

## 1. 数据类型:

1.1 基本数据 number, string, boolean, undefined, null;

1.2 对象: object

1.3 函数: function 特殊值, nan

javascript 是弱类型的, 变量当中能够存放任意数据类型。

当我们声明一个变量 (var a (局部变量) 不用var, 是全局), 而未赋值的情况下, 改变量是undefined类型, 改变量有唯一的值, 改值就是undefined, (未声明变量就使用是错误的)

null类型, 有唯一的值null。有运算符typeof返回变量的数据类型, 一个对象为空null, 返回的是object类型。如果说你想为一个变量存放object类型, 在还没有赋值之前, 可以显示初始化为null. 以表示, 改变量是对象类型, 由于undefined派生自null, 所以undefined==null是为true。null===undefined时为false (全等符)

number: 无论整数, 小数都是number类型。根据实际结果, 自行决定, 解释执行, 解释一句执行一句。赋值可以0开头八进制, 十进制, 0x开头十六进制, 科学计数法。

类型的转换 : 字符串转换为number : Number( '122' )----->122。

:

parseInt( '123a' )----->123。

:

parseInt( '123a' , 16)----->4666。16进制转换

;parseInt ( '10' , 8) ---->8。8进制

parseFloat ( '123.33A' ) ----->123.33

parseFloat( '123.33.123' ) ----->123.33 (解析到第二个小数点之前)

undefined+10 (隐式转换)

隐式转换:

数据类型	值	隐式转换后的值
undefined	undefined	NaN

null	null	0
boolean	true	1
	false	0
string	number函数可转换的值	就是number转换后的值
	不是number可转换的值	NaN

转换函数Number（），解析函数：parseInt（），parseFloat（）

NaN是number中的特殊值，not a number 不是一个数。

Boolean类型，true和false，向boolean类型的转换。转换函数：

Boolean（）

类型	值	转换后的值
undefined	undefined	false
null	null	false
string	非空字符串为true、	true
	空字符串转换为false	false
number	0	false
	非0	true
object	非null对象	true
	null对象	false

string类型：字符串类型，可以使用String 包装类的所有属性和方法。

变量的声明：

变量的声明有两种形式，可以使用或不使用var来声明变量：

var a;           //var声明一个局部变量

a=90;           //声明一个全局变量，要和赋值一起使用。

funcation ：是定义一个方法，

funcation 函数名（参数列表）{函数体}

javascript中函数也是对象。有自己的属性和方法。

arguments是函数内部的一个属性。是类数组对象。例子，

arguments 是函数的内部的一个属性，

函数内部的作用域不是语句块级别的。当我们在执行函数时，它的运行环境创建了一个作用域对象，我们所声明变量，都是这个作用域对象的属性，该对象在没有引用的情况下销毁。默认情况下，在函数执行完后销毁。

```
function a(){  
  var c= 90;  
  return function () {  
    return c;  
  }  
}
```

闭包，访问一个函数在执行过程中的局部变量，实现的机制是通过局部作用域对象的访问链实现了，  
由于返回了内部的函数对象，该函数对象的局部作用域对象未被销毁，从而可以访问对在该作用域对象访问链上的局部变量。

object类型：是一切类的超类，父类。

创建对象的方式，

object类的创建。添加属性和方法

```
var people= new Object();  
people.name='wang';  
people.age=20;  
people.gender='male';  
  
people.work=function () {  
  document.writeln(this.name+"working.....")  
}
```

---

构造方法创建对象：以新创建的对象为当前对象。

```
function People(name,age,gender) {  
  this.name=name;  
  this.age=age;      //局部变量
```

```

    this.gender=gender;
    this.getmsg = function () {
    return msg;
}
    //ding定义原型方法
People.prototype.work = function () {
    document.writeln(this.name + "working.....<br>")
}
}var o=new Object();
People.call(o,'wang','90','男');
document.writeln(o);
var p= new People('xie','80','女')

