后盾人 人人做后盾

www.houdunren.com

class类

后盾人 2011-2018

- ES6 提供了更接近传统语言的写法，引入了 Class（类）这个概念，作为对象的模板。通过class关键字，可以定义类。

- 基本上，ES6 的class可以看作只是一个语法糖，它的绝大部分功能，ES5 都可以做到，新的class写法只是让对象原型的写法更加清晰、更像面向对象编程的语法而已。

class类

```
//定义类
class Computer {
  constructor(color, size) {
    this.color = color;
    this.size = size;
  }

  playgame() {
    console.log(this.size+'寸的'+this.color+'颜色电脑可以吃鸡');
  }

}
```

类的示例

- 生成类的实例对象的写法，与 ES5 完全一样，也是使用new命令。前面说过，如果忘记加上new，像函数那样调用Class，将会报错。

```
class Point {
  // ...
}

// 报错
var point = Point(2, 3);

// 正确
var point = new Point(2, 3);
```

类的实例对象

class表达式

- 与函数一样，类也可以使用表达式的形式定义。

```
const MyClass = class Me {
  getClassName() {
    return Me.name;
  }
};
```

- 上面代码使用表达式定义了一个类。需要注意的是，这个类的名字是MyClass而不是Me，Me只在 Class 的内部代码可用，指代当前类。

- 如果类的内部没用到的话，可以省略Me，也就是可以写成下面的形式。

```
const MyClass = class {
    /* ... */
};
```

class表达式

类的继承

- Class 可以通过extends关键字实现继承。

```
class Point {
}
class ColorPoint extends Point {
}
```

上面代码定义了一个ColorPoint类，该类通过extends关键字，继承了Point类的所有属性和方法。但是由于没有部署任何代码，所以这两个类完全一样，等于复制了一个Point类。

- super这个关键字，既可以当作函数使用，也可以当作对象使用。在这两种情况下，它的用法完全不同。

- 第一种情况，super作为函数调用时，代表父类的构造函数。ES6 要求，子类的构造函数必须执行一次super函数。

super

```
class A {}
        class B extends A {
          constructor() {
            super();
          }
}
```

- 上面代码中，子类B的构造函数之中的super()，代表调用父类的构造函数。这是必须的，否则 JavaScript 引擎会报错。

super

- 第二种情况，super作为对象时，在普通方法中，指向父类的原型对象；在静态方法中，指向父类。

```
class A {
  p() {
    return 2;
  }
}
class B extends A {
  constructor() {
    super();
    console.log(super.p()); // 2
  }
}
let b = new B();
```

super