**class**

js语言的传统方法是通过构造函数，定义并生成新对象，是一种基于原型的面向对象系统。这种写法跟传统的面向对象语言（比如c++和java）差异很大，很容易让新学习这门语言的人感到困惑。所以，在es6中新增加了类的概念，可以使用class关键字声明一个类，之后以这个类来实例化对象。

原来的构造函数方法：

function Person(name, age) {

this.name = name;

this.age = age;

}

Person.prototype.showName = function() {

console.log(this.name);

}

Person.prototype.showAge = function() {

console.log(this.age);

}

var p1 = new Person('小王', 12);

p1.showName();

p1.showAge();

class的格式

class 构造函数名 {

constructor() {

// this.属性 = 属性

};

方法1() {};

方法2() {};

}

实例

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

};

showName() {

console.log(this.name);

};

showAge() {

console.log(this.age);

};

}

let p1 = new Person('小明', 10);

p1.showName();

p1.showAge();

console.log(typeof Person); // function，Person本质上还是一个函数

console.log(Person.prototype); // 原型

class的语法

1. Person函数中的constructor方法对应构造方法，this关键字则代表实例对象，也就是说，ES5的构造函数Person，对应es6的Person这个类的constructor。
2. Person这个类除了构造方法，还定义类的方法，注意：定义类的方法的时候，前面不需要加上function这个关键字，直接把函数定义放上去就可以了，另外，方法之间不需要逗号分隔，加了会报错。
3. 构造函数的prototype属性，在ES6的类上面继续存在，而且类的所有方法都定义在类的prototype属性上面。
4. 定义在类中的方法都是不可以被枚举的（不能被in方法找到）
5. constructor方法是类的默认方法，通过new命令生成对象实例时，自动调用该方法，一个类必须有constructor方法，如果没有显示定义，一个空的constructor方法会被默认添加。
6. 生成类的实例对象的写法，与ES5完全一样，也是使用new命令，如果忘记加上new，像函数哪样调用class，将会报错。

实例对象.hasOwnProperty("属性名"); 返回这个属性是不是这个对象自有的属性

**Class继承**

1. 子类继承父类，使用extends关键字
2. super:在构造函数中可以当一个函数来使用，相当于调用父类的构造函数
3. 要修改从父类继承过来的方法，可以直接在子类中写方法，就会覆盖从父类继承过来的方法
4. 父类的静态方法子类可以直接使用

// 父

class Student {

constructor(name) {

this.name = name;

}

hello() {

console.log('Hello, ' + this.name + '!');

}

}

// class 子 extends 父

class PrimaryStudent extends Student {

constructor(name, grade) {

super(name); // 记得用super调用父类的构造方法!

this.grade = grade;

}

myGrade() {

console.log('I am at grade ' + this.grade);

}

hello() {

console.log('修改父类的方法');

}

}

let p = new PrimaryStudent('彭作洪', '3');

console.log(p);

p.hello();

p.myGrade();