Js的核心语法只包括两部分:

1. 基本语法构造(比如操作符，控制结构，语句)
2. 标准库（就是一系列具有各种功能的对象比如 Array，date，Math等）。

除了之，各种宿主环境提供额外的 API（即只能在该环境使用的接口），以便 JavaScript 调用。以浏览器为例，它提供的额外 API 可以分成三大类。

* 浏览器控制类：操作浏览器
* DOM类：操作网页的各种元素
* Web类：实现互联网的各种功能

语句

先用 var 命令，声明了 变量，然后将 表达式 的结果赋值给 变量。

var a = 1+1

var a; a = 2;

语句以分号结尾，一个分号就表示一个语句结束(表达式不需要分号结尾)

var a = 1;

var b = 2;

变量

变量是对‘值’的具名引用,变量就是为‘值’起名,引用这个名字，就等同于引用这个值。变量的名字就是变量名

变量声明和赋值，是分开的两个步骤，

1. 先声明一个变量。

var a;

1. 表达式再赋值给变量。

a = 1;

JavaScript 的变量名区分大小写，A和a是两个不同的变量。

如果只是声明变量而没有赋值，则该变量的值是undefined。

如果一个变量没有声明就直接使用，JavaScript 会报错

变量提升

JS 引擎的工作方式是，先解析代码，获取所有被声明的变量，然后再一行一行地运行，所有的变量的声明语句，都会被提升到代码头部,这就叫变量提升

var a;

console.log(a);

a = 1;

标识符

JavaScript 有一些保留字，不能用作标识符：arguments、break、case、catch、class、const、continue、debugger、default、delete、do、else、enum、eval、export、extends、false、finally、for、function、if、implements、import、in、instanceof、interface、let、new、null、package、private、protected、public、return、static、super、switch、this、throw、true、try、typeof、var、void、while、with、yield。

**注释**

JavaScript 提供两种注释的写法：一种是单行注释，用//起头；另一种是多行注释，放在/\*和\*/之间。

## 区块

JavaScript 使用大括号，将多个相关的语句组合在一起，称为“区块”（block）。

{

//自带一个作用域

写多个相关的语句

}

对于var命令来说，JavaScript的区块不构成单独作用域

## 条件语句

JavaScript提供 if 结构 和 switch 结构，完成条件判断，即只有满足预设的条件，才能执行相对应的语句

* **if 结构**

布尔值：true代表真 false 代表假

如果布尔值为true就执行紧跟着的语句, () 的表达式如果为true就执行紧跟着后面的语句;如果为false，则跳过紧跟在后面的语句

if (布尔值) 语句；

if（m===3）m = m+1

上面的代码表示，只有在m等于3时，才会将其加上1

写法要求

这种写法要求条件表达式后面只能有一个语句。如果想执行多个语句，必须在if的条件判断之后，加上大括号，表示代码块（多个语句合并成一个语句）

if(m === 3){

m += 1;

}

建议总在if语句中使用大括号，因为这样方便插入语句;

* **if…else结构**

if代码块后，还可以跟一个else代码块，表示不满足条件时,所要执行的代码

if判断语句为false的时候 走 if else 代码块

if(m===3){

// 满足条件时，执行语句

} else {

//不满足条件时，执行语句

}

对同一个变量进行多次判断时，多个 if else语句可以链写在一起

if(m===0){

// ….

}else if (m ===1){

// ….

} else if (m === 2){

// ….

} else{

// ….

}

* switch 结构

多个if...else连在一起使用的时候，可以转为使用更方便的switch结构。

switch (fruit) {

case "banana":

// ...

break;

case "apple":

// ...

break;

default:

// ...

}

上面代码根据变量fruit的值，选择执行相应的case。如果所有case都不符合，则执行最后的default部分。需要注意的是，每个case代码块内部的break语句不能少，否则会接下去执行下一个case代码块，而不是跳出switch结构。

正确写法：switch结构每个对应的case下都需要写上 break语句让他跳出 switch语句；

switch (x) {

case 1:

console.log('x 等于1');

break;

default:

console.log('x 等于其他值');

}

switch语句部分和case语句部分，都可以使用表达式。

switch语句内部采用的是“严格相等运算符”， 这意味着比较时不会发生类型转换。