

17.1 语句概述

语句（Statement）表示一个完整、可执行的命令，用来完成特定的任务，使某件事情发生。语句之间通过分号分隔，程序都是由一个或多个语句组成的集合。当一个语句单独一行显示时，可以省略分号，JavaScript 会自动补加分号。

以结构划分，语句可以分为单句和复句。简单说明如下：

- 单句一般由一个或多个关键字和表达式构成，用来完成运算、赋值等简单任务。
- 复句一般由大括号构成，用来设计流程结构，控制程序的执行顺序。

程序中的代码结构可以分为三种类型：顺序结构、分支结构和循环结构。在正常情况下，所有代码将按顺序从上到下执行，这种结构是顺序结构。通过 if、switch、for、while 等流程语句可以改变代码的执行顺序，实现分支或循环执行顺序。

以关键字类型划分，JavaScript 中的语句可以分为很多类型，详细说明如表 E17.13 所示。

表 E17.13 JavaScript 语句类型

句子类型	句子子类型	逻辑概述	示例
声明语句	变量声明语句	为变量指定存储地址	var variable1 [= value1] [,variable2 [=value2]...];
	标签声明语句	为语句建立索引	label : statements
表达式语句	赋值语句	为变量克隆值	variable = value;
	函数调用语句	执行函数	function ();
	方法调用语句	执行函数	object.method();
	属性赋值语句	为对象变量克隆值	object.property = value;
分支语句	if	选择执行	if (expression) statements [else statements]
	switch	选择执行	switch (expression){ case label: statementList case label: statementList ... default: statementList }
循环语句	for	重复执行	for ([var] initialization ; test ; increment) statements
	for/in	遍历执行	for ([var] variable in <object array>) statements
	while	重复执行	while (expression) statements
	do/while	重复执行	do statements while (expression)
控制语句	continue	继续执行	continue [label];
	break	中断执行	break [label];
	return	执行返回	return [expression];
	throw	抛出异常	throw [exception];
	try	尝试执行	try { statements }

	catch	捕获异常	catch (exception){ statements }
	finally	最后执行	finally { statements }
其他语句	空语句	不执行	;
	with	暂时作用域	with (object) statements

【提示】

在上表中，被中括号括起来的选项表示可选部分，被尖括号括起来的选项表示必选部分，竖线表示列表项任选。其中包含的语法名次说明如下：

- variable：表示变量名。
- value：表示值。
- expression：表示表达式。
- statement：表示单句。
- statements：表示复合语句。
- statementList：表示语句列表。
- label：表示标签名。
- object：表示对象。
- initialization：表示初始值。
- test：表示测试表达式。
- increment：表示递增量。