**1.循环**

循环就是将**相同或相似**的代码反复执行

两要素：

循环条件：控制循环次数

循环体： 被执行的相同或相似代码

1. while 循环

|  |
| --- |
| while（循环条件）{  循环体（可内嵌if-else语句）  }  满足循环条件则执行循环体。直至不满足或break |

1. break 与 continue

在任何形式循环中插入break，可以提前结束循环。

ps：

isNaN( ) 判断一个值是否为NaN， 是返回true

Number（''） 空字符 返回 0 。

1. do-while 循环

|  |
| --- |
| do{  循环体;  }while(循环条件);  先执行一遍循环体，再判断循环条件，即使错误也会执行一次 |

1. for循环

|  |
| --- |
| for(初始值; 循环条件; 循环里的增量){  循环体  } |

for循环三个语句都可以以其他方式代替，循环条件若有两个以后一个为准。

例：

9\*9乘法表第三行

var num=''; //原理：数字+ 字符串 ——》 字符串

for (var i=1; i<=3 ; i++)

{ num += i+'\*3='+ i\*3 +' ';

}

console.log(num);

闰年：

for (var year = 2000; year<2101 ; year++ )

{ if ((year%4==0 && year%100 !=0)||year%400==0)

{ console.log(year);

}

}

|  |
| --- |
| break：结束循环，不会再执行后续循环  continue：跳过后面的循环体，继续执行增量（下一次循环） |

例：

1.求和1-100内偶数和

for (var i=1,sum=0; i<101 ;i++ )

{ if (i%2==1)

{ continue;

} sum += i;

}console.log(sum);

2.打印1-100间所有整数，除掉 能被3和4整除的

for (var i=1;i<101 ;i++ )

{ if (i%3==0 ||i%4==0)

{ continue;

}console.log(i);

}

1. 循环嵌套

任何的循环之间可以相互嵌套