# css总结笔记

**css常识**

**作用：**设置网页的外观样式

**注释**：/\* \*/

**获得焦点：**鼠标点击选中元素或通过 tab 键定位到元素时

**样式书写顺序：**

布局定位属性：display/position/float/clear/visibility/overflow/z-index

自身属性：width/height/margin/padding/border/background

文本属性：color/font/text-decoration/text-align/vertical-align/white-space/break-word

其他属性(CSS)：context/cursor/border-radius/box-shadow/text-shadow/background:linear-gradient....

**(0, 0)坐标点：**左上角，向下，向右为正

**颜色写法：**

pink

#FFFFFF

rgb(0,0,0)

rgba(0,0,0,0.3)

**CSS三个特性**

**1.层叠性**

就近原则，哪个样式近，用哪个样式（覆盖之前的样式）

**2.继承性**

孩子会继承父亲的某些样式（text-,font-,line-,color）

**3.优先级（样式权重越大，优先级越高）**

!important > 行内样式 > id选择器 > 类选择器和伪类选择器 > 标签选择器 > 继承/\*

1000 0100 0010 0001 0000

复合选择器：权重叠加，不进位

**css引入方式**

1.行内样式：写在行内

<div style=**"color:red; font-size:15px;"**>**hello**</div>

2.内嵌样式：样式写在head中，行内用选择器引入

<style type=**"text/css"**>

**.pink {**

**color: pink;**

**}**

**.fontSize15 {**

**font-size: 15px;**

**}**

</style>

<div class=**'pink fontSize15'**>**hello**</div>

1. 外部样式表(样式多推荐使用)：样式写在外部文件中，head中引入外部文件，行内用选择器引入

<link rel=**"stylesheet"** type=**"text/css"** href=**"resources/css/style.css"**/>

<div class=**"skyblue"**>**hello**</div>

**css选择器：**

**css结构**

选择器：{

属性:值;

属性:值;

