

[←](#) 2-22 Function Update

function update



Function



Default Parameters



对于函数而言，经常会用到参数，关于参数的默认值通常都是写在函数体中，如在 ES5 的时候大家都会这么写：



```
function f (x, y, z) {  
  if (y === undefined)  
    y = 7;  
  if (z === undefined)  
    z = 42;  
  return x + y + z;  
};  
f(1) === 50;
```

当一个函数有很多参数涉及初始化的时候，这样写代码极其丑陋，所以在 ES6 中改变了对这种知识的写法：

```
function f (x, y = 7, z = 42) {  
  return x + y + z  
}  
f(1) === 50
```

[!TIP]

函数参数是从左到右解析，如果没有默认值会被解析成 undefined

如果我们想让具体某个参数使用默认值，我们可以使用 undefined 进行赋值，如下段代码所示：

```
function f (x, y = 7, z = 42) {  
  return x + y + z  
}  
console.log(f(1, undefined, 43))// 51
```

在ES6中我们不仅可以给参数默认赋值具体的数值，同时参数赋值支持参数的逻辑运算进行赋值，如下段代码所示：

```
function f (x, y = 7, z = x + y) {  
  return z * 0.5  
}  
  
console.log(f(1, 7))// 4
```

拓展

在函数体内，有时候需要判断函数有几个参数，一共有2个办法。在 ES5 中可以在函数体内使用 arguments 来判断。

```
function test (a, b = 1, c) {  
  console.log(arguments.length)  
}  
test('a', 'b')//2
```

然而在 ES6 中不能再使用 arguments 来判断了，但可以借助 Function.length 来判断。

```
function test (a, b = 1, c) {  
  console.log(test.length)  
}  
test('a', 'b')// 1
```

细心的同学发现 Function.length 结果和 arguments 的结果不同！没错，Function.length 是统计第一个默认参数前面的变量数：