

代码笔记总结

1, 行内样式>其他样式表

其他样式表, 优先级一样, 按照导入的顺序来确定他们是否起作用。

2, 选择器的优先级:

类名选择器>标签选择器

行内元素: (不会独占一行, 不能设置宽高,)

font b strong i em s strike del a span

块元素: (独占一行, 可以设置宽高)

h1-h6 p pre hr ul li ol dd dt div

行内块元素: (不会独占一行, 但可以设置宽高)

img input

块元素没有设置宽高时:

子元素的宽度是由他的父元素决定的, 子元素的宽度和父元素的宽度一样

子元素的高度是由他自己的内容决定, 内容会将他自己撑大。

行内元素: margin值只有左右, 没有上下。

行内块元素是识别空格的。

body的margin 上下左右都有

h1-h6 ul ol dl hr p pre 的margin只有上下,且ul ol dd还有padding-left

```
body,p,ul,li,dl,dd,dt,h1,h2,h3,h4,h5,h6,input{
  margin:0;padding:0;
  list-style:none;
  font-family: Arial,"微软雅黑"
}
a{
  text-decoration: none;
  color:#b0b0b0;
}
img{
  border: none;
}
```

overflow:

hidden--超出部分隐藏; scroll--出现滚动条; auto--自动; overflow-x: hidden(水平方向隐藏); overflow-y: hidden(垂直方向隐藏)

display:

block--行内元素转块元素; inline--块元素转行内元素; inline-block--块元素行内元素转行内块元素; none--隐藏

让input隐藏边框: outline:medium;

margin-top的bug:

如果父元素没有设置border或padding值, 子元素用margin-top, 子元素和父元素之间并不会会有间距, 子元素的margin-top会作用到父元素上。

解决方法: 给父元素设置padding-top, 会影响父元素的大小, 减去相应的值。

加了浮动也可以解决

浮动以后行内元素可以设置宽高。

清除浮动的方法:

1, 给父元素添加overflow:hidden; 2,兄弟子元素添加clear:both;

相对定位：是相对它原先的位置进行定位的。

绝对定位：是相对于具有定位属性的父元素来定位的。如果他的父元素没有定位属性，它就相对于浏览器左上角进行定位。

固定定位：

定位和浮动比其他层级都高。定位相对浮动更高。

字体属性具有继承性，

背景颜色：

background-color, background-image:url(), background-repeat:no-repeat,repeat-x,repeat-y; background-position;
background-attachment (设置背景图像的滚动方式 fixed为
固定 scroll为滚动)

(用来解决IE兼容问题)

透明度：1, opacity:0.5

filter:alpha(opacity=70) 值是0-100

2, a img{
border:none;
} a链接里包含img在低版本IE浏览器会有蓝色边框。

3, 一个盒子放img时有空白部分

3种解决方法

a 给img转块元素； b 给父元素加font-size: 0; c 给img加vertical-align:top/bottom

4, 具有浮动的元素，并且设置了margin-left，第一个浮动的元素在IE6中出现双倍边距。

解决办法：

a, 给第一个元素添加，_margin-left: 值为原来的一半；

b, 给第一个元素添加：display: inline;

5,在IE6中不支持png-24的透明图片，虽然支持png-8，但是有锯齿。

6, ie6不支持固定定位，需要用js解决。

7, 子元素的高度超出父元素

在IE中父元素的高度会随着子元素的高度变大而自己的高度变大。

解决方法：给父元素添加：overflow:hidden.

8,行内元素添加浮动属性或定位属性后可以设置宽高，但我们不要为了让行元素设置宽高而用定位，仍是要将它转为块元素：display: inline-block

9,ie6中行元素和浮动元素不能显示在一行。

解决方法：同时添加浮动。浮动后要记住清除浮动

box-shadow:左偏 上偏 模糊度 颜色 (内阴影) (inset)，可以有多个阴影

transition:all 1s ease;

指定边框是否合并

□ border-collapse:separate;

□ 默认样式边框分开

□ border-collapse:collapse

□ 边框合并

固定或自动宽度

□ table-layout:fixed;

□ 固定宽度

□ table-layout:auto

□ 自动宽度

函数中**return**的用法：

- 1，终止函数执行
- 2，返回值

函数的声明方式(创建)

A.基本语法

function 关键字

function 函数名([参数1],[参数2]....){

函数体

[return] //返回值

}

B.字面量定义的形式(匿名函数)

var 变量=function ([参数1],[参数2]....){

函数体

[return] //返回值

}

C.以对象的形式来声明

new 关键字。

var 变量=new Function([参数1],[参数2]...,"函数体");