}

**基本选择器**

**1.标签选择器： 标签名{}**

p**{}**

**2.类选择器： .类名{}**

**.**red**{}**

**3.id选择器： #id名{}**

id值是唯一的所以只能被引用一次

#qwe**{}**

**4.通配符选择器：\*{}**

定义全局所有标签的属性

\***{}**

**复合选择器**

**1.后代选择器： 选择器1 选择器2 {}**

表示：选择器1中的所有选择器2具有的属性

div a**:{}**

**2.子选择器： 选择器1>选择器2 {}**

表示：选择器1中最近一级的选择器2具有的属性

div**>**a**:{}**

**3.并集选择器： 选择器1, 选择器2{}**

表示：选择器1和选择器2都具有的属性

div, a{}

**4.伪类选择器： 选择器1: 伪类{}**

表示：选择器1在某种伪类状态下具有的属性

input **:focus{}**

**伪类选择器(按顺序写)**

**.**class2**,** a**:link** **{** /**\* 超链接点击前，点击后需要清理cookie才能变为点击前 \*/**

**color: red;**

**}**

**.**class2**,** a**:visited { /\* 超链接访问后 \*/**

**color: orange;**

**}**

**.**class2**,** a**:hover** **{ /\* 鼠标悬停时 \*/**

**color: green;**

**}**

**.**class2**,** a**:active** **{ /\* 鼠标按下(不松开)时 \*/**

**color: pink;**

**}**

**.**class2 input**:focus** **{ /\* 标签获得焦点时 \*/**

**border: 2px solid pink;**

**}**

**emmet语法**

**快速生成标签**

生成标签 标签名 + tab

生成多个相同标签 标签名\*10 + tab

生成父子关系标签 ul>li + tab

生成兄弟关系标签 div+p + tab

生成带有类名或id名字的标签 div + .demo/#two + tab

生成的div类名是有顺序的, 可以用自增符号$ div.demo$\*5 + tab

生成的标签内写内容 {}表示 div{你好} + tab

**快速生成样式**

打每个样式的首字母就好

**重排代码格式**

Ctrl+K

**盒子模型**

border 边框

padding内边距（盒子与内容的距离）

margin 外边距（盒子之间的距离）

总大小 = 盒子大小 + 边框大小 + 外边距大小

盒子本身未指定宽高属性，则padding就不会撑开盒子，如果有盒子宽高会包含padding

**外边距塌陷**

对于两个嵌套关系的块元素，父元素有上外边距同时子元素也有上外边距，此时父元素会塌陷较大的外边距值

解决方案：

为父元素定义上边框

为父元素定义上内边距

为父元素添加 overflow:hidden（常用）

浮动，固定，绝对定位 不会有塌陷问题

行内元素只设置左右的内边距（或者转换为块级和行内块元素）

**布局方式**

**标准流**：

多个块级元素纵向排列时用（在地面）

**float：浮动**

多个块级元素横向排列时用（相当于在空中，会压住标准流）

浮动元素会部分脱离标准流，原有位置不保留，压住块元素，压不住行内/行内块元素

浮动元素会在一行内显示，根据顶端对齐排列

浮动元素具有行内块元素的特性

**position：定位**

可以自由的在某个盒子内移动位置，或者固定屏幕中某个位置，可以压住其他盒子

**排列规则**

先用标准流排列上下位置，内部子元素浮动排列左右位置

内部子元素一个浮动，所有子元素都要浮动

浮动的盒子只影响浮动盒子后面的标准流，不影响前面的标准流

先定大小，后定位置

**css属性**

**align: 水平居中**

**vertical-align: middle;** 图片后面文字位于图片垂直居中

**text-align: center;** 行内元素居中

**margin: 0 auto;** 块级元素居中

**display：元素类型**

block: 显示为块元素

inline: 显示为行内元素

inline-block: 显示为行内块元素

none: 隐藏，不在网页中占位置

**隐藏元素**

**1.display: none; 从页面中剥离，无位置**

**2.visibility: hidden; 仍然在页面中有位置，只是看不见**

**hidden:隐藏,**

**visible:可见**

**3.overflow: scroll;**

**visible:无滚动条显示**

**hidden:隐藏溢出**

**scroll:有滚动条显示**

**auto:溢出的时候显示滚动条，无溢出则无滚动条**

**white-space: nowrap;** /\* normal:自动换行, nowrap:强制一行内显示 \*/

**text-overflow: ellipsis;** /\* 省略号替代超出的部分 \*/

**background:背景**

background-color**:** pink**;** /\*背景颜色，最下层 \*/

