# Css

###### 盒模型

盒模型的的组成：内容 边框(border) 内边距(padding) 外边距(margin)。

###### Flex

弹性布局

###### css的单位

常见的有： % 百分比 px 像素 1em 等于当前的字体尺寸 2em等于当前字体尺寸的两倍。

###### Css的常见选择器

标签选择器 类选择器 ID选择器 ID选择器一般是给js使用

后代选择器 子选择器 并集选择器 序选择器

伪类选择器 的4种状态 a:link,默认状态 a:visited,被访问过

a:hover,鼠标悬停时 a:active,鼠标点击时 一般情况下遵循爱恨原则love hate＝ lvha。

###### BFC清除浮动

1：用伪类选择器 .clearfix:after{ content: ""; clear: both; display: block; height: 0px; visibility: hidden; } .clearfix{ \*zoom:1; }

2：给父元素添加一个:display:inline-block 3：父元素也添加一个浮动

2和3的弊端 无法使用margin

4：给父元素增加高

###### 常见页面布局

1. 固定布局 2.流式布局（百分比布局） 3.弹性布局（伸缩布局）

###### 响应式布局

1. 就是一个网站能够兼容多个终端
2. 过程

* 在html的中添加媒体查询和相关配置
* 构想媒体查询的节点尺寸
* 添加媒体查询
* 提炼css中的公用代码，放在公共的index.css中，直接引入，然后针对不同断点尺寸新建不同的css，

1. css3新特性

创建动画的步骤：给需要执行动画的元素指定动画的名称 创建指定名称的动画 给需要执行动画的元素指定动画的执行时间

animation：动画的名称 动画的执行时间 动画的速度 延迟时间 动画的执行次数 是否执行往返动画; 简写：animation：all 3s。

添加过渡的步骤：布局页面 触发修改属性 给修改属性添加过渡效果

transition：需要过渡的属性 执行过渡的时间 执行过渡的速度 延迟时间,需要过渡的属性 执行过渡的时间 执行过渡的速度 延迟时间. 简写：transition:all 3s

2D旋转：transform 3D转化：在父元素上添加 transform-style: preserve-3d;

###### display哪些取值

none 此元素不会被显示，并且不占据页面空间

inline 行内元素 元素会在一行内显示，超出屏幕宽度自动换行，不能设置宽度和高度，元素的宽度和高度只能是靠元素内的内容撑开。

block 块级元素 会独占一行，如果不设置宽度，其宽度会自动填满父元素的宽度，可以设置宽高，即使设置了宽度，小于父元素的宽度，块级元素也会独占一行。

inline-block 行内块元素 与行内元素一样可以再一行内显示，而且可以设置宽高，可以设置margin和padding。

###### 相邻的两个inline-block节点为什么会出现间隔，该如何解决

1：改变书写方式 2：font-size:0

3： 使用margin设置负值 4：使用word-spacing

###### meta viewport 移动端适配

概念：通常viewport是指视窗、视口，浏览器上用来显示网页的那部分区域

如何设置理想视口代码：

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

###### rem布局的优缺点

rem是相当于html根元素来判断的大小，屏幕变化的时候不是很好控制，用百分比了话，可以根据屏幕的大小，占比多少，相对比较好控制

###### 1像素边框问题

1. 可以用box-shadow 阴影来实现
2. viewport + rem 方案
3. 设置 border-image 方案

# HTML

###### 语义化

就是用合理、正确的标签来展示内容

易于用户阅读，样式丢失的时候能让页面呈现清晰的结构。

方便其他设备解析，如盲人阅读器根据语义渲染网页

有利于开发和维护，语义化更具可读性，代码更好维护，与CSS3关系更和谐。

###### 新标签的特性

用于绘画的 canvas 元素

用于媒介回放的 video 和 audio 元素

对本地离线存储的更好的支持

新的特殊内容元素，比如 article、footer、header、nav、section

新的表单控件，比如 calendar、date、time、email、url、search

.详细语义（兼容性好的标签）www

header:页眉（网页（部分区域）的头部 顶部 导航区域等等）;

footer:页脚（网页（部分区域）的底部|版权区域等等）;

section 标签定义网页中的区域（部分）；

article 内容是引用其他地方的；

aside 跟 article 是一起使用；是辅助 article 区域的内容;

nav 导航链接部分;