语言的预解析

- 1，先解析语法的正确性
- 2,<script></script>
- 3， var function

数据类型：

- 1,初始数据类型（简单的数据类型）
string number boolean undefined null
- 2 引用数据类型（复杂的数据类型）
function

/*

*/

难点

```
/*var aa=10;  
function bb(){  
    var aa=5;  
    var aa=20;  
    alert(aa++)  
}  
bb()  
alert(aa)*/
```

/*

*/

全局变量与局部变量的最好例子：

```
var num=10;  
function aa(){  
    // var num=20;  
    function bb(){  
        // var num=30;  
        function cc(){  
            // var num=40;  
            alert(num);  
        }  
        cc();  
    }  
    bb();  
}  
aa();
```

/*

*/

回调函数

```
1, function aa(b){  
    b();  
}  
aa(function (){alert("回调函数")})
```

```
2, function aa(a,b,callback){  
    return callback(a,b)  
}  
console.log(aa(150,25,function(a,b){  
    return a*b;  
})))
```

函数的调用方法（3种）

函数名调用， 自己调用， 事件调用

数组

数组存的是一组数据（必须是关联的）

- 1, var arr=[1,2,3]
- 2, 可以是任意的数据类型
- 3, .length 查看到数组的长度
- 4, []查看数组里面指定的元素，从0开始
- 5, for

对象

一，创建对象

- 1, 用构造函数
function tv(){ }
var sony=new tv();
alert(typeof sony);
- 2, json
var obj={};
- 3, Object方法
var obj=new Object()

二，添加属性和方法

先声明后添加 先添加后声明

三，属性的方法和访问

属性：

obj.属性
obj['属性']

方法：

obj.属性
obj['属性']()

对象的特性之封装

把对象的组成部分组合起来，尽可能的隐藏对象的部分细节，使其受到保护，只留有限的接口和外部发生联系

Math对象：

1.取绝对值

Math.abs();

2.取近似整数

Math.round() 四舍五入

Math.floor() 对数进行向下取整

Math.ceil() 对数进行向上取整

3.取最大值或最小值

Math.max(参数.....)

Math.min(参数.....)

4.取随机数

Math.random();

Math.pow(x,y) 返回 x 的 y 次幂 平方

Math.sqrt(x) 平方根

numberObj.toFixed(2)

指定小数四舍五入后保留的位数

JavaScript字符串对象

一、属性

1.length

计算字符串的长度(不区分中英文)

2.constructor

对象的构造函数

二、方法

□获取类型

1. myString.charAt(num)

返回在指定位置的字符。

2. myString.charCodeAt(num)

返回指定位置的字符的Unicode编码

3. String.fromCharCode()

接受一个或多个指定的Unicode值，然后返回一个或多个字符串

查找类型

1. myString.indexOf("")

返回某个指定的字符串，在字符串中首次出现的位置

2. myString.lastIndexOf()

返回指定的字符串值最后出现的位置

3. myString.match()

在字符串中检索指定的值，返回的值就是指定的 (值)

4. search()

只能作用于正则。

5. myString.replace()

将字符串中的一些字符替换为另外一些字符

截取类型

1.myString.slice(start,end)

从指定的开始位置，到结束位置(不包括)的所有字符串。如果不指定结束位置，则从指定的开始位置，取到结尾slice参数可以是负数，如果是负数，从-1开始指的是字符串结尾。

2.substring(start,end)

从指定的开始位置，到结束位置(不包括)的所有字符串。如果不指定结束位置，则从 指定的开始位置，取到结尾

3.substr(start,length)

从指定的位置开始取指定长度的字符串。如果没有指定长度，从指定开始的位置取到结尾。

转换类型

1.split("分割位置",[指定的长度])

将一个字符串分割成数组

2.toLowerCase();

用于把字符串转换为小写。

3.toUpperCase()

将字符串转换为大写

样式类型

1.fontcolor()

给字符串指定颜色，十六进制表示、red、rgb(255,0,0)

2.fontSize()

指定字符串的大小 (1-7)

练习二

查找字符串 “[优逸客官网地址http://www.sxuek.com](http://www.sxuek.com),[优逸客课程体系：http://www.sxuek.com/uisj/](http://www.sxuek.com/uisj/),[优逸客学生作品：http://www.sxuek.com/sxzp/](http://www.sxuek.com/sxzp/)” 将所有“ sxuek”出现的位置，打印出来

