

html标签——网页的根节点，有两个子标签head和body

head标签——用于存放title，meta，style，script，link等标签，包含了浏览器和搜索引擎使用的其他不可见信息

title标签——页面的标题，搜索引擎搜索时最先看到的最醒目的内容

body标签——页面的主体部分。注意：在<body>中多个空格或换行符，会被当做一个空格来处理。

标签属性：

<标签名 属性1=”属性值1” 属性2=”属性值2” …>内容</标签名>

标签可以拥有多个属性，之间不分先后，均以空格分开，属性都有默认值，省略该属性则取默认值

注释标签：

<!-- 注释语句 -->

HTML文件中使用Ctrl + / 快捷键创建注释

常用标签：

<!DOCTYPE> 文档类型

<meta> 设置网页的元数据，不同的属性会使<meta>标签具备不同的功能

如：<meta charset=”utf-8 />指定字符集编码为utf-8（优化后的全球码）

<meta name=”keywords” content=”xxx” />用于设置关键字

<meta name=”description” content=”xxxx” />用于设置描述信息

标题标签 <h1~6>标题文本</h1~6>

段落标签 <p>文本内容</p>

水平线标签 <hr />

换行标签 <br />

图像标签 <img src=”图像URL” />

链接标签 <a href=”跳转目标” target=”目标窗口的弹出方式”>文本或图像</a>

列表标签（组合标签）：

无序列表——ul 包含 li

Emmet语法：如ul>li\*5

有序列表——ol 包含 li

列表具有严格的嵌套关系，<ul></ul>中只能嵌套<li></li>，<li></li>之间相当于

一个容器，可以容纳所有元素

自定义列表——dl dt dd生成一个类似于目录的结构

<dl>

<dt>名词</dt> 目录

<dd>解释</dd> 🡺 解释

<dd>解释</dd> 解释

</dl>

路径：（相对路径和绝对路径）

相对路径

相对当前文件

同一级目录下，输入图像文件的名称即可，如<img src=”avatar.jpg” />

图像文件位于当前文件的下一级目录：如<img src=”img/avatar.jpg” />

图像文件位于当前文件的上一级目录：<img src=”../avatar.jpg /> 上两级：<img src=”../../avatar.jpg />

绝对路径

本地绝对路径 D:/web/img/avatar.jpg

网络路径 <http://w3school.com.cn/i/html_editor_notepad.gif>

**DAY 2**

**HTML基本规范（语法规范）**

注释不能嵌套；标签必须完整；标签可以嵌套，但要注意语义；标签的属性必须加双引号；标签属性尽量小写；

标签的语义化

div 和 span标签没有任何的语义

两个常用的布局标签：

借助div标签对网页进行分块，div会自动换行

span用于处理小的地方（局部），通常会包裹文字。span不会自动换行，这一行排满了才会换行

文本格式化标签：

加粗——strong、b

斜体——em、I 区别在于语义化

上标和下标——sup、sub

下划线和删除线——ins、del

引用文本——blockquote(长引用)、p(短引用)

代码——code

预格式化文本——pre 空格和回车等格式都会被保留

转义字符：

空格 &nbsp; &#160

小于号 &lt; &#60

大于号 &gt; &#62

和符号& &amp

版权© &copy

注册商标® &reg

商标™ &trade

表格

<table>-----------------表格：用于定义一个表格

<caption>标题</caption>------标题：显示在表格上方

<th>表头1</th>-----表头：表头一般位于表格的第一行或第一列

<th>表头2</th>

<tr>----------------表格中的行

<td></td>------表格中的单元格

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

合并单元格——<td rowspan=”2”></td>合并两列的单元格，<tr cowspan=”2”></tr>合并两行的单元格

**CSS（Cascading Style Sheets层叠样式表）**

语法：习惯一行只写一个样式，方便阅读

选择器 {

样式名：样式值；

样式名：样式值；

……

}

简单(基本)选择器：

1. 标签名选择器（通过标签名来选择）比如p div a h1
2. id选择器（通过id选择，前面要加#）

id就是这个标签的标识，类似于身份证，独一无二

1. 类选择器（通过class名选择，前面加 ”.”）
2. 后代选择器（用空格隔开）
3. 交集选择器（没有空格隔开）
4. 并集选择器（用逗号隔开）
5. 可以同时选中页面中的所有元素

行内样式

直接给标签添加style属性，进行样式修改。定位准确，但是样式和结构严重耦合，而且样式不能复用。

如：<div style=”color: blue; font-size: 24px;”>行内样式</div>

内部样式

head标签里面的style标签，样式也不太能复用

<head>

<style>

…

</style>

</head>

外部样式

新建一个css文件保存样式，使用link标签引入。样式可以复用，工程中最常用的方式

<head>

<link rel=”stylesheet” href=”outstyle.css” >

</head>

CSS的继承性：

子类元素会继承父类元素的某些属性，比如字体颜色。特殊：a标签的字体颜色无法继承，a标签必须选中才能修改颜色。

CSS权重问题：同时满足多个选择器，哪个样式生效？

标签的样式使用你设置的很多样式共同作用的结果。如果样式冲突，权重高的生效。

简单选择器的权重：

行内样式(1000) > id选择器(100) > class选择器(10) > 标签名选择器(1) > 通配符和继承属性(0)