**background-image:** url**(..**/img/小蒜头王八**.**jpg**);** /\* 背景图片 none:无背景图片 ,url("地址") \*/

**background-repeat:** repeat**;** /\* repeat:平铺, no-repeat:不平铺, repeat-x:x轴平铺, repeat-y:y轴平铺 \*/

**background-position:** 10px 20px**;** /\* x y :数值 / (left|center|right)(top|center|bottom); \*/

**background-attachment:** scroll**;** /\* scroll:背景图像随着内容滚动, fixed: 背景图像固定 \*/

/\* 复合写法顺序: 颜色 图片 平铺 滚动 位置 \*/

**background: pink url(../img/小蒜头王八.jpg) repeat scroll center top;**

**border:边框**

/\*border-width:5px;\*/

/\*border-style: solid;\*//\* solid:实线边框, dashen:虚线边框, dotted:点线边框\*/

/\*border-color: blue;\*/

**border: 5px solid red;** /\*简写方式，无序\*/

/\* border-top:5px solid red;\*//\*top,bottom,right,left:分别设置边框 \*/

/\* border-collapse: collapse;\*//\*合并相邻的边框 \*/

**border-radius: 100px;** /\* 圆角边框 值为圆角的半径 \*/

/\* border-top-left-radius: 100px; \*/

/\*简写方式 左上开始 顺时针方向\*/

/\* border-radius: 50px 60px 70px 80px; \*/

**padding:内边距**

**padding: 20px;**

/\* padding-left: 20px;\*//\*top,bottom,right,left:分别设置内边距\*/

/\* padding: 20px 15px;\*//\*上下|左右\*/

/\* padding: 20px 15px 20px\*//\*上|左右|下\*/

/\* padding: 20px 15px 20px 15px\*//\*上|右|下|左 顺时针\*/

**margin:外边距**

/\* margin: 20px; \*/

/\* margin-left: 20px;\*//\*top,bottom,right,left:分别设置外边距\*/

/\* 简写和padding一致 \*/

**margin: 20px auto;** /\* 居中：对块元素有效 \*/

**text-align: center;** /\* 居中： 对行内元素和行内块元素有效 \*/

**box-sizing:盒子大小**

**box-sizing: content-box;** /\* 盒子大小 = width+padding+border(默认) \*/

**box-sizing: border-box;** /\* 盒子大小 = width \*/

**font:字体**

/\* font-size: 20px; \*//\* 字号 \*/

/\* font-family: '宋体'; \*//\* 字体 \*/

/\* font-weight: bold; \*//\* 字体粗细 bold/700:加粗 normal/400:只变大\*/

/\* font-style: normal; \*//\* 字体样式 normal:让倾斜的字体不倾斜 italic:让字体倾斜 \*/

/\* font简写, 按序, size和family不能省略，

font: font-style font-weight font-size/line-height font-family;\*/

**font: italic 700 16px** '宋体'**;**

**text:文本**

**text-align: left;** /\* 水平对齐方式: center, right, left \*/

**text-decoration: underline;**/\* 添加装饰, none:没有装饰线(取消链接下划线), underline:下划线, line-through:删除线, overline:上划线 \*/

**text-indent: 2em;** /\* px/em为单位,1em指当前1文字的大小 \*/

**line-height: 30px;** /\* 行高 = 上间距 + 文本高度 + 下间距 行高=盒子高度，则文字垂直居中 \*/

**shadow:阴影**

/\*box-shadow: h-shadow v-shadow blur spread color inset;