//第一种

/*do{

```

document.write(str.indexOf('sxuek')+'&nbsp;');
document.write(str.lastIndexOf('sxuek')+'&nbsp;');
str=str.replace('sxuek','swqsw');
}while(document.write(str.indexOf('sxuek')));*/
//第二种
/*for(i=0;i<str.length;i++){
    if(str.substr(i,5)=='sxuek'){
        document.write(i+'&nbsp;');
    }
} */
// 第三种
/*var num=0;
while(str.match('sxuek')){
    var pos=str.indexOf('sxuek')
    document.write((pos+num*5)+'<br>');
    num++;
    str=str.slice(0,pos)+str.slice(pos+5);
}*/

```

BOM对象 (browser object model)

window是核心对象

一, window属性

位置类:

距离屏幕的左边距

IE window.screenLeft

W3C window.screenX

距离屏幕的上边距

IE window.scrollTop(到内容) IE 6-8

window.screenY (到内容) IE 9+

w3c window.screenY (到浏览器的边缘)

获取浏览器的尺寸

w3c, ie9+: window.innerWidth window.innerHeight

兼容性方法

document.documentElement.clientWidth

document.documentElement.clientHeight

分辨率: window.screen.width

window.screen.height

window对象

方法:

1.窗体控制

A.对窗体的移动

window.moveBy(x,y) 相对于当前位置沿着X\Y轴**移动指定的像素**,

如负数是反方向

moveTo(x,y) 相对于浏览器的左上角沿着X\Y轴**移动到指定的像素**,

如负数是反方向

B.窗体尺寸的改变

resizeBy(x,y) 相对于当前窗体的大小,调整宽度和高度

resizeTo(x,y) 把窗体调整为指定宽度和高度

2.对窗体滚动条的控制

scrollBy(x,y) 相对于当前滚动条的位置**移动的像素(前提有滚动条)**

scrollTo(x,y) 相对于当前窗口的高度或宽度, **移动到指定的像素**

3.时间间隔的函数

□ **setInterval(函数指针,指定的时间(毫秒))**

按照指定的周期(毫秒)不断的执行函数

clearInterval() 清除时间函数进程

□ **setTimeout(函数指针,指定的时间(毫秒))**

在指定的毫秒数后只执行一次函数

clearTimeout() 清除时间函数进程

*****背景阴影*****

```
box-shadow:0px 5px 5px rgba(0,0,0,0.1);
margin-top:-2px;
transition:all 0.5s ease;
```

*****DOM*****

一， DOM document(html) object model

document对象

一、属性

title 返回或设置当前文档的标题

URL 返回当前文档的url

bgColor 设置文档的背景色

fgColor 设置文档的前景色(设置文字颜色)

二、方法:

1 getElementById(idname)

返回拥有指定id的（第一个）对象的引用

2 getElementsByTagName(tagname)

返回带有指定标签名的对象的集合

3 getElementsByName(name) 【IE浏览器中只识别标准属性，例如：input标签】

返回带有指定name指定名称的对象的集合

4 getElementsByClassName() 【IE6~8不支持】

返回带有指定class指定名称的对象的集合

5 write()

节点

一，获得节点的关系属性

obj.parentNode 获得父元素

obj.childNodes 获得子元素的集合【IE6-8 正常；IE9-w3c 识别标签之间的空格】

父元素和子元素都没有定位

offsetLeft=父元素的左外边距+父元素左边框宽度+父元素左内边距+子元素的左外边距

父元素没有定位，子元素有定位属性

offsetLeft=left+margin-left

父元素定位，子元素没有定位属性

offsetLeft=父元素padding-left+自身margin-left

父元素子元素都有定位属性

offsetLeft=自身left+margin-left

获取焦点：focus

失去焦点：onblur

Juquery 两个作用：隐式循环，链式调用

传址，传值问题：浅拷贝 深拷贝

*****Javascript重点*****

1.原型（被继承的对象就是对象的原型）

2.继承 在Javascript中一个对象会拥有另一对象所有的属性和方法，这个特性叫做继承

构造函数

this的用法：a，普通函数中this指的是window对象 b构造函数中this指的是实例化对象的原型 c，

3，闭包（在函数外部访问函数内部的变量）

4, 回调函数 (把一个函数作为另一个函数的参数使用)

5, 事件流 (事件触发的流程,分为:冒泡型和捕获型事件)