复杂选择器的权重：累加

如：.box .inner-box { } 10+10=20

.text.red { } 10+10=20

如果权重一样怎么办？后写的生效

**DAY 3**

用户输入框

通过type属性的不同，input的特性也不同

输用户名：<input type=”text” placeholder=”请输入用户名/邮箱”>

输密码：<input type=”password”>

按钮

<button>这是一个按钮</button>

CSS常见单位

长度

px像素单位

百分比，相对于父类元素的百分比

em基于当前字体的倍数

颜色：三种表示形式

预定义颜色：blue、yellow、pink、purple、red等

十六进制：#333333可以省略成#333，是由指定浓度的红绿蓝混合而成

rgb()：如rgb(221, 0, 27) 也是由红绿蓝混合而成，就相当于把16进制换算成10进制的结果

rgba()：带有透明度的rgb，rgba(221, 0, 27, 0.5)最后一个参数表示透明度，0完全透明，1完全不透明

字体相关属性

宽 width

高 height

颜色 color

背景颜色 background-color

字体大小 font-size

字体 font-family，可以设置多个字体，浏览器优先使用第一个，如果没有找到则使用第二个，以此类推

粗细 font-weight，可以写100~900的数值，也可以写normal、bold、bolder，400=normal，700=bold

斜体 font-style，italent表示斜体

行高 line-height，行高会影响文字行间距，行高默认是字体的1.3~1.4倍之间。行高为数字的时候，表示是当

前字体大小的倍数

对齐方式 text-align，center居中，left左对齐，right右对齐，justify两边对齐

垂直居中 技巧：设置盒子的行高（line-height）等于盒子的高度

字符间距 letter-spacing，word-spacing

缩进 text-indent

文本修饰 text-decoration，underline下划线，overline上划线，line-through删除线，none没有修饰

图片

引入图片 通过img标签引入图片，<img src=”图片路径” alt=”图片无法显示时显示的文字”>

CSS引入 .box{ background-image: url( 图片路径 );} 引入后自动平铺，设置background-repeat: no-repeat

取消平铺。必须要有标签来放置图片，而且标签得有大小。

图片位置 background-position: left bottom; 第一个值表示水平方向(左中右)，第二个值表示竖直方向(上中下)

第二个值不写时，默认是中间位置。除了关键词，也可以是具体的像素值。

简写：background: 颜色 图片路径 平铺 位置；

背景滚动 background-attachment，值为fixed图片不随网页滚动，scroll图片随网页滚动

背景尺寸 背景图片设置大小background-size: 85px 30px; 或者background-size: 100% 100%; 背景图片会铺

满整个盒子，可能会产生形变

CSS sprite

又叫做CSS精灵图或者CSS雪碧图，是把多张图片整合在一张图片上，避免了向服务器多次http请求图片，提高了网页的性能，也减轻了命名的压力。通过控制盒子的大小，让不相关的图片看不见。

标签的表现形式

块级元素 可以设置宽高；独占一行；如：div、ul、li、ol、dl、dt、dd、table、p、h

行内元素 设置宽高无效；同行展示，不会自动换行，这一行排满才会换行；如：span、a、em、strong

行内块级元素 设置宽高有效；同行展示；如：img、input、button

设置/改变标签的表现形式 display: inline-block/inline/block；

**DAY 4**

盒子模型

content 内容区，放置内容的区域，如果没有设置内边距和边框，则内容区大小默认和盒子大小是一致的。通

过width和height设置内容区的宽和高

padding 内边距，元素内容和元素边框之间的空间。 padding: 10px 20px; 可以是两个、三个、四个值，按照顺时针顺序上右下左，省略的值和对侧的相同。也可以通过padding-top、padding-right、padding-bottom、padding-left单独设置

border 边框。border: 10px solid #333; 第一个值边框宽度，第二个值边框样式，第三个值边框颜色。也可以通过border-style设置边框样式，none默认，solid实线，dashed虚线，dotted点线，double双实线。border-width边框宽度，border-color边框颜色。单独设置某一方向边框，同padding

margin 外边距，元素之间边框与边框之间的距离，设置margin和padding类似。margin: xxx auto; 可以使块级元素水平居中，但是上下没办法auto，因为网页上下高度是不确定的。

margin重叠现象 垂直方向上的margin不会累加，而是会重叠，谁大听谁的。水平方向上的margin会累加。

嵌套崩塌 给子类盒子设置的margin-top，子类盒子不会生效，而是直接作用到了父类盒子身上

解决方法：1. 给父类盒子添加overflow: hidden;