box-shadow: 阴影不占空间

h-shadow:水平阴影位置,必写

v-shadow:垂直阴影位置,必写

blur:模糊距离

spread:阴影尺寸

color:阴影颜色

inset:将外部阴影(outset)改为内部阴影 ; \*/

**box-shadow: 10px 10px 10px 10px rgba(0,0,0,0.3);**

**text-shadow: 5px 5px 5px rgba(0,0,0,0.3);**

**cursour:光标类型（很多，自行百度）**

pointer：光标显示为小手

**float:浮动 和 清除浮动**

**float: 浮动类型**

none:不浮动

left:元素向左浮动

right:元素向右浮动

**清除浮动：**让父元素根据子元素自适应高度，如果不清除浮动父盒子height会为0

**1.额外标签法：需要添加额外标签**，不推荐使用

**.**clear **{**

**clear: both;**

**}**

<div class=**"clear"**></div>

**2.父元素添加 overflow:hidden 属性：无法显示要溢出的元素**，不推荐使用

**3.使用after为元素清除浮动：推荐使用**

**.**clearfix**{**

**\*zoom: 1;**

**}**

**.**clearfix**:after** **{**

**content:** ""**;**

**display: block;**

**height: 0;**

**clear: both;**

**visibility: hidden;**

**}**

<div class=**"clearFloat clearfix"**>

**4.使用before和after双伪元素清除浮动：推荐使用**

**.**clearfix **{**

**\*zoom:1;**

**}**

**.**clearfix**:before,**

**.**clearfix**:after** **{**

**content:** ""**;**

**display: table;**

**}**

**.**clearfix**::**after **{**

**clear: both;**

**}**

<div class=**"clearFloat clearfix"**>

**position: 定位** ，可以自由的在某个盒子内移动位置，或者固定屏幕中某个位置，可以压住其他盒子

**定位 = 定位模式 + 边偏移**

**定位模式 (position)**

**static：**静态定位：默认的，无定位

**relative：**相对定位:相对于原来位置的定位来移动

行内元素添加绝对定位/固定定位，可以直接设置高度和宽度

左上角为（0，0），不脱标

**absolute：**绝对定位: 相对于他的祖先元素的定位来移动，子绝父相原则

如果祖先元素没有定位（static不算），则以浏览器为准

如果有，则以最近一级带有定位的祖先元素为参考点移动位置

脱标

绝对定位后，不能通过margin：0 auto 来设置水平居中，需要一下方式

left:50% margin-left: -盒子width的一半

**fixed**：固定定位:以浏览器的 可视窗口 来移动

脱标

**sticky：**粘性定位：相对定位和固定定位的混合，滚轮滚到某个位置时转为固定定位

以浏览器的 可视窗口 来移动

保留原有位置（相对定位）

必须添加，top,left,right,bottom中的一个才有效果，否则就相当于相对定位

**使用条件**：

父元素不能overflow:hidden或者overflow:auto属性。

必须指定top、bottom、left、right4个值之一，否则只会处于相对定位

父元素的高度不能低于sticky元素的高度

sticky元素仅在其父元素内生效

**边偏移（top,bottom,left,right）**

决定了该元素的最终位置

left和right属性共存，默认执行left

top和bottom属性共存，默认执行top

**定位叠放次序属性 z-index：1**

只有 有定位的盒子才有的属性，static除外

数值大的盒子在上层，若数值相同，则后写的盒子在上层

**CSS高级技术**

**精灵图技术**

针对背景图使用，将多个小的背景图整合到一张大图片中

插入的图片不做精灵技术，因为要经常更换

借助背景位置来实现：background-position

目的：为了有效的减少服务器接受和发送请求的次数，提高页面的加载速度

缺点：图像文件比较大，放大会失真，更换很麻烦

**字体图标**

展示是图标，本质是字体（方便高效）

轻量级：文件小，字体加载，图标就有了，减少服务器请求

灵活性：是文字，可以随意改变颜色，阴影，透明，选择等

兼容性：支持所有浏览器

不能替代精灵技术，只对部分图标提升和优化（适合简单的小图标）

**下载方式**：

icomoon字体库：http://icomoon.io

阿里iconfont字体库：http://www.iconfont.cn/

复制@font-face，并修改好url

添加属性 font-family:’’ 为下载的字体

去demo.html中复制图标对应的 代码 到html中

**字体图标的追加**

将压缩包里的selection.json重新上传，然后选中自己想要的新图标，再下载替换原来的文件

**CSS3新特性（IE9+才支持）**

**新增选择器**

1.**属性选择器 （权重为10）**

**.**CSS3 input**[**name**]** **{** /\* 带有属性名的元素\*/

**color: pink;}**

**.**CSS3 input**[**name=user**]** **{** /\* 属性值=属性值的元素 \*/

**background-color: skyblue;}**

div**[**class^=qq**]** **{** /\* class以qq开头的元素 \*/

**color: red;}**

div**[**class$=ee**]** **{** /\* class以ee结尾的元素 \*/

**color: purple;}**

div**[**class\*=ww**]** **{** /\* class 包含ww的元素 \*/

**color: green;}**

2.**结构伪类选择器（权重为10）**

nth-child： 将所有元素都排序号（不看类型），在我们要的序号中选 类型相同的孩子

nth-of-type(n)：按照类型排序,然后在相同的 类型 中选我们要的序号孩子