```

通过构造创建对象，1. 通过new关键字创建了一个新的对象；2，以新创建的对象为当前对象。将this指向这个对象，调用构造方法。

### 1.3.2 prototype

prototype 是函数（构造函数）对象的一个属性，类型就是prototype，该属性指向一个对象，我们可以在该对象上添加属性和方法，当通过该构造创建的对象访问一个方法或属性时，先从该对象自身上查找相应方法或属性，如没有找到，将从该构造的prototype对象上进行查找。prototype上定义的属性和方法，是该类型的对象所共有的。

## 2， 运算符

++, --, \*, /, %, +, -,  
 ~, &, |, ^, <<, >>, >>>  
 &&, ||, !

<, >, ==, <=, >=, !=, , , , , === （全等）， != （不全等）

conditions ? a: b

=, +=, -=, \*=, /=, %=

typeof, instanceof,

## 3， 数组

js中数组也是个对象，js中的数组是可变长数组，java是定长。其实我们所添加的数组元素，都是该对象的属性（索引属性），

使用Array（）构造函数创建数组对象。

参数	示例	说明

一个number	new Array (3)	创建一个3个长的数组
无参	new Array ()	创建0个长的数组
有多个参数	new Array (2, 3, 4, 5)	按参数静态初始化, 创建4个长度的数组

#### 4, 流程控制

if-else,, switch (无类型限制), for, while等

#### 5, js常用函数

window对象中常用的方法。设置计时器, 从当前时间流逝的毫秒数, 当超时  
时, 执行函数。setTimeout(function, 毫秒); clearTimeout

var \_id = setInterval (fun, milliseconds); 设置时间每间隔多少  
window.eval (string), 该方法类似一个js的编译器, 可以将参数进行运算处理。

内置对象

jion

date

string

数组

#### 6, 正则表达式

7, 事件, onclick, onblur, onchange, onkeydown, onload (页面加载完了以后  
触发), onsubmit

#### 8, DOM

DOM是文档对象模型 (Document object model), 将文档看做一个树形的数据模  
型。对文档的操作, 就是对这个树的操作。

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
</head>
<body>
hello </body>
</html>
```

以上，  
获取节点

序号	方法	示例
	document.getElementById 根据这个元素的ID获取这个元素	document.getElementById ( 'mobile' ) ;
	document.getElementsByTagName('tag'); 根据HTML标记名获取一组元素，返回的是一个类数组的NodeList类型。	document.getElementsByTagName('input'); 实现对checkboxbox的全选功能。
	document.getElementsByName('name'); 根据html的name属性，获取一组元素，返回依旧是NodeList类型	http://localhost:63342/xiaodiandian/lianxi/taobao.html
	document.getElementsByClassName ('classname');	
	document.querySelectorAll('input'); 根据选择符返回多个NodeList节点。	
	document.querySelector('input');	

创建元素/节点

创建属性

创建文本元素

添加元素节点

删除元素节点

BOM操作

浏览器对象模型，ECMA Script中是没有标准的，是各浏览器厂家自定义的，如果大家定义有相同，则以事实上的标准。BOM为对浏览器的操作。

打开，关闭浏览器的窗口：

```
window.open ( 'url' , windowname, param ) ;  
window.close ( ) ;
```

打开alert窗口:

打开警告框。仅有提示作用。

示例

打开confirm窗口

确认对话框，有返回值，确认返回true，否则返回false

打开prompt窗口

可接受用户输入的窗口，确认输入内容返回，输入的字符串，否则返回false

设置浏览器url

location对象，其实就是window的对象的一个属性，用来访问url，设置url

location.host: 返回主机名

location.hostname: 返回主机名和端口号

location.pathname: 返回路径

location.port: 使用的端口

location.protocol: 所使用的协议

设置浏览器历史记录:

history对象，history.go ( N ) 在历史记录中前进或后退N步，N为负值进后退，为正值就前进。

history.back () ==history.go (-1) ; history.forward=history.go (1)

history.replace () 设置新的urlbing并替换当前的历史记录，从而不产生新的history记录。

6

jdbc

html

js

