1. 语法
2. 语句和表达式

* 语句相当于句子表达式相当于短语，运算符相当于标点符号和连接词。
* js中的表达式可以返回一个结果值。默认情况下，控制台会显示所执行的最后一条语句的结果值。

var a=3\*6;

var b;//声明语句

b=a;//赋值表达式

b;//表达式语句

* 如果在控制台输入var a=42;会得到结果值undefined而不是42
* 语法不允许我们获得语句的结果值并赋值给另一个变量。可以用eval但不支持。es7有一个do提案
* var a,b;
* a=eval("if(true){b=3+1}");
* console.log(a);//4

var a,b;

a=do{

if(true){

b=1+1;

}

};

console.log(a);//2

浏览器不支持

* ++a++会产生ReferenceError错误
* 语句系列逗号运算符将多 独立的表达式串联成一个语句

var a=42,b;

b=(a++,a);

console.log(a,b);//43 43

* 如果删除成功，delete返回true否则返回false
* contine foo并不是跳转到foo所在的位置继续执行，而是执行foo循环的下一轮循环。使用标签语句可以实现从内层循环跳转到外层循环

1. 运算符优先级

&& |

var a=42;

var b="foo";

console.log(a&&b);//foo

console.log(a||b);//42

用,来连接一系列语句时，它的优先级最低。

<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Operator_Precedence>

1. 有时js会自动为代码行补上缺失的分号，即分号自动插入(ASI)，只在换行时插入。ASI实际上是一个纠错(解析器错误)机制
2. js有运行时错误(TypeError ReferenceError SyntaxError等)，还有编译时错误(语法错误等)。程序有语法错误会无法运行。

TDZ即暂时性死区。由于代码中的变量还没初始化而不能引用的情况。对未声明的变量使用typeof不会产生错误，在暂时性死区中报错。

1. 函数参数

在es6中，如果函数参数省略或为undefined，则取参数默认值。