6, 事件委派

forEach map filter some every

box-sizing:border-box;

*****整站引入页面*****

category 栏目页 list 列表页 show 内容页

1, 导航栏{ template "content","header_my" }

2, 列表栏: { pc:content action="category" catid="0"}{/pc} 获取内容

action:调用哪个模块 ("category" "lists" "position")

catid: 调用哪个栏目下的内容 一级栏目:(0)

num:最多取多少条数据

siteid :\$siteid 全局变量

order: 排序方式 listorder ASC (升序) listorder DESC(降序)

return: [data] 返回值得名字

3, { loop \$data \$v} {/loop }

输出方式: <?php var_dump(\$v)?>

链接方式:

{ \$v[catname] }

4, catid="\$v[catid]"

5, catid="\$catid" 当前页面

6, \$top_parentid 当前页面的顶层父栏目的ID

7, siteurl(\$siteid) 首页的连接地址

8, { \$CATEGORYS[9][catname] } 获取栏目9的栏目名称

{ \$CATEGORYS[9][url] } 获取栏目9的栏目地址

{ \$CATEGORYS[9][image] } 获取栏目9的栏目图片

{ \$CATEGORYS[9]s[description] } 获取栏目9的栏目描述

9, <div class="bread">

您现在的位置:

首页

>{\$CATEGORYS[\$top_parentid][catname]}

>{\$catname}

{catpos(\$top_parentid)}成功案例

</div> 当前页面面包屑导航

10,src="{ \$v[thumb] }" 缩略图

{thumb(\$v[thumb],100,100)} 裁剪缩略图

{str_cut(\$v[title],28,...)} 截取文字

{data("Y-m-d H-m-s",\$v[inputtime])} <!-- 获得发布内容的时间 -->

11,要获取副表中的内容(比如文章的内容) 需要在pc标签中添加 moreinfo="1"

12 翻页效果 在pc中添加 page="\$page" 在需要翻页效果的地方 {\$pages}

13, 模型管理--文章模型--字段管理--添加字段 (不推荐)

模型管理--添加模型--添加字段 (推荐使用)

14, 推荐位

内容 --> 内容相关设置 --> 推荐位管理 --> 添加推荐位-->在内容中推荐 (字段设置在推荐位中可以调用)

连接地址 : sqllduapp.com

Access Key

已创建1个，共可建20个

Access Key ID	Secret Access Key	说明	创建时间	操作
deb61b55474342fc9a a62f28f44b916f	3281cc0e5b52477d84a5 75ce7afc6216隐藏	-	2016-05-04 09:04:27	删除

隐藏最后一个边框：

th:last-of-type,td:last-of-type{border-right:0}

jQuery阻止浏览器默认行为：

```
$(".add").mousedown(function(e){
    e.preventDefault()
    e.returnValue=false; (ie6-8)
})
```

鼠标右击事件：contextmenu \$(document).on("contextmenu")

//hcb211

HoYislgsQNYSCHGJBHyQ

已启动

MySQL数据库

*BAE基础版扩展服务中的MySQL只能被BAE基础版应用访问，无法通过公网、BAE专业版或本地开发环境访问。

基本信息

数据库名称：HoYislgsQNYSCHGJBHyQ

连接地址：sqllduapp.com

端口：4050

用户名/密码：AK/SK

类型：共享

到期时间：永久免费

数据库描述：

用户权限：insert delete update select create drop index alter

默认字符编码：utf-8

运行状态信息

Access Key ID	Secret Access Key	说明	创建时间	操作
deb61b55474342fc9aa62f2 8f44b916f	3281cc0e5b52477d84a575ce7af c 6216隐藏	-	2016-05-04 09:04 :27	删除

http://cn.uptodown.com

http://acaa.adsk-certification.cn/

<http://www.uedsc.com/fullpage.html> 全屏轮播插件

www.dowebok.com

响应式布局

min-device-width 设备宽度
background-position:center

让

```
父容器 : {  
    line-height;  
    text-align:justify;  
父容器 : after{  
    conernt:""  
    display:inline-block;  
    width:100%;  
}  
a:{display:inline-block}
```

个人主页:

首页

自我介绍

作品展示

技能介绍

联系方式

cursor: default; (浏览器默认样式)
user-select: none; (阻止用户复制文字)

```
-webkit-user-select: none;  
-moz-user-select: none;  
-ms-user-select: none;  
user-select: none;
```

```
<input type="text" class="middlesearch" placeholder="搜索 apple.com">
```

```
.clearfix:after{  
content:"";  
display:block;  
clear:both;  
}
```