**.**CSS3 li**:first-child** **{**/\* 选择第一个孩子 \*/

**color: red;}**

**.**CSS3 li**:last-child** **{**/\* 选择最后一个孩子 \*/

**color: yellowgreen;}**

**.**CSS3 li**:nth-child(**2**)** **{**/\* 选择第2个孩子 \*/

**color: yellow;}**

**.**CSS3 li**:nth-child(**even**)** **{** /\* 选择第偶数个孩子 \*/

**background-color: pink;}**

**.**CSS3 li**:nth-child(**odd**)** **{** /\* 选择第奇数个孩子 \*/

**background: orange;}**

/\*公式选择孩子 n, 2n, 2n+1, 5n, n+5:从第5个开始, -n+5:前5个 \*/

**.**CSS3 li**:nth-child(**n**+**3**)** **{**

**font-weight: 700;}**

3.**伪元素选择器（重点 ）（权重为1）**

可以用CSS创建元素（默认为行内元素），无需HTML标签，从而简化了HTML结构

::before：父元素的内容前创建

::after: 父元素的内容后创建

在文档树中是找不到的，所以叫伪元素

必须有content属性

**.**box**::**before **{**

**content:** '我'**;}**

**.**box**::**after **{**

**content:** '\e62c'**;}**

**图像模糊**

**filter: blur(20px);** /\* 图像模糊 \*/

**公式计算**

**width: calc(100% - 30px);** /\* 用公式来规定,运算符左右要有空格 \*/

**动画**

**transition:CSS过渡属性,当鼠标放到元素上时会对变换进行过渡**

**.**animation **.**demo1 **{**

**width: 100px;**

**height: 50px;**

**background-color: skyblue;**

/\* transition: 属性 花费时间 运动曲线 何时开始; \*/

**transition: width 0.5s ease 0.3s, height 0.5s ease 0.3s, background-color 0.5s ease 0.3s;**

/\* transition: all 0.5s ease 0.3s; \*//\* 变化多个属性的简写 \*/

**}**

**.**animation **.**demo1**:hover** **{**

**width: 200px;**

**height: 60px;**

**background-color: lightpink;**

**}**

**2D基本转换**

**移动：transform: translate(x,y); translateX(); translateX();**

不会影响其他元素的位置，对行内元素无效，可以用百分比（相对于自身的xy）

用来将盒子水平和垂直居中，

**旋转：transform: rotate(45deg);**

deg就是度数，正为顺时针，默认绕中心点选择

transform-origin: x y;设置中心点 可以是百分比，像素，方位名词（top bottom left right center)

**缩放： transform: scale(1.5,1.5);**

两个参数为宽的倍数，高的倍数，

一个参数：宽的倍数和高的倍数一样

优点：不会影响其他盒子可以设置缩放的中心点

**混合写法：transform: translate(200px, 200px) rotate(90deg) scale(1.5);**

顺序对转换有影响，建议将移动写在前面

**动画：from和to 等价于 0%和100%**

**@**keyframes 动画名称**{**

0% **{**

**transform: translate(0, 0);}**

50% **{**

**transform: translate(400px, 200px);}**

100%**{**

**transform: translate(0, 0);}**

**}**

div **{**

**animation-name: demo\_move;**

**animation-duration: 5s;**

**}**

**定义动画: @keyframes**

**动画名称: animation-name: demo\_move;**

**持续时间: animation-duration: 5s;**

**速度模式: animation-timing-function: ease;**

ease(默认)：慢-块-慢

linear:匀速

ease-in:低速开始

ease-out:低速结束

ease-in-out:低速开始和结束，

steps()：指定时间函数中的间隔数量（步长）

**开始时间: animation-delay: 2s**

**播放次数: animationn-iteration-count:infinite;**

数字/infinite:表示无限循环

**是否反向播放: animation-direction:normal;**

alternate：反向播放，

normal(默认)：不反向播放

**结束后状态: animation-fill-mode:forword;**

forword:保持

backwards：回到起始

**鼠标经过是否暂停: animation-play-state:pause;**

running:默认值

paused:暂停

**简写：**

animation: 动画名称 持续时间 运动曲线 何时开始 播放次数 是否反方西 动画起始/结束的状态

**3D动画**

**z轴**：（屏幕到图像的距离）

往外是正，往里是负

**透视：perspective（人到图像的距离）**

透视写在被观察元素的父盒子上面

目前浏览器都不支持 perspective 属性。

Chrome 和 Safari 支持替代的 -webkit-perspective 属性。

# 移动端开发

**基本知识**

**私有前缀**

-moz-: firefox浏览器私有属性

-ms-: ie浏览器私有属性

-webkit-: chrome,safari私有属性

-o-: Opera私有属性

示例写法（老版本浏览器）

-moz-border-radius: 10px;

