



育知同创

YU ZHI TONG CHUANG

一、语句和程序结构

1.1 语句

ECMA-262 规定了一组语句（也称为流控制语句）。从本质上看，语句定义了 ECMAScript 中的主要语法，语句通常使用一或多个关键字来完成给定任务。语句可以很简单，例如通知函数退出；也可以比较复杂，例如指定重复执行某个命令的次数。

我们写的任何一行代码其实都是一条语句。一个程序总是由很多的语句来组成

1.2 程序流程控制

结构化程序有三种结构。顺序结构、选择结构、循环结构

- 顺序结构：顺序结构是一种线性、有序的结构，它依次执行各语句模块。
- 选择结构：选择结构是根据条件成立与否选择程序执行的通路。需要条件语句。
- 循环结构：循环结构是重复执行一个或几个模块，直到满足某一条件为止。需要循环语句。

二、if 条件语句

2.1 if 结构

语法：

```
if(condition){  
    //语句1  
}  
//if结构外面的代码
```

说明：

- 1、condition(条件)，可以是任意表达式，表达式的值不要求必须是布尔值。
- 2、如果condition不是布尔值，则使用转型函数Boolean()转换成布尔值。
- 3、如果对condition求值后的结果是true则执行 "语句1"。如果是false，则不执行 "语句1"，开始执行if结构外面的代码。

2.2 if...else 结构

语法：

```
if(condition){  
    //语句1  
}else{  
    //语句2  
}  
//if结构外面的代码
```

说明：

- 1、**condition**(条件)，可以是任意表达式，表达式的值不要求必须是布尔值。
 - 2、如果**condition**不是布尔值，则使用转型函数**Boolean()**转换成布尔值。
 - 3、如果对**condition**求值后的结果是**true**则执行 "语句1"。如果是**false**，则执行 "语句2"。
- 注意：从上面的说明可以看出，"语句1"和"语句2"永远不会同时执行，一定会执行其中的一句。

2.3 if...else if...elseif... 结构

语法：

```
if(condition1){  
    //语句1  
}else if(condition2){  
    //语句2  
}...else if(condition3){  
    //语句n  
}
```

说明：

- 1、首先判断**condition1**，如果是**true**，则执行"语句1"。
- 2、如果**condition1**是**false**，则判断**condition2**，如果是**true**，则执行"语句2"。以此类推。
- 3、如果中间碰到任何一个**condition**是**true**，则执行相应的语句，执行完毕之后，则整个**if**结构的语句也结束了。不会再判断后面的**condition**。
- 4、如果所有的**condition**都是**false**，则一个"语句"都不执行，整个**if**结构语句结束。

注意：这个结构中，有可能一个语句都不执行。

2.4 if...else if...else 结构

语法：

```
if(condition1){
    //语句1
}else if(condition2){
    //语句2
}...else if(condition3){
    //语句n
}else{
    //else语句
}
```

说明:

- 1、首先判断condition1, 如果是true, 则执行"语句1"。
- 2、如果condition1是false, 则判断condition2, 如果是true, 则执行"语句2"。以此类推。
- 3、如果中间碰到任何一个condition是true, 则执行相应的语句, 执行完毕之后, 则整个if结构的语句也结束了。不会再判断后面的condition。
- 4、如果所有的condition都是false, 则会自动执行else中的语句。

注意: 该结构中, 一定会有一个而且仅有一个语句执行。

三、 switch条件语句

switch语句也是与if语句一样使用广泛的选择语句。与c、java的语法一样, 但是使用起来比他们更灵活。

语法:

```
switch (expression) {
    case value1:
        //语句1
        break;
    case value2:
        //语句2
        break;
    case value3:
        //语句3
        break;
    case value4:
        //语句4
        break;
    default:
        //default语句
}
```

说明:

1. switch语句的含义是如果某个case的value与express相等, 则执行这个case对应的语句, 碰到break语句就结束switch语句。
2. 如果没有一个case都不匹配, 则自动执行default语句。
3. 可以没有default, 如果没有则当所有的case都不匹配的情况下, 自动结束switch, 一条语句都不执行。
4. 对JavaScript来说, expression可以是任何类型的(基本类型和引用类型), 而case后面的value可以是变量、常量、对象、表达式。(其他语言如: java和c则只能是常量, 而且expression表达式的类型也有限制)
5. 注意: 在执行的过程中, 如果匹配了某个case, 则从这个case的语句开始执行, 直到碰到break或者switch的结尾才会结束。

案例1:

```
var a = 2;
switch (a) {
  case 1:
    alert("1");
    break;
  case 2: //a的值是2, 所以与这个case匹配
    alert("2"); // 执行这个段代码
    break; //碰到break, switch语句结束
  case 3:
    alert("3");
    break;
  case 4:
    alert("4");
    break;
  default:
    alert("default")
    break;
}
```

案例2:

```
var a = 10;
switch (a) {
  case 1:
    alert("1");
    break;
  case 2:
    alert("2");
    break;
  case 3:
    alert("3");
    break;
  case 4:
    alert("4");
    break;
  default: //a 为10, 所以与所有的case都不匹配
    alert("default"); //执行default中的语句
    break;
}
```

案例3:

```
var a = 3;
switch (a) {
  case 1:
    alert("1");
    break;
  case 2:
    alert("2");
    break;
  case 3: //a为3, 与这个case匹配
    alert("3"); //开始执行case3中的语句
  case 4:
    alert("4"); //没有碰到break语句, 继续执行此语句
  default:
    alert("default"); //没有碰到break语句, 继续执行此语句
    break; //碰到break, switch语句结束。 如果此处没有switch语句, 代码也会结束, 后面没有代码了。哈哈
}
```