响应式的栅格化布局：

把一个完整的网页划分成不同的栅格，把栅格作为网页布局的单位 12 16 24 36 栅格

Less、Sass

320-480px 移动端
small: 750px; mid:975px; lg:1180px;

-- 移动端 --

像素：一个像素就是计算机屏幕能显示一种特定颜色的最小区域

设备像素：设备屏幕的物理像素，任何设备的物理像素都是固定的

逻辑像素：真正用来展示内容的像素数和识别像素成比例，这个比例称之为DPR

iPhone6/6s：1334*750像素 分辨率667*375 dpr 2

华为：1280*720像素 分辨率640*360 dpr 2

三星：2560*1440像素 分辨率853*480 dpr 3 每3个逻辑像素对应1个物理像素

布局视口：

在小屏幕设备上如果视口的宽度和浏览器的宽度一样会很丑陋，

视觉视口：

理想视口： meta标签用来定义理想视口，与设备无关 320 340 480

<meta name="viewport" content="width=device-width initial-scale=1">

这一代码告诉我们布局视口的宽度应该与理想视口一致，所谓理想视口就是我们所说的分辨率。

缩放：从技术上说缩放是一个放大和缩小长沙市像素的过程，放大会让一个css像素跨越更多的

禁止缩放：<meta name="viewport" content="user-scalable=no">

meta width 设置布局视口的宽度为特定的值

initial-scale 布局视口与视觉视口的缩放比例

移动端事件：touchstart touchmove touchend ---触摸事件

var div=document.querySelector(".box") --/*重点*/

zepto.js touch.js swiper.js

tap singleTap doubleTap longTap swipe swipeRight swipeLeft swipeUp

width:60%;padding-top:50%;

position:fixed;left:20%;top:10%;

z-index:999;

div.notice>div.content>div.top{是否删除}+div.bottom{是/否}>span*2

点击{yes/no}

var target;---（点击的目标）

480*614

移动端

1, 移动端概念

2, 移动端特点

- 屏幕分辨率小,
- 用户操作不同,
- 兼容性问题（分辨率，css，js）

3,移动端的布局

<meta name="viewport" content="width=device-width initial-scale=1 minimum-scale=1 maximum-scale=1 ">

视口的概念：布局视口，视觉视口，理想视口

4,移动端的事件

--原生事件：touchstart,touchmove,touchend;

--zepto.js(移动端jquery库,

移动端事件：tap singleTap doubleTap longTap swipe swipeRight

swipeLeft swipeUp)

--touch.js(移动端事件库源于百度)--touch.on(ele,event,function()){}

tap,hold,doubletap,totate,pinch,drag,dragend,swipeleft,swiperight

swiper(全屏滚动，轮播，功能齐全的移动端插件，它提供给我们针对移动端轮播效果的大量API)

5, 移动端布局

- 固定宽度布局
- 百分比布局 页面比较简单
- rem布局 更加的灵活方便，使用所有宽高，基于字体单位rem

响应式布局和less

1, 响应式布局概念

基于css3媒体查询技术（media query）技术编写css代码，通过合理的安排html的结构结合css能够实现一套网页在不同的分辨率情况下有不同的展示。从而达到更好的用户体验的效果。

2, @media screen and (min-width/max-width){}

3, less 可以编译css的一套程序化语言。

-- node.js npm 安装less

结合编辑器file watch功能实现实时编译

less语法: 变量-混合模式-嵌套-引导-函数

4, bootstrap 一套成熟完整的css框架 提供了定制的功能

核心-->栅格化 组件 插件

fullpage

全屏滚动插件

响应式 结合css3动画 css3过渡 css3转换等效果

jQuery.transition.js 插件 animate操作css3/2 属性transition

css3 (新属性)

选择器, 圆角阴影, 边框背景, 渐变倒影, 文本, 字体, 字体图标, 2d 3d转换, 动画, animation

网页: 百度舆情(全屏滚动 css3动画 响应式)

苹果首页(响应式设计的实现 导航效果的制作)

网易邮箱(全屏滚动, jquery.transition.js插件)

沙发网(栅格化布局的应用)

苏宁移动端(rem布局 swiper的使用)

个人主页(全部)

小效果: 通知栏的效果(zepto.js 百分比布局)

拖拽轮播图效果(touch.js)

response.less...(对于less的应用)

