

day2-javascript逻辑分支

笔记本： 凡尘：二阶段

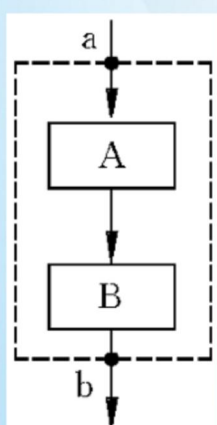
创建时间： 2017/7/4 9:34

更新时间： 2017/9/25 17:22

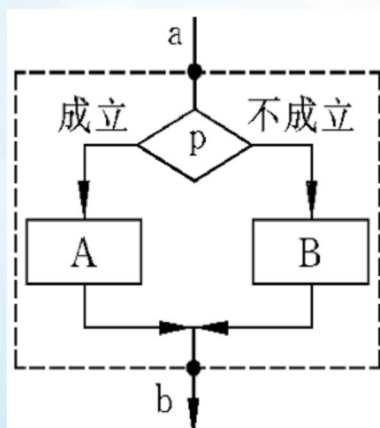
作者： 437389128@qq.com

1、程序的三大结构

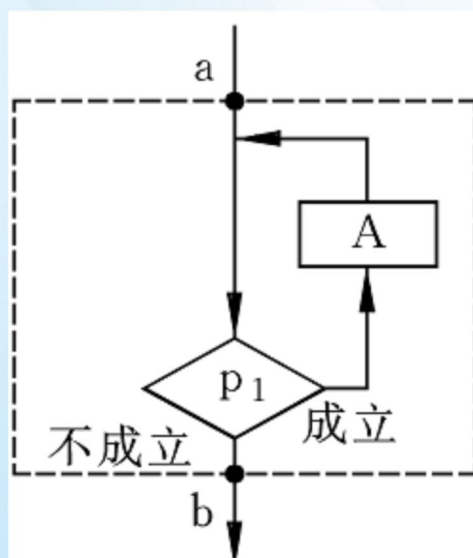
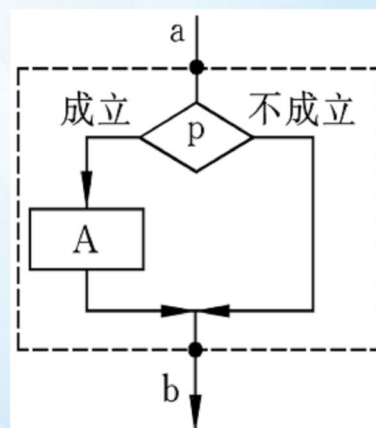
顺序、选择、循环



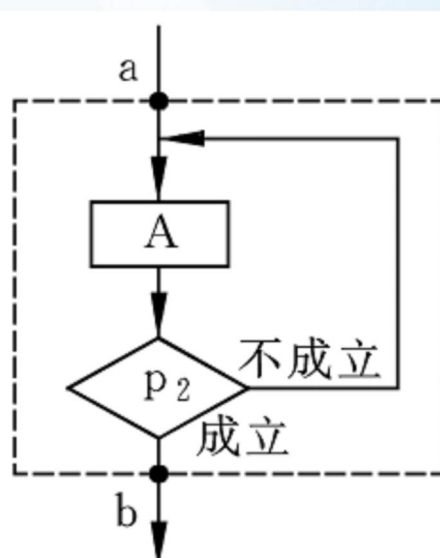
顺序结构



选择结构



(a)



(b)

循环结构

2、If(如果)语句

3、if判断注意的地方

- 1、if后面的()不能省略，()后面不要加分号
- 2、if分支中即使一句话也要把{}写上
- 3、if嵌套不要超过三层

if判断接受的数据类型

非布尔类型的条件判断

if(2) number类型 ----->所有的非0数字代表 true , 0代表false

if("abc") string类型 -----> 所有的非空字符串都代表 true , "" 代表false

if(obj) object类型 ----->所有的非null的对象都代表true , null代表false

if(undefined) -----> 代表false ,

if(NaN) -----> 代表false

if(function) -----> 如果函数存在 , 代表true

练习：if练习题

- 1、判断一个整数，属于哪个范围：大于0；小于0；等于0
- 2、判断一个整数是偶数还是奇数，并输出判断结果
- 3、字体放大时限制最大为32px，最小为12px
- 4、输入年份，判断是否为闰年
- 5、输入数字，显示星期几
- 6、输入分数，判断成绩等级
- 7、开发一款软件，根据公式（身高-108）*2=体重，可以有10斤左右的浮动。来观察测试者体重是否合格

if扩展（一）

if是我们用的最多的,值得注意的一点是if其实和||很像,举一个简单的小栗子,成熟吃...

```
if(a){
  ....代码块1
}else{
  ....代码块2
}
```

```
var b = 3>5||6<7;
```

