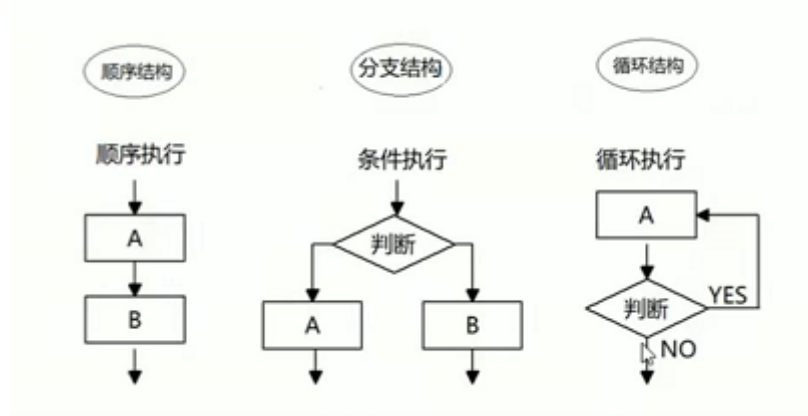


流程控制概念

在一个程序执行的过程中，各条代码的执行顺序对程序的结果是有直接影响的。我们可以通过控制代码的执行顺序来实现想要显示的效果。

简单来说：流程控制就是来控制代码按照哪一种结构来执行。

流程控制主要有三种结构，分别是 顺序 结构、分支 结构和 循环 结构，这三种结构代表三种代码执行的顺序

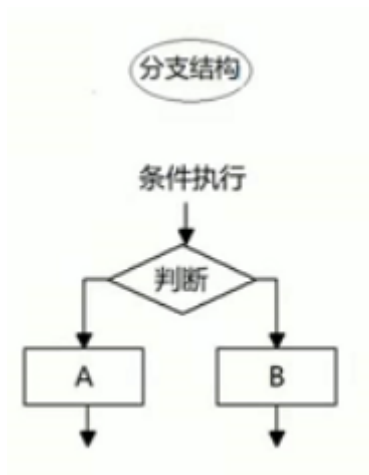


顺序结构的流程控制（了解）

顺序结构是程序中最简单、最基本的流程控制，没有特定的语法结构，程序会按照代码的 先后顺序，依次执行，程序中大部分代码都是这样执行的。

分支结构的流程控制（重要）

由上到下执行代码的过程中，根据不同的代码路径（通过不同的条件选择不同的代码路径），从而得到不同的结果。



JS 语言提供了两种分支结构语句：**if语句** **switch语句**

if分支语句

```
1      <script>
2          //if的语法结构，if如果
3          if (条件表达式) {
4              //执行语句
5          }
6          //执行顺序，如果if里面的条件表达式为真true时，则执行大括号里面的执行语句，如果条件
          表达式为false时，则不执行大括号里面的语句，按照顺序结构执行代码。
7      </script>
```

```
1      <script>
2          if (3 < 5) {
3              console.log('测试');
4          }
5      </script>
```

案例：

弹出一个输入框，要求用户输入年龄，如果年龄大于等于 18 岁，允许进网吧

```
1
2      <script>
3          var age = prompt('...');
4          if (age >= 18 ) {
5              console.log('欢迎光临');
6          }
7      </script>
```

if else 分支语句

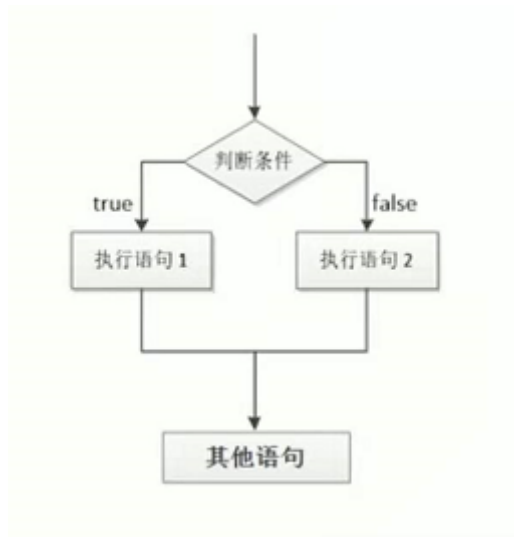
语法结构

```
1      <script>
2          // 条件成立，执行if里面代码，否则执行else里面的代码
3          if(条件表达式)
4          {
5              //[如果]条件成立执行的代码
6          }
7          else{
8              //[否则]执行的代码
9          }
10     </script>
```

案例：

```
1      <script>
2          var age = prompt('输入值');
3          if (age >=18) {
4              console.log('大于等于18');
5          } else {
6              console.log('小于18');
7          }
8      </script>
```

执行流程



if else if 语句（多分支语句）

利用多个条件来选择不同的语句执行，得到不同的结果。

语法结构

```
1  <script>
2      if(条件表达式1){
3          语句1;
4      }else if(条件表达式2){
5          语句2;
6      }else if(条件表达式3){
7          语句3;
8      }else{
9          //上述条件都不成立执行此处代码
10
11      }
12  </script>
```