3d轮播图 分层 六棱柱形式(css3 3d的转换)

进度条的效果(css3动画的应用)

iconfont首页的效果(css3动画以及less应用)

时钟的效果(css3 2d转化的应用)

下雨的效果(css3 过渡的应用)

五环的效果(css3 渐变的应用)

// 圆角 阴影 选择器

```
localStorage.message=JSON.stringify(arr);
```

```
location.href="login.html";
```

```
if(local)
```

原生的拖拽事件,

面向对象:

```
function drag(obj){
```

```
  this.obj=obj;
```

```
  this.ox=0;this.oy=0;this.cx=0;this.cy=0;this.left=0;this.top=0;
```

```
  this.mousedown();
```

```
}
```

```
drag.prototype={ //(构造函数的对象)
```

```
  _mousedown:function(){
```

```
    var that=this;
```

```
    this.obj.onmousedown=function(e){
```

```
      var ev=e||e.event;
```

```
      that.ox=ev.offsetX;
```

```
      that.oy=ev.offsetY;
```

```
      that._mousemove();
```

```

        that._mousedown();
    }
},
_mousemove:function(){
    var that=this;
    document.onmousemove=function(e){
        var ev=e||e.event;
        that.cx=ev.clientX;
        that.cy=
        that.left=that.cx-that.ox;
        that.top=that.cy-that.oy;
        that.obj.style.left=that.left+"px";
        that.obj.style.top=that.top+"px";
    }
},
_mouseup:function(){
    var that=this;
    that.obj.onmouseup=function(e){
        document.onmousemove=null;
    }
}
};
new drag(item);

```

轮播：双下标 切换层级 左右滑动 节点操作

rem：基于根元素体大小的单位

*****HTML5*****

---- 前端设计师：平面设计师--网页设计师。
 ---- 2007年，iPhone4的诞生--划时代。
 ---- HTML5，css3只为移动端而生。
 ---- open-gl web-gl 是css3为什么可以处理图形图像的根源。
 HTML5分为：广义和狭义（所引发的一系列技术）

ecmascript-->6.0 偶数版本比较稳定--基数不稳定

dom----bom

flash--->耗费资源 -->actionscript语法为了操作绘制的图形

*****canvas*****

canvas：宽高不能到css里设置，在属性中设置

createPattern("img","no-repeat")----给画布填充图片

beginPath(); /* 重点 */

cobj.moveTo(0,0);

cobj.lineTo();

cobj.closePath();

cobj.lineWidth=10; //不加单位

cobj.strokeStyle="red";

cobj.drawImage(img,0,0,100,100,0,0,100,100);

截取一张图片任意区域，将其呈现在画布中的任意大小的任意位置

位图放大会失真，矢量图简单说可以无限放大而且不失真。

擦除: `cobj.clearRect(0,0,600,300);`

鼠标事件, 键盘事件, 浏览器自带事件 (`window.onload`)

*****js里函数的功能*****

1, (普通函数) 将特定代码块放在一起重新调用。

2. 模拟类 --> 构造函数

3, 当对象使用。

```
function aa(){
  alert("bb1");
}
aa();
aa.bb=function(){
  alert("bb2")
}
aa.bb();
aa.prototype.bb=function(){
  alert("bb3")
}
new aa().bb();
```

面向对象是基于过程的, 有了类就使得我们认识对象更加简单。

对象: 1,json--->更多的是存储数据。 `var obj={name:"zangsan",age:12};`

2,通过构造函数 `function aa(){`

```
  this.name="zhangsan";
  this.age=12;
  this.sex="man";
  this.say=function(){
    alert("say")
  }
}
```

`new aa();`--->做

```
function aa(){
  alert(this) -->this指向window
}
aa.call("a") --->由window转向a
```

初始数据类型: 数值, 字符串, `undefined`, `null`, `boolean`

引用数据类型: 数组, 函数, 对象

原型的最大功能就是能极大节省对象所占的内存;

svn,http,ftp,file,https协议: 每一种协议规定数据传输的一种规范。

如何压缩网页代码?

