一、函数

1） 作用

封装一段待执行的代码

2）语法

```
//函数声明
function 函数名(参数列表){
  函数体
  return 返回值;
}
//函数调用
函数名(参数列表);
```

3）使用

函数名自定义，见名知意，命名规范参照变量的命名规范。普通函数以小写字母开头，用于区分构造函数(构造函数使用大写字母开头，定义类)

4）匿名函数

匿名函数：省略函数名的函数。语法为：

- 匿名函数自执行

```
(function (形参){

})(实参);
```

- 定义变量接收匿名函数

```
var fn = function (){};
fn(); //函数调用
```

5）作用域

JavaScript中作用域分为全局作用域和函数作用域， 以函数的{ }作为划分作用域的依据

1. 全局变量和全局函数
    - 只要在函数外部使用var关键字定义的变量,或函数都是全局变量和全局函数,在任何地方都可以访问
    - 所有省略var关键字定义的变量,一律是全局变量
2. 局部变量/局部函数
    - 在函数内部使用var关键字定义的变量为局部变量,函数内部定义的函数也为局部函数,只能在当前作用域中使用,外界无法访问
3. 作用域链
   局部作用域中访问变量或函数,首先从当前作用域中查找,当前作用域中没有的话,向上级作用域中查找,直至全局作用域

二、 内置对象

1） 对象

对象是由属性和方法组成的,使用点语法访问

2） Array 数组

1. 创建

2. 特点

- 数组用于存储若干数据,自动为每位数据分配下标,从0开始
- 数组中的元素不限数据类型,长度可以动态调整
- 动态操作数组元素： 根据元素下标读取或修改数组元素， arr[index]

3. 属性和方法

1. 属性 : length 表示数组长度,可读可写
2. 方法 :
    - push(data)
      在数组的末尾添加一个或多个元素,多个元素之间使用逗号隔开
      返回添加之后的数组长度

- pop()
  移除末尾元素
  返回被移除的元素
- unshift(data)
  在数组的头部添加一个或多个元素
  返回添加之后的数组长度
- shift()
  移除数组的第一个元素
  返回被移除的元素
- toString()
  将数组转换成字符串类型
  返回字符串结果
- join(param)
  将数组转换成字符串,可以指定元素之间的连接符,如果参数省略,默认按照逗号连接
  返回字符串
- reverse()
  反转数组,倒序重排
  返回重排的数组,注意该方法直接修改原数组的结构
- sort()
  对数组中元素排序,默认按照Unicode编码升序排列
  返回重排后的数组,直接修改原有数组
  参数 : 可选,自定义排序算法
  例:

```
//自定义升序
function sortASC(a,b){
  return a-b;
}
```

  作用: 作为参数传递到sort()中,会自动传入两个元素进行比较,如果a-b>0,交换元素的值,自定义升序排列

```
//自定义降序
function sortDESC(a,b){
  return b-a;
}
//如果返回值>0,交换元素的值,b-a表示降序排列
```

4. 二维数组

数组中的每个元素又是数组

```
var arr1 = [1,2,3];
var arr2 = [[1,2],[3,4],[5,6,7]];
//操作数组元素
var r1 = arr2[0] //内层数组
var num = r1[0]; //值 1
//简写
var num2 = arr2[1][0];
```

3）String 对象

1. 创建

```
var str = "100";
var str2 = new String("hello");
```

2. 特点

字符串采用数组结构存储每位字符,自动为字符分配下标,从0开始

3. 属性

length： 获取字符串长度

4. 方法

- 转换字母大小写
  toUpperCase() 转大写字母
  toLowerCase() 转小写字母
  返回转换后的字符串,不影响原始字符串
- 获取字符或字符编码
  charAt(index)　　获取指定下标的字符
  charCodeAt(index) 获取指定下标的字符编码
  参数为指定的下标,可以省略,默认为0
- 获取指定字符的下标
  - indexOf(str,fromIndex)
    作用 : 获取指定字符的下标,从前向后查询,找到即返回
    参数 :
    str 表示要查找的字符串,必填
    fromIndex 表示起始下标,默认为0
    返回 :
    返回指定字符的下标,查找失败返回-1
  - lastIndexOf(str,fromIndex)
    作用 : 获取指定字符最后一次出现的下标,从后向前查找,找到即返回
    参数 :
    str 必填,表示要查找的内容
    fromIndex 选填,指定起始下标
- 截取字符串
  substring(startIndex,endIndex)
  作用 : 根据指定的下标范围截取字符串,startIndex ~ endIndex-1
  参数 :
  startIndex 表示起始下标
  endIndex 表示结束下标,可以省略,省略表示截止末尾
- substr(startIndex,len)

  作用：根据下标截取指定的字符串
- 分割字符串
  split(param)
  作用 : 将字符串按照指定的字符进行分割,以数组形式返回分割结果
  参数 : 指定分隔符,必须是字符串中存在的字符,如果字符串中不存在,分割失败,仍然返回数组

- 模式匹配
- 正则表达式对象 RegExp

  RegExp : Regualr Expression

  1. 语法 :
     var reg1 = /微软/ig;
     var reg2 = new RegExp('匹配模式','修饰符');
     正则表达式对象可以接收一个变量。
  2. 属性 :

     lastIndex : 可读可写，表示下一次匹配的起始索引
     注意 :

     1. 默认情况下，正则表达式对象不能重复调用方法,
        如果重复调用，结果会出错:
        由于 lastIndex 保存再一次匹配的起始下标,
        重复调用时，不能保证每次都从下标0开始
        验证，可以手动调整 lastIndex 为 0。
     2. 只有正则对象设置全局匹配 g ， 该属性才起作用。
  3. 方法 :

     test(str) :验证字符串中是否存在满足正则匹配模式的内容，存在则返回true,

     不存在返回false参数为要验证的字符串。
- 作用 : 借助正则表达式实现字符串中固定格式内容的查找和替换
  正则表达式 :
  var reg1 = /字符模式/修饰符;
  修饰符 :
  i : ignorecase 忽略大小写
  g : global 全局范围
  字符串方法 :
  - match(regExp/subStr)
    作用 : 查找字符串中满足正则格式或满足指定字符串的内容
    返回 : 数组,存放查找结果
  - replace(regExp/subStr,newStr)
    作用 : 根据正则表达式或字符串查找相关内容并进行替换
    返回 : 替换后的字符串,不影响原始字符串。

4) Math 对象

1. 定义

Math对象主要提供一些列数学运算的方法

2. 属性

1. 圆周率 : Math.PI
2. 自然对数 : Math.E

3. 方法

1. Math.random(); 生成0-1之间的随机数
2. Math.ceil(x); 对x向上取整,忽略小数位,整数位+1
3. Math.floor(x); 对x向下取整,舍弃小数位,保留整数位
4. Math.round(x); 对x四舍五入取整数

5）日期对象

1. 创建日期对象

```
1. var date2 = new Date("2011/11/11");
2. var date3 = new Date("2011/11/11 11:11:11");
```

2. 日期对象方法

1. 读取或设置当前时间的毫秒数：getTime()
2. 获取时间分量
    - getFullYear()
    - getMonth()
    - getDate()