

js基础

标识符

中文是合法的标识符，可以用作变量名。

```
var 临时变量 = 1;
```

```
> var 临时变量 = 1;
   临时变量
< 1
```

注释

由于历史上 JavaScript 可以兼容 HTML 代码的注释，所以也被视为合法的单行注释。

```
> x = 1; <!-- x = 2;
   --> x = 3;
> 1
> x
< 1
```

需要注意的是，`-->`只有在行首，才会被当成单行注释，否则会当作正常的运算。

```
function countdown(n) {
  while (n --> 0) console.log(n);
}
countdown(3)
// 2
// 1
// 0
```

上面代码中，`n --> 0`实际上会当作`n-- > 0`，因此输出2、1、0。

循环

`for`语句的三个部分（initialize、test、increment），可以省略任何一个，也可以全部省略。

```
for ( ; ; ){
  console.log('Hello World');
}
```

标签

JavaScript 语言允许，语句的前面有标签（label），相当于定位符，用于跳转到程序的任意位置，标签的格式如下。

```
label:
  语句
```

标签可以是任意的标识符，但不能是保留字，语句部分可以是任意语句。标签通常与break语句和continue语句配合使用，跳出特定的循环。

```
top:
  for (var i = 0; i < 3; i++){
    for (var j = 0; j < 3; j++){
      if (i === 1 && j === 1) break top;
      console.log('i=' + i + ', j=' + j);
    }
  }
// i=0, j=0
// i=0, j=1
// i=0, j=2
// i=1, j=0
```

上面代码为一个双重循环区块，break命令后面加上了top标签（注意，top不用加引号），满足条件时，直接跳出双层循环。如果break语句后面不使用标签，则只能跳出内层循环，进入下一次的外层循环。

continue语句也可以与标签配合使用。

```
top:
  for (var i = 0; i < 3; i++){
    for (var j = 0; j < 3; j++){
      if (i === 1 && j === 1) continue top;
      console.log('i=' + i + ', j=' + j);
    }
  }
// i=0, j=0
// i=0, j=1
// i=0, j=2
// i=1, j=0
// i=2, j=0
// i=2, j=1
// i=2, j=2
```

上面代码中，continue命令后面有一个标签名，满足条件时，会跳过当前循环，直接进入下一轮外层循环。如果continue语句后面不使用标签，则只能进入下一轮的内层循环。