

# JS 基础

---

## 1. 标识符

中文是合法的标识符，可以用作变量名。

```
let 临时变量 = 1;  
临时变量 // 1
```

## 2. 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');  
}
```

## 3. 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：

```
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 (let i = 0; i < 3; i++){  
        for (let 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（`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 (let i = 0; i < 3; i++){
    for (let 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。