如果条件表达式1满足就执行，语句1执行完毕后，退出整个if分支语句

如果表达式1不满足，则判断表达式2，满足的话，执行语句2，以此类推。

注意点：

- 1、多分支语句是多选1，最后 只有一个语句执行。
- 2、else if 理论上可以创建任意多个的 判断语句

练习：判断成绩级别

接收用户输入的分，根据分数输出对应的等级字母 A、B、C、D、E

其中：

90分(含)以上，输出：A

80分(含)~ 90 分(不含)，输出：B

70分(含)~ 80 分(不含)，输出：C

60分(含)~ 70 分(不含)，输出：D

60分(不含) 以下, 输出: E

案例分析:

按照从大到小的判断的思路

```
1      <script>
2          var fenshu = prompt('分数');
3          if (fenshu >= 90) {
4              console.log('90分以上');
5          }
6          else if(fenshu >= 80){
7              console.log('80分以上');
8          }
9          else if(fenshu >= 70){
10             console.log('70分以上');
11         }
12         else if(fenshu >= 60){
13             console.log('60分以上');
14         }
15         else {
16             alert('啊这, 分数。');
17         }
18     </script>
```

三元表达式

语法结构

条件表达式 ? 表达式1 : 表达式2

如果条件表达式结果为真, 则 `返回` 表达式1的值, 如果条件表达式结果为假 则 `返回` 表达式2的值

```
1      <script>
2          var num = 10;
3          var re = num > 5 ? '是的' : '不是的'; //10>5,为真, 返回表达式1的值。
4          console.log(re);
5      </script>
6
7
8      //使用if如何做出以上效果。
9      <script>
10         var num = 10;
11         if(num > 5 ){
12             console.log('正确');
13         }else{
14             console.log('错误');
15         }
16     </script>
```

switch语句（多分支语句）

`switch`语句也是多分支语句，它用于基于不同的条件来执行不同的代码。当要针对变量设置一系列的特定值的选项时，就可以使用`switch`。

语法结构

```
1  switch(表达式){
2      case value1:
3          //表达式等于 value1 时要执行的代码
4          break;
5      case value2:
6          //表达式等于value2 时要执行的代码
7          break;
8      ...
9      default:
10         //表达式不等于任何一个value时要执行的代码
11 }
12 //执行思路，利用我们表达式的值和case后面的选项值相匹配，如果匹配上，就执行该case里面的语
13 句，如果都没有匹配上，就执行default里面的语句。
```

例子：

```
1  <script>
2      switch (4) {
3          case 1:
4              console.log('这是1');
5              break;
6          case 2:
7              console.log('这是2');
8          case 3:
9              console.log('这是3');
10         default:
11             console.log('啥都不是');
12     }
13 </script>
```

switch注意事项

```
1  <script>
2      //switch注意事项
3      var num = 1;
4      switch (num) {
5          case 1:
6              console.log(1);
7              break;
8          case 2:
9              console.log(2);
10             break;
11         default:
12             console.log(3);
13     }
14     //1.开发里面，表达式我们经常写变量。
15     //2.我们num的值和case里面的值相匹配的时候是 全等 === ，必须是值和数据类型一致才可以
    num === 1;
```

```
16 //3.break如果当前的case里面没有break，则不会退出switch，会执行下一个case。
17 </script>
```

switch语句——案例（查询水果）

用户在弹出框输入一个水果，如果有该水果，就弹出水果的价格，没有就提示无。

思路：

- 1、弹出prompt输入框，让用户输入水果名称，把这个值取过来保存变量。
- 2、将这个变量作为switch括号里面的表达式。
- 3、case后面的值写几个不同的水果名称，注意一定要加引号，因为是全等匹配。
- 4、弹出不同的价格即可。同样注意每个case之后加上break，需要退出switch语句。
- 5、将default设置为没有无。

```
1 <script>
2 // 1、弹出prompt输入框，让用户输入水果名称，把这个值取过来保存变量。
3
4 // 2、将这个变量作为switch括号里面的表达式。
5
6 // 3、case后面的值写几个不同的水果名称，注意一定要加引号，因为是全等匹配。
7
8 // 4、弹出不同的价格即可。同样注意每个case之后加上break，需要退出switch语句。
9
10 // 5、将default设置为没有无。
11
12 var fruit = prompt('输入水果名称');
13 switch (fruit) {
14     case '苹果':
15         console.log('apple price is 3.5/500g');
16         break;
17     case '榴莲':
18         console.log('榴莲不好吃');
19         break;
20     default:
21         alert('无，我只有两个。');
22 }
23 </script>
```

多分支语句，switch语句和if else if语句的区别

- 1、一般情况下，两个语句可以相互替换。
- 2、switch...case 语句通常处理case为比较确定值的情况下，而if...else... 语句更加灵活，常用于范围判断（大于、等于某个范围）。
- 3、switch语句进行条件判断后直接跳到执行程序的语句中，效率更高。而if...else语句有几种条件，需要多次判断。
- 4、分支较少时，if...else语句的执行效率比switch语句高。

