



传智播客 · 前端与移动开发学院

<http://web.itcast.cn>

前端基本功—javascript 第二天

目录

目录.....	2
1.2 复习.....	4
1.3 函数(function).....	5
1.3.1 函数的声明.....	5
1.3.2 自定义函数.....	6
1.3.3 函数直接量声明.....	6
1.3.4 利用 Function 关键字声明.....	6
1.3.5 变量声明提升.....	6
1.3.6 函数参数.....	8
1.4 案例 鼠标展示.....	10
1.5 返回值 return.....	11
1.6 算术运算符.....	12
1.7 条件语句(if).....	12
1.8 获得焦点 失去焦点 事件.....	13
1.9 this (自己的).....	14
1.10 属性和方法.....	14
1.10.1 表单自动获得焦点.....	15
1.10.2 鼠标经过选择表单.....	16
1.11 For 循环.....	16

1.12 getElementsByTagName() 获取某类标签..... 16

1.1 复习

一、提示框

1. 弹出警示框 `alert();` `window.alert();` `window` 窗口

2. 控制台输出 `console.log()`

3. 文档打印 `document` 文档 `document.write();`

1. 下面哪种 JavaScript 语法格式是正确的 ()

A. `echo "I enjoy JavaScript";`

B. `document.write(I enjoy JavaScript)`

C. `response.write("I enjoy JavaScript ")` ;

D. `alert("I enjoy JavaScript ");`

二、变量

Java `Var aa:int = 10;`

`Var num = 10;`

字母 `_` `$` 开头

`Var dd=100,ee=100` 声明多个变量 逗号隔开

1. Javascript 中, 以下声明变量语句中哪个不正确?

A、`var aa;`

B、`var bb=3; cc='good';`

C、`var dd = ee = 100;`

D、`var ff=3, gg='he's good';`

三、事件三要素

事件源 被触发的对象 名词

事件 动词 onclick onmouseover onmouseout

事件处理程序 放到函数里面就行了

事件源.事件 = function(){ 语句; }

入口函数: window.onload = function(){ }

四、

行内式

<div onclick="alert(11)">

内嵌式

外链式

<script type="text/javascript" src="xx.js"></script>

五、数据类型

Number string boolean null undefined

1.2 函数(function)

函数： 函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块。

四大发明 造纸术 火药 指南针 印刷术

1.2.1 函数的声明

函数使用跟变量一样，需要 声明

1.2.2 自定义函数

```
function fun(){  
    alert("我是自定义函数")  
}  
fun(); // 函数不调用，自己不执行
```

1.2.3 函数直接量声明

```
var fun1 = function(){  
    alert("直接量声明")  
}  
fun1(); 也需要调用
```

1.2.4 利用 Function 关键字声明

```
var fun2 = new Function("var a = 10; var b = 20; alert(a+b)");  
fun2();
```

1.2.5 变量声明提升

什么是变量提升

```
1  function fun(){
2      console.log(num);
3      var num = 20;
4  }
```

相当于 ---

```
5  function fun(){
6      var num;
7      console.log(num);
8      Num = 20;
9  }
```

在函数体内部，声明变量，会把该声明提升到函数体的最顶端。只提升变量声明，不赋值。

```
var a = 18;
f1();
function f1(){
    var b=9;
    console.log(a);
    console.log(b);
    var a = '123';
}
```

结果是： undefined 9

1.2.6 函数参数

```
7 <body>
8 <script>
9     function fn(a,b)
10    {
11        console.log(a+b);
12    }
13    fn(1,2);
14 </script>
15 </body>
16 </html>
```

形参

实参

arguments 是存储了函数传送过来实参

Javascript 在创建函数的同时，会在函数内部创建一个 arguments 对象实例。

arguments 对象只有函数开始时才可用。函数的 arguments 对象并不是一个数组，访问单个参数的方式与访问数组元素的方式相同

arguments 对象的长度是由实参个数而不是形参个数决定的

```
<script>
function fn(a,b)
{
    console.log(fn.length); //得到是 函数的形参的个数
    //console.log(arguments);
    console.log(arguments.length); // 得到的是实参的个数
    if(fn.length == arguments.length)
    {
        console.log(a+b);
    }
    else

```



```
{
    console.error("对不起，您的参数不匹配，正确的参数个数为:" + fn.length);
}
//console.log(a+b);
}
fn(1,2);
fn(1,2,3);
</script>
```

Var a = 10, b = 20;

等价的

Var a = 10;

Var b = 20;

