链滴

ES6 新特性

作者：xixiaoming
原文链接：https://ld246.com/article/1531216806723
来源网站：链滴

一个简单的JavaScript时间线

1. 1995年：JavaScript以LiveScript之名诞生
2. 1997年：ECMAScript标准确立
3. 1999年：ES3发布，IE5非常生气
4. 2000年-2005年： XMLHttpRequest，熟知为AJAX，在如Outlook Web Access(2002)、Oddpost(2002)、Gmail(2004)、Google Maps(2005)中得到了广泛的应用
5. 2009年：ES5发布（这是我们目前用的最多的版本），带来了 forEach / Object.keys/ Object.create（特地为Douglas Crockford所造，JSON标准创建者），还有JSON标准。

2015年6月发布ES6（ES2015），下面介绍

ES6新特性

默认参数

```
// 以前
var link = function (height, color, url) {
    var height = height || 50
    var color = color || 'red'
    var url = url || 'http://azat.co'
    ...
}
// 现在
var link = function(height = 50, color = 'red', url = 'http://azat.co') {
  ...
}
```

模版表达式

```
// 以前
var name = 'Your name is ' + first + ' ' + last + '.'
var url = 'http://localhost:3000/api/messages/' + id
// 现在
var name = `Your name is ${first} ${last}`
var url = `http://localhost:3000/api/messages/${id}`
```

拆包表达式

```
// 以前
var data = $('body').data(), // 假设data中有mouse和house的值
  house = data.house,
  mouse = data.mouse
// 现在
var { house, mouse} = $('body').data() // 我们会拿到house和mouse的值的
```

箭头函数 =&>

```
// 以前
function fn(x) {
    return x * x;
}
// 现在
var fn = x => x * x;
```

Promise

```
// 以前
setTimeout(function(){
  console.log('Yay!')
}, 1000)
// 现在
var wait1000 = new Promise(function(resolve, reject) {
  setTimeout(resolve, 1000)
}).then(function() {
  console.log('Yay!')
})
// 或者
var wait1000 = new Promise((resolve, reject)=> {
  setTimeout(resolve, 1000)
}).then(()=> {
  console.log('Yay!')
})
```

块级作用域的let和const

1. 使用var声明的变量，其作用域为该语句所在的函数内，且存在变量提升现象；
2. 使用let声明的变量，其作用域为该语句所在的代码块内，不存在变量提升；
3. 使用const声明的是常量，在后面出现的代码中不能再修改该常量的值。

类

```
// 以前
function Point(x, y) {
  this.x = x;
  this.y = y;
}
Point.prototype.toString = function () {
  return '(' + this.x + ', ' + this.y + ')';
};
var p = new Point(1, 2);
// 现在
class Point {
  constructor(x, y) {
    this.x = x;
    this.y = y;
  }
  toString() {
    return '(' + this.x + ', ' + this.y + ')';
```

```
  }
}
```

模块化

模块功能主要由两个命令构成：export和import。export命令用于规定模块的对外接口，import命用于输入其他模块提供的功能。

```
// profile.js
var firstName = 'Michael';
var lastName = 'Jackson';
var year = 1958;
export {firstName, lastName, year}; // 输出一组变量
// 在main.js下引用，这种加载称为“编译时加载”或者静态加载，即 ES6 可以在编译时就完成模块载
import {firstName, lastName, year} from './profile.js';
// 同理可以输出函数：
export function multiply(x, y) {
  return x * y;
};
// 对外输出一个函数`multiply`
```