2. 给父类盒子添加一个极小的padding或者border

注意：给行内元素设置margin、padding不会生效

opacity 透明度。取值0~1，0表示不透明，1表示完全透明。opacity影响的是整个盒子，rgba透明度设置的是背

景色，只影响背景色。

overflow 超出部分如何显示。默认值是visible。overflow: hidden;超出部分隐藏。scroll是添加滚动条。

overflow: auto;智能添加滚动条，哪个方向超出了就在哪个方向添加滚动条。

使用最多的场景是overflow: hidden;解决嵌套崩塌问题，还可以清除浮动带来的影响；

超出部分显示省略号的方法：（三行代码）

white-space: nowrap; //不换行

overflow: hidden; //超出部分隐藏

text-overflow: ellipsis; 文字超出部分显示省略号

显示和隐藏

.box {

width: 100px;

height: 100px;

}

.p {

display: none; //隐藏，消失掉，不再占据位置，下方内容上移

}

.box:hover p { //选中hover状态下的box中的p标签

display: block; //只要不是none，元素都会显示

}

.text {  
 visibility: hidden; 也是隐藏，依然占据位置，只不过看不见了而已

}

vertical-align 设置对象内容的垂直对齐方式，简单点说就是行内块元素(如图片)与文字的对齐方式。

默认行内块元素沿着文字的基线baseline对齐，vertical-align: middle/bottom/top;

使用场景：图标和后边的文字对不齐的时候可以使用这个属性进行调整。

当用middle对不齐的时候，就写具体数字，一般在-1px到-4px之间

border-radius 边框圆角，border-radius: 50%;可以让正方形变成圆

**DAY 5**

文档流

将窗体自上而下分成一行行，并在每行中按照从左至右的顺序排放顺序，即为文档流。

浮动 float: left/right;

浮动指的是使元素脱离原来的文本流，在父元素中浮动起来。

浮动可以让块级元素在同一行展示。

当一个块级元素浮动以后，宽度会默认被内容撑开，所以当浮动一个块级元素时我们都会为其制定一个宽度。

当一个元素浮动以后，它会脱离文档流，不再在文档中占用位置，其下方的元素会上移，上移元素中的内容会

围绕在浮动元素的周围(文字环绕)

元素浮动以后脱离文档流，浮动元素不会撑开父元素的高度。所以我们必须清除浮动的影响，防止布局错乱。

方法1.严格设置父类元素的高度。

方法2.给父类元素添加overflow: hidden;（推荐方法）

方法3.在浮动元素的最后追加一个空的div，设置clear: both;属性

块级元素和行内元素都可以浮动，当一个行内元素浮动以后将会自动变为一个块级元素，即使设置display: inline;

以后其依然是个块元素

元素设置浮动以后，会一直向上漂浮，知道遇到父元素的边界或者其他浮动元素

遵循一个原则：

子元素要浮动，都浮动

解决网页中的一些局部问题

当标签位置、正常文档流和浮动都不好解决的时候，就可以选择定位解决。

定位

position属性可以把元素的定位到网页中的任何位置，元素定位后，需要通过设置left/right和top/bottom值对元素进行定位。

相对定位 relative，相对于元素在文档流中原来的位置。开启了相对定位以后，可以使用top、right、bottom、left

四个属性对元素进行定位。相对定位不会使元素脱离文档流，元素在文档流中的位置不会改变。元素的

层级会得到提升，使元素可以覆盖文档流中的元素。

相对定位的使用场景：通常都是打辅助，配合绝对定位实现元素的定位

绝对定位 absolute，使元素以视口或者离它最近的祖先定位元素为参照物，进行定位，从自己的父级元素一个一

个网上查找，知道找到一个有定位属性的元素为止，此时以它为参照进行定位。如果找到body也没有，

那就以视口为参照物进行定位。元素会脱离文档流，下方的元素会上移。

定位属性指：position: relative/absolute/fixed/sticky;

绝对定位的块元素的宽度会被其内容撑开，和浮动一样。

绝对定位会使行内元素变成块元素，和浮动一样。

使用技巧：“子绝父相”，一般使用相对定位时，会同时为其父元素制定一个相对定位，以确保元素可以

相对于父元素进行定位。加了相对定位，元素本身的特性没有改变，没有任何副作用。

固定定位 fixed，固定定位也叫浏览器定位，参照物是浏览器窗口，不会随着浏览器滚动而滚动。

特性改变，宽度由内容撑开。行内元素会变成块元素。

粘性定位 sticky，可以任务是相对定位和固定定位的混合。元素在跨越特定阈值前为相对定位，之后为固定定位。

比如滚动吸顶效果。

格式化快捷键 shift + alt + F

层级 借助z-index属性，调整定位元素的层级。初始值0，整数，取值范围0到无限大，值越大层级越高。可以

设置为负数，表示离用于更远，一般不这么用。只有定位元素才能设置层级。

规避脱标流 经验：一般布局采用标准流，如果标准流布局实现不了，用浮动。定位一般用于解决小范围的某个标

签的位置。

块级元素垂直居中？

定位解决：position: absolute;

top: 50%;

margin-top: -自身高度的一半;