-webkit-border-radius: 10px;

-o-border-radius: 10px;

border-radius: 10px;

只处理webkit内核浏览器就好，放心的使用H5和CSS3标签

视口：浏览器显示页面内容的屏幕区域

理想视口：宽度是设备的宽度

**mete视口标签**

<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0" />

width=device-width 设置宽度是viewport宽度，

user-scalable=no 用户是否可以缩放，yes/no 1/0

initial-scale=1.0 初始缩放比，大于0的数字

maximum-scale=1.0 最大缩放比，大于0的数字

minimum-scale=1.0 最小缩放比，大于0的数字

**物理像素：**屏幕上的最小颗粒

在移动端1px不一定等于1个物理像素

物理像素比：1px能显示的物理像素点的个数，称为物理像素比/屏幕像素比

**二倍图**

我们准备的图片，比我们实际需要的大小 大两倍，就是2倍数图

**移动端开发主流方案**

单独制作移动端页面（主流）

响应式页面兼容移动端

**移动端CSS样式初始化使用：normalize.css**

网址：http://necolas.github.io/normalize.css/

**CSS3盒子模型：**padding和border不会撑大盒子（移动端使用）

box-sizing:border-box; css3盒子模型

content-box传统盒子模型

**flex弹性布局**（操作方便，布局简单，移动端应用广泛）

PC端支持IE11以上版本

父元素设置display:flex布局之后,float,clear,vertical-align将失效

父元素叫容器，子元素叫项目

reverse：翻转

flex父项常见属性：

flex-direction: 设置主轴方向，x轴(默认)/y轴

row:从左到右(默认)

ow-reverse:从右到左

column:从上到下

column-reverse:从下到上

justify-content:设置主轴上的子元素的排列方式

flex-start: 默认值，从头开始

flex-end: 从尾部开始

center: 在主轴居中对齐，如果主轴是X轴则水平居中

space-around:平分剩余空间

space-between:先两边贴别，再平分剩余空间（重要）

flex布局中，子元素默认不换行，放不下，就会缩小子元素

flex-warp:设置子元素是否换行

nowrap:不换行(默认)

warp:换行

align-items:设置侧轴上子元素的排列方式（单行使用）

flex-start:从上到下

flex-end: 从下到上

center: 垂直居中

stretch:拉神(默认值，子盒子不要给高度)

align-items:设置侧轴上子元素的排列方式（多行[换行]使用）

flex-start：侧轴头部（默认）

flex-end: 侧轴尾部

center: 侧轴中间

space-around:侧轴平分剩余空间

space-between:先侧轴两头贴别，再平分剩余空间（重要）

stretch:设置子元素高度平分父元素高度

flew-flow:设置主轴方向和是否换行(简写)

column wrap;

flex子项常见属性：

flex：1; 定义子项目分配剩余空间，表示占的份数(默认0),可以写百分号，相对于父元素

align-self:控制子项自己在侧轴上的排列方式

默认auto,继承父元素的align-items,如果没有父元素，等同于stretch

order:定义项目的排列顺序

数值越小，排列越靠前，默认为0，(注意与z-index不一样)

固定定位(fixed)的盒子一定要有宽高，且margin无效

CSS3盒子模型中，行高不等于高度，等于高度-边框

线性颜色渐变，需要加浏览器私有前缀

background: linear-gradient(起始方位, 起始颜色, 结束颜色);

background: -webkit-linear-gradient(left top, red, blue);

**rem配置布局**

文字随屏幕大小变化

高度宽度，等比缩放

em:是父元素的字体大小

rem(root em) 相对单位：html元素的字体大小

跟元素，font-size:12px , 非跟元素设置width:r2em, 换成px就是24px

媒体查询:可以根据不同的尺寸来改变不同的样式

因为层叠行：一半按照从小到大的顺序写

@media mediatype and|not|only (media feature){

CSS-Code;