###### svg canvas

SVG(Scalable Vector Graphics)可缩放矢量图形,一种用来描述二维矢量图形的XML语言

代码格式如：

<svg width='800' height='600' viewBox='0 0 80 60'>

<rect >

</svg>

Canvas是一个画布,用来在网页上绘制图像

<canvas></canvas>

可以通过网上查找

#### input和textarea的区别

Input中的text是单行文本框，textarea是多行文本框。

###### 5.用一个div模拟textarea的实现

给div添加contenteditable=true即可

1. 移动设备忽略将页面中的数字识别为电话号码的方法

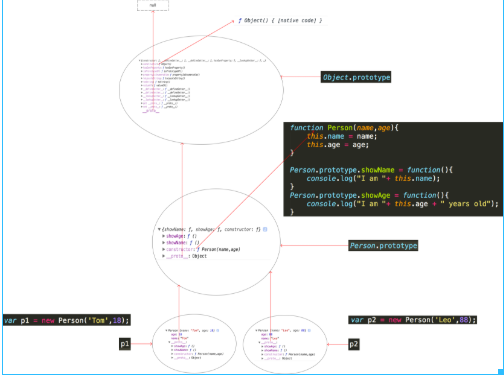
代码如下：<meta name = "format-detection" content = "telephone=no">

# JS

###### 原型/原型链/构造函数/实例/继承

原型：原型就是一个对象，所有的实例共享对象上的方法和属性

原型链：实例对象的\_\_proto\_\_对象-->构造函数的原型对象，构造函数的原型对象的\_\_proto\_\_-->Object的原型对象，Object的原型对象的\_\_proto\_\_ -->null

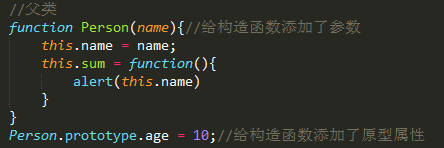


构造函数： 构造函数的首字母必须大写，用来区分于普通函数

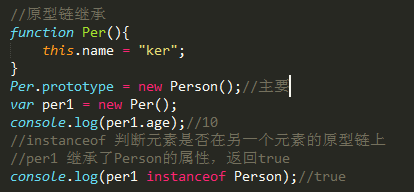
　　　　 内部使用的this对象，来指向即将要生成的实例对象

　　　　 使用New来生成实例对象

继承：　想要继承，就必须要提供个父类（继承谁，提供继承的属性）



**一、原型链继承**



　　　　重点：让新实例的原型等于父类的实例。

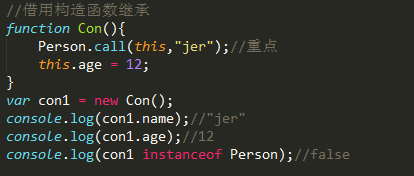
　　　　特点：1、实例可继承的属性有：实例的构造函数的属性，父类构造函数属性，父类原型的属性。（新实例不会继承父类实例的属性！）

