目录

1.流程控制语句

- 程序的三大流程语句
  1. 顺序结构： 代码自上而下顺序执行。
  2. 分支结构/选择结构/条件结构： 要根据不同的条件执行不同的结果
  3. 循环结构： 重复的去做一件事情。
- 顺序结构

```
// 题目 ：输入两个数，交换两个数的位置，再输出他们交换后的结果。
var niunai = 20;         //牛奶容器
var kafei  = 5;          //咖啡容器
var kong   = niunai;     //空杯子
niunai = kafei;          //牛奶取咖啡的值
kafei = kong;            //咖啡取空杯子中牛奶的值
alert(niunai + '|' + kafei);
```

- 分支语句/选择语句/条件语句
  - 单分支语句：

```
// if(判断的条件){
//     执行语句;(判断条件为true时执行)
// }
var num = 10;
if(num % 2 == 0){
    alert('这是一个偶数');
}
```

    - 如果执行语句只有一行，可以省略{}

```
if(条件) alert("true");
else     alert("false");
```

  - 双分支语句：

```
var num = 10;
if(num % 2 == 0){
    alert('这是一个偶数');
}else{
    alert('这是一个奇数')
}
```

    - 编写分支语句步骤：
        1. 确定这个判断条件应该写什么。
        2. 根据不同的结果，编写对应的执行语句。
    - `if()`里面可以写任何表达式，都会自动数据类型转换为布尔值。
  - 多分支语句

```
if(判断条件1){
    // 执行语句1;
}else if(判断条件2){
    // 执行语句2;
}else if(判断条件3){
    // 执行语句3;
}else{
    // 执行语句N;(上述所有条件都不成立，执行这里)
}
```

    - 执行过程：从上往下，满足哪个条件就执行其相对应的语句，都不满足时，执行最后的else语句，只能进入其中之一。
- 选择结构 - switch语句

```
switch(表达式){
  case 常量1 : 执行语句1;break;
  case 常量2 : 执行语句2;break;
  case 常量3 : 执行语句3;break;
  ...
  case 常量n : 执行语句n;break;
  default : 执行语句;break;(上述的case选项匹配失败，执行这里)
  // 常量是不能改变的量
}
```

  - 表达式的结果等于哪个`case`的常量，则执行其后的语句，执行完`break`后就跳出`switch`结构，*都不满足*则执行`default`语句。
  - break的作用：是跳出switch结构，没有break，则继续执行下面分支的语句。

  - 题目：按照考试成绩的等级，输出对应的百分制数段。

```
var degree = 'B';
switch(degree){
  case 'A':
    alert('80~100');
    break;
  case 'B':
    alert('70~79');
    break;
  case 'C':
    alert('60~69');
    break;
  default : alert('error');
}
```

        - 一般情况下不要省略`break`。`break`代表的是终止当前`switch`语句。
        - 一般情况下不要省略`default`。
    - 省略`switch`语句中的`break`简化代码

```
/*
  输入月份，显示当前月的天数。
  答案看6_省略break.html
  习题：输入年月日，计算某一天是该年的第几天(周)
 */
```

    - 匹配确定的值，使用`switch`语句。需要判断的值，用`if`语句。

2.三目运算符

- `表达式1 ? 表达式2 : 表达式3;`
    1. 先去判断`表达式1`是否为真
    2. `表达式1`为真，直接去执行`表达式2`
    3. `表达式1`为假，直接去执行`表达式3`
        - 本质是if语句中的双分支语句

```
//使用三目运算符 判断一个数的奇偶性。
  var num = 11;
  num % 2 == 0 ? alert(num + '是一个偶数') : alert(num + '是一个奇数');
```

    - 三目运算符只需要在最后加一个分号;

3.循环结构。

- 循环 : 重复的去做一件事情。
- 三种循环 :
    - while循环
    - do_while循环
    - for循环
- while循环:

```
while(循环条件){
  循环语句;
}
```

    - 循环条件成立就执行循环语句，直到循环条件不成立为止。
    - `while()`里面可以写任何表达式，都会自动数据类型转换为布尔值。
      (可以利用数字的非零即真在循环条件中写i--/i++);
- 使用循环的好处: 1.代码简洁 2.代码没有冗余 3.后期维护方便
    - 循环条件永远成立，会造成死循环

- do...while循环(了解)

```
do{
  // 循环代码;
}while(循环条件);
```

  - `do...while`循环后面的分号;不要省略。
- do...while 和 while循环区别

```
// 1~100的和
// while()循环的写法
var i = 1;
var sum = 0;
while(i <= 100){
    sum += i;
    i++;
}
alert(sum);

// do...while写法
var i = 1;
var sum = 0;
do{
    sum += i;
    i++;
}while(i <= 100);
alert(sum);
```

  - while循环：先判断循环条件，满足条件再执行内部语句。
  - do...while循环：先执行一次循环语句，再去判断循环条件。
    - do...while使用场景：不管符不符合条件都要执行一次语句的时候。
- for循环

```
for(表达式1;表达式2;表达式3){
  // 执行语句;
}
```

1. 先求解`表达式1`(只求一次)
2. 求解`表达式2`，若其值为真(非0),则执行for语句中指定的内嵌语句;
3. 然后求解`表达式3`;再求解`表达式2`,若为假,则结束for循环，执行for循环外的语句。

```
// 计算1~100的和
var sum = 0;  //记和
for(var i = 1; i <= 100; i++){
    sum += i;
}
alert(sum);
```

4.break和continue关键字

- break： 终止当前循环。

- continue： 跳过这次循环，直接进入下次循环。
- 死循环扩展

```
// 死循环 ： 循环条件永远成立
// 利用break跳出循环;
while(1){
  // 常用的
  break;
}
do{
}while(1);

for(;;){
}
```

5.循环嵌套

- 循环嵌套 : 不是语法, 循环内部写循环。
  - **循环嵌套循环，变量命名不能重复。** 命名 : i j k l m n(习惯性)
- 扩展 : ` `半角空格，只占一个字符的一半。 ` `全角空格，占一个字符。

6.循环练习（答案看12_循环练习.html）

1. 一个新入职，月工资为2000的员工，每一年上涨上一年工资的5%，到20年时月工资为?
2. 山上有一口缸可以装满50升水，现在有15升水。老和尚叫小和尚下山挑水，每次可以挑5升，
   - 问 小和尚要挑几次水才能把水缸挑满。
3. 打印100-200之间所有能被3或者7整除的数
4. 计算10的乘阶 (1*2*3*4*5*6*7*8*9*10 *n的乘阶 1*2*...*n)
5. 计算1+3+5...+99的和
6. 99乘法表 *
7. 判断一个数是不是合数。
   (指自然数中除了能被1和本身整除外，还能被其他的数整除，不包括0)
8. 判断一个数是不是质数。(除了1和他本身以外，不再有其他的除数能整除)
9. 求 1 - 1/2 + 1/3 - 1/4 + 1/5 ... 1/100

7.循环的拓展练习(答案看13_循环的拓展练习.html)

```
// 1.找出所有的水仙花数，三位数，个位立方和等于概数本身。

// 2.输入两个数，求两个数的最大公约数。(能够同时整除两个数的最大数)

// 3.输入两个数，求两个数的最小公倍数。(能够同时被两个数整除的最小数)
```