

Javascript 函数

陈一帅

实务学堂

介绍

基础

- 请说出下面代码的意义

Basic

代码

函数

- 减少重复
- 彼此隔离

定义与调用

- 定义：function 关键字

```
const 平方 = function(x) {  
    return x * x;  
};
```

- x: 函数参数，可以多个，也可以没有
- 函数体，可以 return 返回值，也可以没有返回值
- 调用

```
console.log(平方(12));  
console.log("未来是:", 平方(2));
```

函数声明

```
const 平方 = function(x) {  
    return x * x;  
};
```

- 另一种声明方法

```
function 平方(x) {  
    return x * x;  
}
```

箭头函数

```
function 平方(x) {  
    return x * x;  
}
```

- 另一种声明方法

```
const 平方 = (x) => { return x * x; };  
const 平方 = x => x * x;
```

输入变量

- 多余的输入变量会被忽略

```
const 平方 = x => x * x;
```

```
console.log("未来是:", 平方(2));
```

```
console.log(平方(4, true, "厉害"));
```


检查参数是否被提供

- 如果没提供, 则为 undefined

```
function a减b(a, b) {  
    if (b === undefined)  
        return -a;  
    else  
        return a - b;  
}  
  
console.log(a减b(10));  
console.log(a减b(10, 5));
```

默认参数

- 如果没提供，则利用默认参数

```
function a减b(a, b=0) {  
    return a - b;  
}  
console.log(a减b(10));  
console.log(a减b(10, 5));
```

生成特定参数的函数

- “乘多少” 返回一个函数
- 这个函数会把输入 x 乘 n 后，返回

```
function 乘多少(n) {  
  return x => x * n;  
}
```

- 得到一个会把输入乘2的函数

```
let 乘10 = 乘多少(10);
```

- 调用

```
console.log(乘10(3));
```

递归

- 调用自己

2^2

```
function 幂(底数, 指数) {  
    if (指数 == 0) {  
        return 1;  
    } else {  
        return 底数 * 幂(底数, 指数 - 1);  
    }  
}  
  
console.log(幂(2, 2));
```

练习1

- 写一个函数，返回两个数中的最小值
- 写一个函数，返回两个数中的最大值
- 写一个函数，返回两个函数，一个是最小值的函数，一个是最大值的函数
 - 生成挑选器(挑选类型: 1, 0) --

练习2: FCC

FCC Javascript 介绍

- 高能
- 需要4-10个小时