如果a为true的话那么就会执行代码块1,然后忽略掉代码块2.这样的结果其实跟我们昨天讲的||一样。如果||前面为true的时候也会忽略掉后面的代码块。

优化:基于这个性质,我们可以把用的最多的代码放在前面,减少判断次数。

if扩展 (二)

如果在js中有很多if判断,而且可能执行的次数分布也比较均匀,那么后面的判断语句每次都要把前面的判断一个一个执行完,这样是不利于优化的。比较好的做法是将一层判断语句变成两层判断语句

栗子:

```
var a = 88;
if(a>90 && a<=100){
  console.log('优秀')
}else if(a>80 && a<90){
  console.log('良')
}....
```

转换:

```
if(a>60&&a<=100){
  ....代码块
}else{
  ....代码块
}
```

注意: 嵌套if语句的时候最好不要超过三层

4、switch判断

```
switch(n){
  case 1:
    ...代码块1
    break;
  case 2:
    ...代码块2
    break;
  default:
    当n与case1和case2不同时执行的代码
}
```

工作原理：首先设置表达式`n`(通常是一个变量)。随后表达式的值会与结构中的每个`case`值做比较。如果存在的话，则执行`case`关联的代码块。结束的时候一定要用`break`来组织代码自动向下一个`case`运行

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`就跳出`switch`结构，都不满足则执行`default`的语句

javascript中没有限制`switch`后面的表达式的结果是什么类型的

练习：switch练习题

1、输入数字，显示星期几

2、输入分数，判断成绩等级

```
var str = 80;
var level=0;
if(str>90&&str<=100){
    level=1;
}else if(str>=80&&str<90){
    level=2;
}

switch(level){
    case 1:
        console.log('A');
        ....
}
```

3、输入月份，显示当月的天数

5、switch注意的地方

- 1、`switch`语句里把`default`分支一定要写上去，防止在今后的维护中产生歧义
- 2、把执行概率更大(正常情况)放在前面
- 3、`case`的执行代码块后面一定要加上`break`;(在不需要`case`的穿透效果的时候)

6、case的穿透应用

7、console.log

8、if和switch的应用场景

if :

- 1、具体的值进行判断
- 2、区间的判断
- 3、对运算的结果是boolean类型表达式进行判断 true flash

switch :

- 1、对具体的值进行判断
- 2、值的个数是固定的

对于几个固定的值判断，建议使用switch 语句。因为switch 语句会将具体的答案都加载进内存，效率相对高一点

基于代码的可读性：如果条件较少时,if-else容易阅读,而条件较多时switch更容易阅读

9、switch中的小坑

switch中的判断是相等(==)还是恒等(===)?

```
var a = '5';
switch(a){
  case 5:
    console.log('==');
    break;
  case '5':
    console.log('===');
    break;
  default;
}
```

10、面试题

```
var x =2;
switch(x){
  case 1: console.log(1);
  case 2: console.log(2);
  default:break;
  case 3: console.log(3);
}
```

这段小程序的输出结果是？

break: 跳出本次执行的程序

11、转义字符

\"
\'
\
\n 表示换行
\t tab键

综合应用

根据一个数字日期，判断这个日期是这一年的第几天

例如：20160211，计算后结果为42

作业：

- 1、开发一款软件，根据公式（身高-108）*2=体重，可以有10斤左右的浮动。来观察测试者体重是否合

1、堆与栈

2、小数位数保留

预习

- 1、预习循环
 - 0、三元表达式
 - a、while循环（重）
 - b、do{}while()循环
 - c、for循环（重）
 - d、break（重）
 - e、continue（重）
- 2、已知现在夏天的平均温度为35度，以后每年都会上升2%，问20年后夏天的平均温度是多少（保留整数）？（使用for循环实现）
- 3、打印99乘法表