

JS_jQuery_Bootstrap

@M了个J
李明杰

<https://github.com/CoderMJLee>

<https://space.bilibili.com/325538782>



实力IT教育 www.520it.com



JavaScript

- JavaScript简称JS，是广泛应用于前端开发中的脚本语言
- JavaScript遵循了Java的表达式语法、命名规范、基础流程控制
- 这也是JavaScript从LiveScript更名的原因之一
- JS的标准化组织是[ECMA](#)，这种标准化的JS被称为ECMAScript，简称ES
- 2011/06，ES5.1
- 2015/06，ES2015，ES6
- 2016/06，ES2016，ES7
- 2017/06，ES2017，ES8
- 2018/06，ES2018，ES9
- 2019/06，ES2019，ES10

- 跟用户进行交互
- 对标签的增删改查
- 实现绚丽的动画效果
- 跟服务器进行交互
- ...

- [Mozilla MDN](#)

- [javascript.info](#)

script标签

- 在script标签中编写JS代码（script标签可以在head、body中）

```
<!-- type 的默认值就是 text/javascript -->  
<script type="text/javascript">  
    // 编写JS代码  
</script>
```

- 可以使用script标签引用js文件

```
<script src="js/jquery-3.4.1.min.js"></script>
```

■ 常见的打印方式

□ alert

□ console.log (推荐)

■ 每一条语句结束后可以加分号、也可以不加分号

□ 如果选择不加分号，那么每一条语句需要独立成行

■ JS 中有2种注释：单行注释、多行注释

```
<script>
// 单行注释
/*
    多行注释
    多行注释
*/
</script>
```

■ JS中用var、let定义变量（建议用let），const定义常量

```
let score = 10
score += 20
score += 50
console.log(score) // 80

const age = 10
console.log(age) // 10
```

■ JS中有8种数据类型

- **number**: 整数、浮点数, 1、-10、8.6
- **bigint**: 任意长度的整数, 12345678987654321n
- **string**: 字符串, "jack"、'jack'、`jack`
- **boolean**: 真假, true、false
- **undefined**: 未分配的值, undefined
- **null**: 不存在或无效, null
- **object**: 对象, {}、[]
- **symbol**: 唯一标识符

typeof运算符 (类型运算符)

■ 2种语法形式：返回x的类型

□ typeof x

□ typeof(x)

```
console.log(typeof 1) // number
console.log(typeof 1.0) // number
console.log(typeof 1n) // bigint
console.log(typeof 'jack') // string
console.log(typeof "jack") // string
console.log(typeof true) // boolean
console.log(typeof {
  name: 'jack',
  age: 10
}) // object
console.log(typeof [1, '2', true]) // object
console.log(typeof undefined) // undefined
console.log(typeof null) // object
console.log(typeof Symbol()) // symbol
console.log(typeof alert) // function
```


字符串的拼接

```
const name = 'jack'
const age = 20
// my name is jack, age is 20
console.log('my name is ' + name + ', age is ' + age)

// my name is jack, age is 20
console.log(`my name is ${name}, age is ${age}`)

// my name is jack, age is 20
console.log(`my name is ${'ja' + 'ck'}, age is ${19 + 1}`)
```

字符串的遍历

```
const str = 'cat'
for (const c of str) {
  // c a t
  console.log(c)
}

for (let i = 0; i < str.length; i++) {
  // 0 c
  // 1 a
  // 2 t
  console.log(i, str.charAt(i))
}
```

数组的遍历

```
const arr = [11, '22']
for (const e of arr) {
  // 11 22
  console.log(e)
}

for (let i = 0; i < arr.length; i++) {
  // 0 11
  // 1 22
  console.log(i, arr[i])
}

arr.forEach((e, idx) => {
  // 0 11
  // 1 22
  console.log(idx, e)
})
```

对象的遍历

```
const person = {name: 'jack', age: 20}

for (const k in person) {
  // name jack
  // age 20
  console.log(k, person[k])
}

for (const k of Object.keys(person)) {
  // name jack
  // age 20
  console.log(k, person[k])
}

for (const entry of Object.entries(person)) {
  // name jack
  // age 20
  console.log(entry[0], entry[1])
}
```

```
for (const v of Object.values(person)) {
  // jack
  // 20
  console.log(v)
}
```

常用API参考

- 数字: <https://zh.javascript.info/number>
- 字符串: <https://zh.javascript.info/string>
- 数组
 - <https://zh.javascript.info/array>
 - <https://zh.javascript.info/array-methods>

```
function sum(a, b, c = 10) {  
  return a + b + c  
}  
  
console.log(sum(1, 2, 3)) // 6  
console.log(sum(1, 2)) // 13  
console.log(sum(1)) // NaN  
console.log(sum()) // NaN
```

- 函数可以作为参数、返回值
- ES6开始，推出了箭头函数

```
function execute(fn, a, b) {  
  console.log(fn(a, b))  
}  
  
// 30  
execute(function (a, b) {  
  return a + b  
}, 10, 20)  
  
// 30  
execute((a, b) => {  
  return a + b  
}, 10, 20)  
  
// 30  
execute((a, b) => a + b, 10, 20)
```

```
function test() {  
  return alert  
}  
test()(123)
```

监听标签的点击事件

```
<span onclick="change()">换一换</span>
```

```
function change() {  
    console.log('点击了【换一换】')  
}
```

■ DOM操作就是指对HTML文档中节点的增删改查操作

■ 参考文档: [文档对象模型 \(DOM\)](#)

```
// 删除id为vv的节点
document.getElementById('vv').remove()

// 删除第一个div节点
document.getElementsByTagName('div')[0].remove()

// 删除第二个class为link的节点
document.getElementsByClassName('link')[1].remove()
```

```
// 删除id为vv的节点
document.querySelector('#vv').remove()

// 删除第一个div节点
document.querySelector('div').remove()

// 删除第二个class为link的节点
document.querySelectorAll('.link')[1].remove()
```


- jQuery是前端开发中非常著名的一个开源框架 (Query 的发音: 美['kwɪri])

- <https://jquery.com>

- 文档

- 英文: <https://api.jquery.com/>

- 中文: <https://www.jquery123.com/>

- 使用jQuery来操作DOM节点能够大大提高开发效率

```
$('#vv').remove()  
$('div')[0].remove()  
$('.link')[1].remove()
```

绑定点击事件

```
document.querySelector(sel).onclick = function () {  
      
}  
  
$(sel).on('click', function () {  
      
})  
  
$(sel).click(function () {  
      
})
```

基于jQuery的各种插件

■ jQuery UI

□ <https://jqueryui.com/>

□ <https://www.jqueryui.org.cn/>

■ jQuery Validation Plugin

□ <https://jqueryvalidation.org/>

Bootstrap

- Bootstrap是非常强大的前端库，可以帮助开发者快速开发出简洁美观、响应式的前端页面
- 由Twitter公司出品
- <https://getbootstrap.com/>
- <https://www.bootcss.com/>