　　　　缺点：1、新实例无法向父类构造函数传参。

　　　　　　　2、继承单一。

　　　　　　　3、所有新实例都会共享父类实例的属性。（原型上的属性是共享的，一个实例修改了原型属性，另一个实例的原型属性也会被修改！）

**二、借用构造函数继承**

****

重点：用.call()和.apply()将父类构造函数引入子类函数（在子类函数中做了父类函数的自执行（复制））

　　　　特点：1、只继承了父类构造函数的属性，没有继承父类原型的属性。

　　　　　　　2、解决了原型链继承缺点1、2、3。

　　　　　　　3、可以继承多个构造函数属性（call多个）。

　　　　　　　4、在子实例中可向父实例传参。

　　　　缺点：1、只能继承父类构造函数的属性。

　　　　　　　2、无法实现构造函数的复用。（每次用每次都要重新调用）

　　　　　　　3、每个新实例都有父类构造函数的副本，臃肿。

**三、组合继承（组合原型链继承和借用构造函数继承）（常用）**



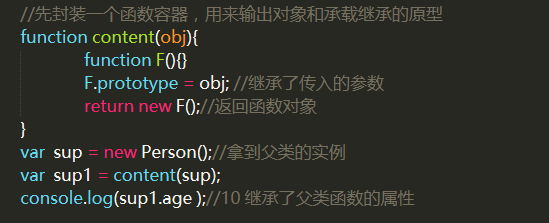
　　　　重点：**结合了两种模式的优点，传参和复用**

　　　　特点：1、可以继承父类原型上的属性，可以传参，可复用。

　　　　　　　2、每个新实例引入的构造函数属性是私有的。

　　　　缺点：调用了两次父类构造函数（耗内存），子类的构造函数会代替原型上的那个父类构造函数。

**四、原型式继承**

****

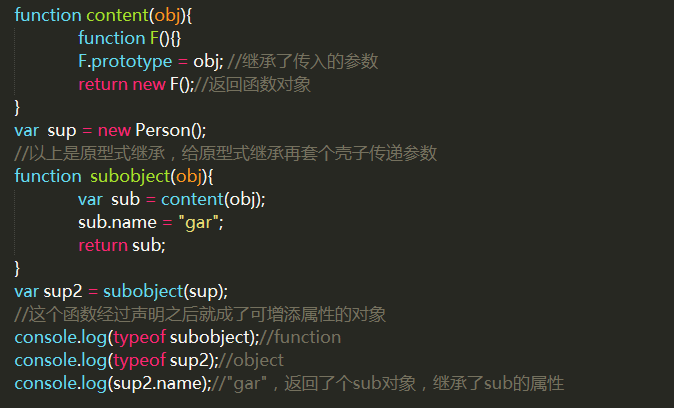
　　重点：用一个函数包装一个对象，然后返回这个函数的调用，这个函数就变成了个可以随意增添属性的实例或对象。object.create()就是这个原理。

　　　　特点：类似于复制一个对象，用函数来包装。

　　　　缺点：1、所有实例都会继承原型上的属性。

　　　　　　　2、无法实现复用。（新实例属性都是后面添加的）

**五、寄生式继承**

****

　　　　重点：就是给原型式继承外面套了个壳子。

　　　　优点：没有创建自定义类型，因为只是套了个壳子返回对象（这个），这个函数顺理成章就成了创建的新对象。

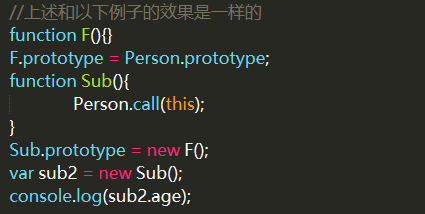
　　　　缺点：没用到原型，无法复用。

**六、寄生组合式继承（常用）**

寄生：在函数内返回对象然后调用

　　　　组合：1、函数的原型等于另一个实例。2、在函数中用apply或者call引入另一个构造函数，可传参





　　　　重点：修复了组合继承的问题

###### Arguments

****arguments****此对象大多用来针对同个方法多处调用并且传递参数个数不一样时进行使用。根据****arguments****的索引来判断执行的方法。

使用方法：

虽然argu对象不是一个数组，但是访问单个参数的方式和数组的方式相同、

###### 数据类型判断

基本数据类型： typeof 可以判断

Undefined 未定义 ：未定义类型只有一个值undefined,这个值的意思是变量没有定义 未定义有两种情况,一个是真的没有定义,一个是定义了但是没有赋值

Null 空对象：空对象类型只有一个值null,代表一个空对象指针

Number 数字类型 ：用来表示数字

String 字符串类型 ：用单引号或者双引号括起来的字符序列

Boolean 布尔类型 ：布尔型也被称为逻辑值类型或者真假值类型

所有类型的值都可以表示布尔值：真:true,非零数字,非空字符,非空对象 假:false,数字零,空字符,null,undefine

Undefined和Null是一种数据类型,而undefined和null是值,是一个常量