---压缩 (去空格)

---图片精灵

---图片的优化 (JPEG 有损压缩的图片格式, 不支持透明通道【大图】),

GIF 有损压缩的支持透明通道,

png 无, 【常用小图】)

---ajax【异步加载】

v8引擎 `crom`,

```
function isString(val){
    return typeof val=="String"
}
```

碰撞：规则碰撞，不规则碰撞

<https://www.douban.com/note/275937697/>

Ajax

http协议

基于浏览器能与本地浏览速度一样。

JavaScript是单线程'异步机制'的一种语言

异步函数：setInterval,setTimeout, 事件

alert(1)

```
var div=document.querySelector("div");
```

```
div.onclick=fucntion ( ) {
```

```
    alert(2)
```

```
}
```

```
alert(3)
```

XMLHttpRequest();--->异步加载的对象。

创建ajax对象

1. new XMLHttpRequest();

2.ie6下

```
new ActiveXObject("Microsoft.XMLHTTP");
```

得到请求对象之后就可以进入请求/响应循环了。

记住，XMLHttpRequest 惟一的目的是让您发送请求和接收响应

Ajax是异步的，创建个对象是为了通过这个对象获取单线程结束后异步读取信息。

```
[2] obj.onreadystatechange=function(){
```

```
    if(obj.readyState==4){
```

```
        if(obj.status==200){
```

```
            obj.responseText ---->从服务端返回的数据
```

```
        }
```

```
    }
```

```
}
```

```
[3] open("get","demo2.html")--->打开方式
```

```
[4] send();
```

wamp:[window-apache-mysql-php](#)

apache:解析PHP语言--

mysql: 数据库---php才能操作数据库

PHP: -->

数组：索引数组(下标索引为数字)，

```
$arr=Array(
```

```
)
```

```
echo json_encode($arr);
```

```
eval("alert(1)")--->
```

关联数组(下标索引为字母);

XML 跨平台的一种语言，它不是个程序语言，没有任何标签，**xml**没有定义任何标签（保存数据）；
html是**xml**派生出的语言，有很多标签（呈现内容）

*****数据库*****

database 数据库

字符集，前端-**PHP**-数据库---->呈现出的字符集必须是相同的；

常见的几种；

GB2312--

utf-8--几乎囊括世界所有字节，拓展性更好

GBK--

MySQL---从广义上说也是数据库

数据库如何存储数据？ --

一个个的库-->表格-->字段(**name,age,sex....**)

cd---进入某个文件

dir--查看某个文件

mysql -h地址 -uroot -p 进入mysql

show databases;查看数据库

use mydemo 选择数据库

show tables;查看所有表

desc stu; //查询表结构

select name from stu; 查询表格

insert into stu (name,age,sex) values ("张三","13","男"); 增加信息

update stu set name="王五" where id=3; /*不常用*/

update stu set name="王五" where id=3 where id=3; 修改信息

delete from stu where id=3; 删除信息

drop table stu; 删除表

show tables; 查看库

drop database mydemo; 删除指定库

create database mydemo 回车 ;

create table demo(

id int(11) primary key(主键) auto_increment(自增),

name varchar(255),

age varchar(255),

sex varchar(255));

show tables;

<?php

header("Content-type:text/html;charset=utf-8")

\$db=new mysqli('localhost','root','root','message');

\$db->query("set names utf8");

var_dump(\$db); //

\$title=\$_GET["title"];

...

\$sql="insert into message (title,con) value ('{title}','{con}')";

\$db->query(\$sql);

if(\$db->affected_rows>0){

echo "<script>alert('插入成功');location.href='../message.html'</script>"


```
include "../notice.html";
}
?>
```

<http://cluba11.yignew.com:8077/index.html>

angularjs---

后台-可视化，逻辑清晰的结构化界面

再次熟悉ajax，了解后台，了解phpcms

```
includ;"db.php";
```

```
$arr=array();
```

```
if($row=$result->fetch_assoc()){    result->fetch_assoc()
```

```
    $arr[]=$row;
```

```
}
```

```
echo json_encode
```

添加: `input.setAttribute("type","text");`

```
input.value=oldv;
```

```
input.focus();
```

```
blur(){
```

```
    getAttribute("attr")
```

```
}
```

```
mvc phpcms limit $page,$number;
```

```
$page=isset($_GET["page"])?$_GET["page"]:0;
```

```
$pages=$results->num_rows;
```

```
echo $nums;
```

ajax 按需时加载,可能导致页面内存不能释放

阻止 a链接 / submit 不让跳转: 1 , JavaScript: ; 2 , e.preventDefault() 3,return false;

jquery中，[type]这种形式的话为可传可不传。

```
function infofun(info,mask,message,type){
```

```
}
```

```
function aa(){
```

```
var num=0;
```

```
alert(num);
```

```
return function(){
```

```
num++;
```

```
alert(num);
```

```
}
```

```
}
```

```
aa();    // 0
```

```
aa()();  // 0 1
```

```
aa()();  // 0 1
```

```
aa();    // 0
```

```
var fun=aa();  // 0    闭包
```

```
fun();  // 1
```

```
fun();  // 2
```

```
aa(); // 0
```