1.3 案例 鼠标展示



代码：

```
var box = document.getElementById("box");
function fn(liid,bg){ // 封装函数 参数的传递
    var obj = document.getElementById(liid);
    obj.onmouseover = function(){
        box.style.backgroundImage = bg;
    }
}
fn("li01","url(images/01big.jpg)"); // 实参
fn("li02","url(images/02big.jpg)");
fn("li03","url(images/03big.jpg)");
```

```
fn("li04","url(images/04big.jpg)");  
fn("li05","url(images/05big.jpg)");
```

1.4 返回值 return

定义：

一个函数实际上就是一个计算过程，计算完成之后的结果就是返回值。

定义函数的返回值：

在函数内部用 `return` 来设置返回值，一个函数只能有一个返回值。

同时，终止代码的执行。

所有的自定义函数默认没有返回值；

`Return` 后面不要换行

```
function $(id){  
    return document.getElementById(id);  
}  
  
$("#demo").style.backgroundColor = 'purple';  
$("#test").style.backgroundColor = "blue";
```

1. 以下代码执行的结果是多少？请分析为什么？

```
1   var total=10;  
  
1   var number = square(5);  
  
2   alert(number);  
  
3   function square(n) {  
  
4       total = n*n;  
  
5       return total;
```

```
6 }
```

结果是 25

1.5 算术运算符

+ - * / % ^

1+1 = 2

5%2 == 1

2%5 == 2

2^3

A++ ++后置 每次自加 1 先运算后自加

++a ++前置 每次自加 1 先自加 后运算

1.分析代码，得出正确的结果。

```
1 var a=10, b=20 , c=30;  
2 ++a;  
3 a++;  
4 e=++a+(++b)+(c++)+a++;  
5 alert(e);
```

77

1.6 条件语句(if)

If(条件表达式) { 语句;}

```
If() {}else {}
```

```
If() else if(){} else if(){} else {}
```

1.7 获得焦点 失去焦点 事件

我们前面学过了 `onclick` 点击 `onmouseover` `onmouseout`

获得焦点: `onfocus` fao ~ 克死

失去焦点: `onblur` 不 len ~



```
<script>
window.onload = function(){

    var txt = document.getElementById("txt");

    txt.onfocus = function(){ //得到焦点
        //alert("得到了焦点");
        //什么时候该清空呢
        //用户没有输入的时候，用户第一次使用的时候
        // 如果 这里 input 里面的文字 是 请输入... 说明用户没有用过，就应该清空
        if(txt.value == "请输入...")
        {
            txt.value = "";
            txt.style.color = "#333";
        }
    }

    txt.onblur = function(){ //失去焦点
        //alert("失去了焦点");
        //什么时候再还原呢？
        //input 的值是 空的时候，我们再复原
        if(txt.value == "")
        {
            txt.value = "请输入...";
            txt.style.color = "#ccc";
        }
    }
}
```

```
    }  
  }  
}  
</script>
```

1.8 this (自己的)

指的是本身

This 主要是指事件的调用者。

className 类名

```
$("#result").className = "wrong";
```

innerHTML

更换 盒子里面的内容 文字 标签都换。

表单更换内容

Input.value

isNaN nan 不是一个数字 is 是 是不是一个数字

isNaN("12") 如果里面的不是个数字 返回 true 否则返回 false

1.9 属性和方法

手机

黑色的 5.5 寸的 塑料的 铁的

外在特性

属性

手机的颜色是黑色的。

Iphone.color = "red"; 属性给值一定是等号

方法:

手机 打电话 发短信 玩游戏 聊 QQ 看电影

动词 可以干什么

Iphone.tel();

方法和属性的区别:

方法一律带有小括号 。 isNaN(); 动词

方法给值: isNaN (“值”);

1.9.1 表单自动获得焦点

Txt.focus(); 方法

Onfocus 事件

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <script>
    window.onload = function(){
      var txt = document.getElementById("txt");
      txt.focus(); //自动获得焦点
    }
  </script>
</head>
<body>
自动获得焦点:
<input type="text" id="txt"/>
</body>
</html>
```

1.9.2 鼠标经过选择表单

```
sele.onmouseover = function(){  
    this.select(); //选择  
}
```

方法 `select()` 选择功能



1.10 For 循环

人生一个循环

`i+=3` `i=i+3`

1.11 `getElementsByName()` 获取某类标签

前面我们可以得到一个盒子 通过 `id` 获得

`getElementById()` 只得到一个 盒子

我们要获取某类标签 比如说所有 的 `div` `li` `span`

`getElementsByName()`; 很多个所以是复数 很多个

得到的是一个伪数组。

`Lis` 数组

Lis[索引号] 一个

1.12 作业

1. 编写一个程序，计算增加后的工资。要求工龄满 5，增加 20% 工资；若 2—4，则增加 15%；若小于 2，则增加 10% 工资。
2. 编写一个程序，用户输入一个数，可以求出这个数的阶乘。
3. 利用 for 循环 遍历 京东背景图片。

