

day-03

01switch 是一种特殊的分支结构

```
switch(表达式 变量){  
    case 值:  
        do something...  
        [break;]  
    case 值:  
        do something...  
        [break;]  
    case 值:  
        do something...  
        [break;]  
    [default:]  
}
```

- switch case结构 一般配合break语句使用
break不是必须的 要看需求
- switch 做分支 其实是判断变量 等于 某个范围内的特定值
- 如果case没有配合break语句使用 则会向下穿透执行
直到遇到break语句为止
- 所有的switch结构都可以使用if-else 代替
- switch和if-else语句
 - 但是不是所有的if-else都可以用switch代替
 - if-else 可以判断 相等 不相等 等情况，适用性广
 - switch-case 只能判断相等情况，结构清晰，效率高

02while

```
while(布尔表达式){  
    需要重复执行的代码 1  
}
```

- while是一个循环结构 小括号内放置一个布尔表达式
- 当布尔表达式的结果为真(true) 则执行花括号内的语句
- 花括号内执行完毕会自动回到布尔表达式 继续判断 如果结果还为true则继续执行花括号内的内容
- 直到布尔表达式的结果为false 则跳过循环结构
- 特殊情况
 - 没有变成false的情况 就叫做 死循环
 - 所谓死循环 指的是循环没有出口 一直执行
 - 在页面中如果有死循环 会导致页面卡死
 - 写循环时 需要注意给循环添加出口(让布尔表达式有变成false的情况)

03break

- break关键字 可以在任何循环中使用
- 它在循环中的作用是 跳出(结束)循环
- 在循环中 如果执行了break语句 后续的代码就不再执行

04continue

- continue 关键字 可以在所有循环语句中使用
- 作用 跳过本次循环 进入下一次循环
- 在控制台输出100以内 各位不等于3的数字
- 当执行到continue 跳过当前循环执行 直接进入下一次的条件判断

05do-while语句

```
do{  
    需要循环执行的代码 1  
}while(布尔表达式);
```

- 先执行一遍 再判断条件(布尔表达式)
- while 和 do-while 区别
 - while当循环条件不满足时 一次都不执行
 - do-while 不管循环条件是否满足 至少执行一次

06for语句

```
for(表达式1;表达式2;表达式3){  
    需要循环的代码 1  
}
```

- for语句格式
 - 表达式1 1. 表示循环初始化条件 变量声明和赋值
 - 表达式2 2. 表示循环条件 布尔表达式
 - 表达式3 3. 增量 改变循环条件 一元运算
- 执行顺序
 - 1. 初始化 执行表达式1 只执行一次
 - 2. 判断循环条件
 - 3. 循环条件为true 执行循环花括号内的语句
 - 3.1循环条件为false 结束循环 跳过for语句
 - 4. 执行过花括号内的语句 才执行 执行表达式3
 - 5. 判断循环条件
 - 6. 循环条件为true 执行循环花括号内的语句
 - 6.1循环条件为false 结束循环 跳过for语句
 - 7. 执行过花括号内的语句 才执行 执行表达式3

07三种循环结构

- 在大部分情况下是允许互换的(明确的知道循环的次数)
- 一般明确循环次数 使用 for
- 不明确循环次数 使用 while 或 do-while