MVC

数据又称作模型，MVC--model view controller;

后台：

大量的phpcms：ecshop dede discuz smarty(PHP写出的模版引擎)

admin--template-index.html

index--

template--

static--css,js,img

index.php--入口

PHP中定义常量 `define("");`

`echo "<pre>"`

```
var_dump($_SERVER)
```

```
define ("PATH",$_SERVER[""])
```

```
echo substr()
```

```
define("WEB_PATH","http://")
```

ECMAScript 6 里有常量的介绍 `const b=10;` 常量不可以改变

块级变量

```
if(true){
```

```
let a=10;
```

```
}
```

"use strict" 严格定义变量

```
aa=10; alert(aa)
```

```
<?php
// m v c 架构
// model(数据类似数据库) view(视图类似html+css) control(类似PHP)
// 基于bs架构
// 前端语言仅剩JavaScript
// 后台语言太多 php/.net mysql
?>
```

HTTP协议是即时的，无状态的，无记录的。`cookies`唯一的目的是让各个页面之间共享数据。声明一个变量 `var a=10;` 在另一个页面是找不到的，因为他们所对应得window是不同的。

(`window.a=10`)

```
unset($_SESSION["login"]);
```

```
unset($_SESSION["aid"]);
```

```
unset($_SESSION["user"]);
```

```
$message="退出成功"
```

*****Nodejs*****

Flash自带脚本语言：`actionscript`。移动端的出现导致Flash不得不改版，为了适应手机等移动端，但并没有成功。再者它不是开源的。同时谷歌，苹果一起研发了安卓系统，推进了系统的升级换代。。。

JavaScript的局限性导致后台script也就是Nodejs的出现。

Node只是代码准备好了，所有环境以及作用域都得代码实现。

Node.js 是一个基于 **Chrome V8** 引擎的 JavaScript 运行环境。Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效。Node.js 的包管理器 **npm**，是全球最大的开源库生态系统

包 模块 ---->Nodejs 两个重要概念。使得我们写代码更加条理。

包-->一堆文件（文件夹）； 模块-->一个小的功能，一个文件，我们需要很多功能

http是一个无状态的协议。创建一个http，但是为了安全考虑不要创建1024下的端口

uri 从广义上就是一个资源，其实就是URL

URL 是查找资源的方式

nodejs--天然就是MVC架构的。

jade, ejs-->nodejs就可以解析模板标签

gulp --dist+ --src(css,js)+

```
gulp.task("demo",function(){
  gulp.src("src/js/*.js").pipe(gulp.dest)
})
gulp.task('default',['demo']);
```

*******Angularjs*******

双向数据绑定 m（数据） v（视图） c（控制器）

指令

```
<body ng-app>
<!-- m v c -->
<input type="text" value="" ng-model="name"> // m
<div ng-bind="name"></div> // c
<div ng-bind="name"></div>
<div ng-bind="name"></div>
<div ng-bind="name"></div>
```

后台的MVC架构

- 1, 地址
- 2, 服务器->apache
- 3, apache->相应的文件
- 4, apache->数据
- 5, apache->视图
- 6, apache->数据+视图
- 7, 返回浏览器

前端的MVC架构

- 1, 地址
- 2, apache->接收请求
- 3, apache->取数据
- 4, apache->返回数据ok
- 5, 浏览器接收数据
- 6, 浏览器指定什么样的数据放到什么样的视图里面
- 7, 装配

前端的MVC趋向angular

angular本质上跟jQuery类似，jQuery【查询问题，兼容性问题】--->由很多模块组成，模块划分功能。

angularjs是为了解决前端编程思想的。

angular.module("myapp",[])---->启动模块，两个参数，一个是启动angular，另一个是它所依赖的模块，如果一个参数的话是调用模块。

控制器决定作用范围

双向数据绑定-->视图，数据 相关联 mv vm 只有进行了指令的特殊处理才能进行双向数据绑定
脏值检测：

a b 脏==b 定义好的处理方式

```
// 服务 $filter("json"),(temp))  
    $filter("currency")(100,"¥")  
    yy-MM-dd hh:mm:ss
```


