

7. JavaScript函数

(1) JavaScript的函数：

- 函数是什么：
 - 函数（function）是一段完成指定任务的已命名代码块。函数可以遵照给它的一组值或参数完成特定的任务，并且可能返回一个值。
- 在JavaScript中有两种函数：自定义函数与系统函数。
- 函数的优越性：
 - 控制程序设计的复杂性
 - 提高软件的可靠性
 - 提高软件的开发效率
 - 提高软件的可维护性
 - 提高程序的重用性
- 标准格式：

```
function 函数名 ([参数列表..]) {  
    函数体。。。  
    [return 返回值;]  
}
```

- JavaScript三种定义函数方法：
 - 第一种是使用function语句定义函数 如上面格式
 - 第二种是使用Function()构造函数来定义函数（不常用）
 - var 函数名 = new Function("参数1", "参数2", "参数3"....."函数体");
 - 如： `var 函数名 = new Function("x","y","var z=x+y;return z;");`
 - 第三种是在表达式中定义函数
 - var 函数名 = function(参数1, 参数2, ...){函数体};
 - 例如：

```
//定义  
var add = function(a,b){  
    return a+b;  
}  
//调用函数  
document.write(add(50,20));
```

- 函数的使用：

```
//通过函数名直接调用函数
//函数名(参数1, 参数2, ...)
myadd(10,20);

//函数自调用
(function () {
    console.log("Hello Function! ");
})();
```

- arguments 对象
 - 在函数代码中，使用特殊对象 arguments，开发者无需明确指出参数名，就能访问它们。
 - arguments是一个对象,是一个伪数组，arguments[索引]---->实参的值
 - 也可以访问这个值，即第一个参数的值（第一个参数位于位置 0，第二个参数位于位置 1，依此类推）。
 - arguments.length--->是实参的个数
- 关于变量作用域：
 - 由于函数的出现导致了变量的定义分为函数内定义和函数外定义，即全局和局部变量
 - 全局变量:在函数外使用var声明的变量就是全局变量,全局变量可以在页面的任何位置使用
 - 全局变量,如果页面不关闭,那么就不会释放,就会占空间,消耗内存。
 - 局部变量:在函数内使用var关键字定义的变量是局部变量，即出了函数外边无法获取。
 - 隐式全局变量:没有使用var声明的变量,就叫隐式全局变量，在函数外可以获取到。
 - 全局变量是不能被删除的,隐式全局变量是可以被删除的（使用delete）
 - 定义变量使用var是不会被删除的,没有var是可以删除的
 - 函数外面定义的变量是全局变量，函数内可以直接使用。
 - 在函数内部没有使用var定义的=变量则为全局变量。
- 函数中的返回值：
 - 如果一个函数中有return ,那么这个函数就有返回值
 - 如果一个函数中没有return,那么这个函数就没有返回值
 - 如果一个函数中没有明确的返回值,那么调用的时候接收了,结果就是undefined
 - (没有明确返回值:函数中没有return或函数中有return,但是return后面没有任何内容)
 - 函数内容执行遇到return，则立刻执行返回，且下面的代码是不会被执行的

(2) 系统中常用的内置函数：

```
escape()    //字符串编码
unescape()  //字符串反编码
*eval()     //将参数字符串作为脚本代码来执行。
*isNaN()    // is not a number (不是一个数值)
*parseInt()
*parseFloat()
```