

## 24. 类 class 继承

学习要点：

### 1. 类的继承

本节课我们来开始学习 ES6 新增的类 class 的继承功能。

#### 一. 类的继承

1. ES6 也支持子类继承父类，使用 `extends` 关键字实现；

```
class Person {  
    //父类  
}  
  
class Children extends Person {  
    //子类继承  
}
```

2. 当子类继承了父类，实例化子类后，就可以直接拥有父类的构造、属性和方法；

```
class Person {  
    //父类  
    constructor(name) {  
        this.name = name;  
    }  
  
    get user() {  
        return this.name;  
    }  
  
    set user(value) {  
        this.name = value;  
    }  
  
    run() {  
        return 'name:' + this.name;  
    }  
}  
  
class Children extends Person {  
    //子类继承  
}
```

//子类继承父类后，父类的构造，get，set 以及方法均可使用

```
let c = new Children('Mr.Lee');
c.user = 'Mr.Wang';
console.log(c.user);
console.log(c.run());
```

3. 继承之后，一般来说，我们需要覆写父类，然后对子类进行增强；

```
class Children extends Person {
  //子类继承
  constructor(name, age) {          //覆写构造
    super(name);                    //执行父类构造并传参
    this.age = age;
  }

  run() {                            //覆写方法
    return super.run() + this.age;  //执行父类方法并返回内容
  }
}

//覆写
let c = new Children('Mr.Lee', 100);
console.log(c.run());
```

PS: super 作为函数时，调用父类构造；而作为对象时，在普通方法返回指定父类方法；

4. 可以使用 Object.getPrototypeOf()判断子类是否继承了父类；

```
console.log(Object.getPrototypeOf(Children) === Person);
```

5. ES6 的类支持静态属性和方法，也支持静态被子类继承；

```
class Person {
  static gender = '男';

  static go() {
    return 'GO GO GO !' + Person.gender;
  }
}

class Children extends Person {
  static gender = '女';          //覆写静态

  static go() {
    return 'o o o ' + Person.gender; //这里使用 Person 和 Children 是不一样的
  }
}
```

```
console.log(Person.gender);  
console.log(Person.go());  
console.log(Children.gender);  
console.log(Children.go());
```