**Es6**

1. let、const

为js新增了块级作用域，所声明的变量，只在命令所在的代码块内有效

const声明的是常量

var a = [];

for(let i = 0; i < 10; i++){

a[i] = function(){

console.log(i);

}

}

a[6](); //6

当我们尝试去修改const声明的变量时，浏览器就会报错。因此，当我们引用第三方的库时声明的变量，用const来声明可以避免未来不小心重命名而导致出现BUG。

1. class、extends、super

class Animal {

constructor(){

this.type = “animal”;

}

says(say){

console.log(this.type + say);

}

}

let animal = new Animal();

animal.says(“hello”); //animal hello

class Cat extends Animal{

constructor(){

super();

this.type = “cat”;

}

}

let cat = new Cat();

cat.says(“hello”); //cat hello

class类，constructor里定义的属性和方法是实例对象自己的，constructor外面定义的属性和对象是所有实例对象共享的

Class之间可以通过extentds关键字实现继承

Super关键字代表父类的实例（this）。子类必须在constructor中调用super方法。因为子类没有自己的this对象，而是继承父类的this对象，然后对其进行加工

1. 箭头函数 arrow function

(i)=>i+1; 等同于 function(i){return i+1;}

如果方程式比较复杂，用{}包起来

(x,y)=>{

x++;

y--;

return x+y;

}

class Animal{

constructor(){

this.type = “animal”;

}

says(){

setTimeout(()=>console.log(this.type);, 1000)

}

}

使用箭头函数时，函数体内的this对象，就是定义时所在的对象，原因是箭头函数没有自己的this，它的this是继承外面的

1. template string 模板字符串

`woshi${name}\` has \` dog`

用 (\`) 代表 `

用 ${} 来引用变量

所有的空格和缩进都会被保存在变量之中

1. destructuring 解构

从数组或对象中提取值，对变量进行赋值

let dog= “a”, cat = “b”;

let obj = {dog, cat};

console.log(obj); //{dog:”a”, cat:”b”}

也可以这样：

let obj = {dog:”a”, cat:”b”};

let{dog, cat} = obj;

console.log(dog, cat); //a b

1. default 默认值

function f(arg = “test”){

console.log(arg);

}

只有传入的参数=== undefined，传入的变量才会被赋值为test

1. import、export

//content.js

export default “a cat”;

export function say(){return “hello”}

export const type = “dog”;

//index.js

import animal, {say, type} from “./content.js”;

console.log(`${animal} ${say()} ${type}`); //a cat hello dog

es6可以用as实现一键换名

//index.js

import animal, {say, type as animalType} from “./content.js”

console.log(`${animal} ${say()} ${ animalType }`); //a cat hello dog

可以使用整体加载，用一个 \* 指定一个对象，所有的输出都加载在这个对象上：

//index.js

import animal, \* as content from “./content.js”;

console.log(`${animal} ${content.say()} ${content.type}`); //a cat hello dog