

switch语句

switch 语句中的每一种情形 (case) 的含义是: “如果表达式等于这个值 (value), 则执行后面的语句 (statement)”。而 break 关键字会导致代码执行流跳出 switch 语句。如果省略 break 关键字, 就会导致执行完当前 case 后, 继续执行下一个 case。最后的 default 关键字则用于在表达式不匹配前面任何一种情形的时候, 执行机动代码 (因此, 也相当于一个 else 语句)。

从根本上讲, switch 语句就是为了让开发人员免于编写像下面这样的代码:

```
if (i == 25){
    alert("25");
} else if (i == 35) {
    alert("35");
} else if (i == 45) {
    alert("45");
} else {
    alert("Other");
}
```

而与此等价的 switch 语句如下所示:

```
switch (i) {
    case 25:
        alert("25");
        break;
    case 35:
        alert("35");
        break;
    case 45:
        alert("45");
        break;
    default:
        alert("Other");
}
```

通过为每个 case 后面都添加一个 break 语句, 就可以避免同时执行多个 case 代码的情况。假如确实需要混合几种情形, 不要忘了在代码中添加注释, 说明你是有意省略了 break 关键字, 如下所示:

```
switch (i) {
    case 25:
        /* 合并两种情形 */
    case 35:
        alert("25 or 35");
        break;
    case 45:
        alert("45");
        break;
    default:
        alert("Other");
}
```

虽然 ECMAScript 中的 switch 语句借鉴自其他语言，但这个语句也有自己的特色。首先，可以在 switch 语句中使用任何数据类型（在很多其他语言中只能使用数值），无论是字符串，还是对象都没有问题。其次，每个 case 的值不一定是常量，可以是变量，甚至是表达式。请看下面这个例子：

```
switch ("hello world") {  
  case "hello" + " world":  
    alert("Greeting was found.");  
    break;  
  case "goodbye":  
    alert("Closing was found.");  
    break;  
  default:  
    alert("Unexpected message was found.");  
}
```

[*SwitchStatementExample03.htm*](#)

在这个例子中，switch 语句使用的就是字符串。其中，第一种情形实际上是一个对字符串拼接操作求值的表达式。由于这个字符串拼接表达式的结果与 switch 的参数相等，因此结果就会显示 "Greeting was found."。而且，使用表达式作为 case 值还可以实现下列操作：

```
var num = 25;  
switch (true) {  
  case num < 0:  
    alert("Less than 0.");  
    break;  
  case num >= 0 && num <= 10:  
    alert("Between 0 and 10.");  
    break;  
  case num > 10 && num <= 20:  
    alert("Between 10 and 20.");  
    break;  
  default:  
    alert("More than 20.");  
}
```



switch 语句在比较值时使用的是全等操作符，因此不会发生类型转换（例如，字符串 "10" 不等于数值 10）。