}

如果 在屏幕上 并且 宽度<=800px ，那么背景色是pink

@media screen and (max-width:800px){

body {

background-color: pink;

}

}

mediatype（媒体类型）: all/print(打印机和打印预览)/scree(电脑，平板，手机)

media feature（媒体特性）：width/mix-width/max-width

引入资源:针对不同的屏幕尺寸，调用不同的css文件

<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="mystylesheet.css">

**less的使用**

less变量 @变量名:值;

以@为前缀

不能以数字开头,不能有特殊符号

less编译

安装easyless 编译插件

hbulider中右击less文件-外部命令-less编译

less嵌套

嵌套不要太多，一般三四层就好

less运算

运算符左右要有空格隔开

如果只有一个数有单位，就以这个单位为准

如果都有单位，以第一个单位为准

**rem适配方案**

根据媒体查询设置html字体大小

页面元素都用rem做单位

方案1

less + 媒体查询 + rem

方案2(推荐) 简洁高效

flexble.js + rem

根据当前屏幕，平均分成了10等份

flexble.js下载地址：https://github.com/amfe/lib-flexible

更改rem大小

工具-设置-源码视图-"editor.codeassist.px2rem.proportion": "16",

固定定位是根据浏览器定位的，和body的属性没关系

摹客插件安装，ps中用摹客切图

swiper插件:https://www.swiper.com.cn/

下载需要的css和js文件

官网找类似案例，复制html结构，css样式，js语法

根据需求定制修改模板

opacity: 0.4;透明度

git可以把我们的本地网站提交到远程仓库（码云gitee）类似之前的ftp

码云是远程仓库，类似服务器

git使用（git bash here）

git init 初始化

git add . 将本地文件放到暂存区

git commit -m '黑马面面网站' 放到本地仓库里去

git remote add origin https://gitee.com/mmwwtt/css\_study.git 链接远程仓库

git push -u origin master 将本地仓库的文件推送到远程仓库 push

发布网站

服务 - gitee pages -启动

会自动生成网址

创建带redme文件的仓库，然后文件-点击上传文件（一个小时只能上传20个文件）

**响应式开发Bootstrap**

可以PC端，移动端，ipad端,都能用，不用分开端来写

需要一个父级作为布局容器，来配合子元素实现变化效果

不同屏幕下，通过媒体查询来改变这个布局容器的大小，在改变子元素的排列方式和大小

超小屏幕（手机，小于768px），宽度设为100%

小屏幕(平板，大于等于768px)宽度设为750px

中等屏幕（桌面显示器，大于等于992px）宽度设置为970px

大屏幕（大桌面显示器,大于等于1200px）宽度设为1170px

响应式快速开发

Bootstrap前端开发框架：http://bootstrap.css88.com/

使用：

1.创建文件夹结构（Bootstarap文件夹），

2.创建html骨架结构

3.引入相关样式文件

4.书写内容

<!--[if lt IE 9]>

解决ie9以下浏览器对html5新增标签的不认识，并导致CSS不起作用的问题

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

解决IE9以下浏览器对css3 Media Query的不识别

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

container：容器

bootstrap需要为页面内容和栅格系统包裹一个.container容器，Bootstarap预先定义好这个类加 .container容器，Bootstarap预先定义好这个类加

container类（响应式布局）

大屏（>=1200px）宽度为1170px .col-lg-

中屏（>=992px）宽度为970px .col-md- 划分12份

小屏（>=768px）宽度为750px .col-sm-

超小屏（100%） .col-xs-

实现列的平均划分，需要给列添加类前缀

每列默认有左右15px的padding

划分不同的分数 .col-lg-3 占3份

响应式工具

.hidden-xs 超小屏隐藏 其余可见

.hidden-sm 小屏隐藏 其余可见

.hidden-md 中屏隐藏 其余可见

.hidden-lg 大屏隐藏 其余可见

.bisible-xs 超小屏可见，其余隐藏

div.demo表示<div class=’demo’></div>

button.disabled = true 禁用按钮