

题目

- ◆ 如何准确判断一个变量是不是数组？
- ◆ 手写一个简易的 jQuery ，考虑插件和扩展性
- ◆ class 的原型本质，怎么理解？

知识点

- ◆ class 和继承
- ◆ 类型判断 instanceof
- ◆ 原型和原型链

class

- ◆ constructor
- ◆ 属性
- ◆ 方法

imoo

@3234109

继承

- ◆ extends
- ◆ super
- ◆ 扩展或重写方法

```
EXPLORER
js-demo
  CODE
    js-demo
      js class-demo-1.js
      js class-demo-2.js
      js deepClone.js
      A index.html

js class-demo-2.js
1 // 父类
2 class People {
3   constructor(name) {
4     this.name = name
5   }
6   eat() {
7     console.log(`${this.name} eat something`)
8   }
9 }
10
11 // 子类
12 class Student extends People {
13   constructor(name, number) {
14     super(name)
15     this.number = number
16   }
17   sayHi() {
18     console.log(`姓名 ${this.name} 学号 ${this.number}`)
19   }
20 }
```

类型判断 - instanceof

```
xialuo instanceof Student // true
xialuo instanceof People // true
xialuo instanceof Object // true

[] instanceof Array // true
[] instanceof Object // true

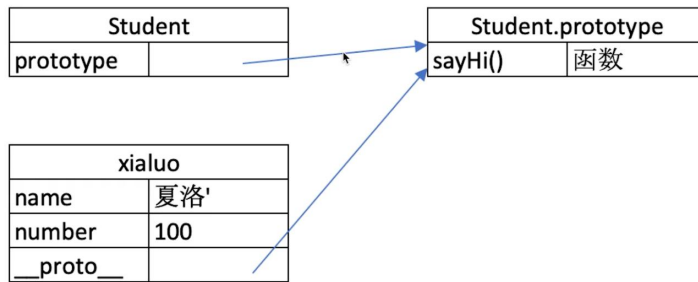
{} instanceof Object // true
```

原型

```
// class 实际上是函数，可见是语法糖
typeof People // 'function'
typeof Student // 'function'

// 隐式原型和显示原型
console.log( xialuo.__proto__ )
console.log( Student.prototype )
console.log( xialuo.__proto__ === Student.prototype )
```

原型



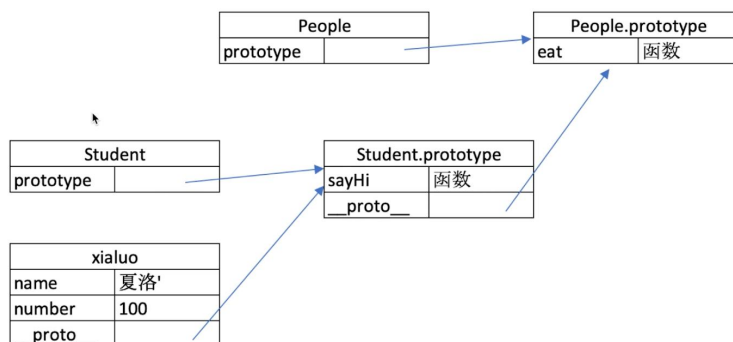
原型关系

- ◆ 每个 class 都有显示原型 `prototype`
- ◆ 每个实例都有隐式原型 `__proto__`
- ◆ 实例的 `__proto__` 指向对应 class 的 `prototype`

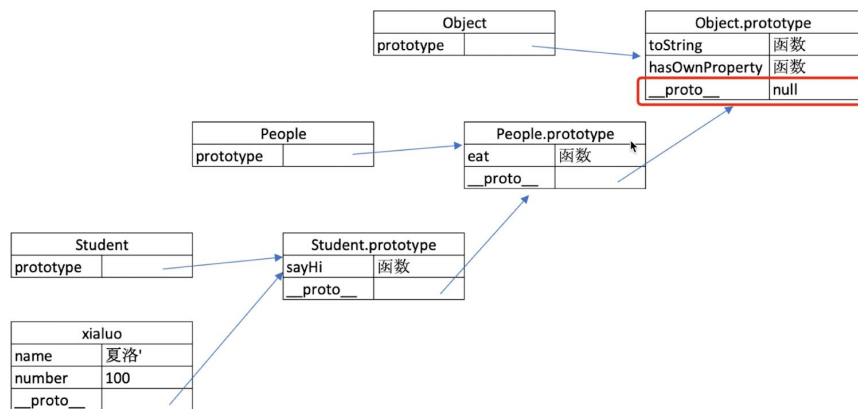
原型链

```
console.log( Student.prototype.__proto__ )  
console.log( People.prototype )  
console.log( People.prototype === Student.prototype.__proto__ )
```

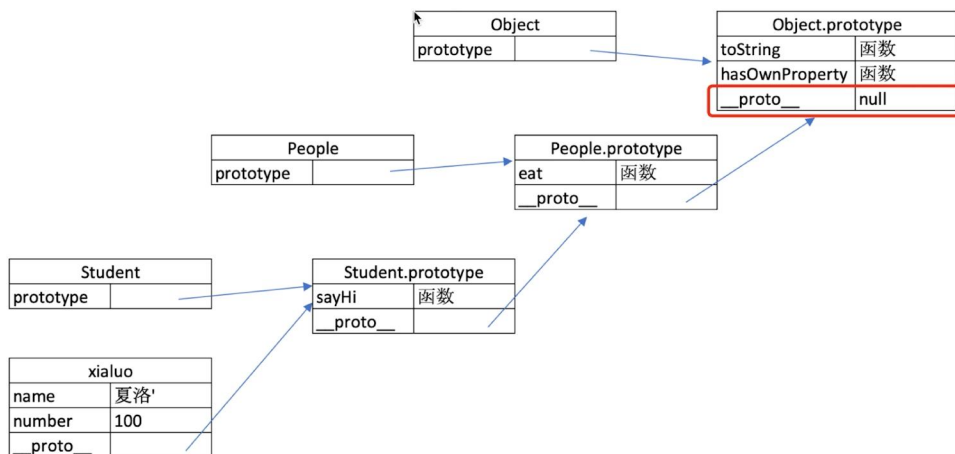
原型链



原型链



再看 instanceof



重要提示 !!!

- ◆ `class` 是 ES6 语法规则，由 ECMA 委员会发布
- ◆ ECMA 只规定语法规则，即我们代码的书写规范，不规定如何实现
- ◆ 以上实现方式都是 V8 引擎的实现方式，也是主流的

题目解答

- ◆ 如何准确判断一个变量是不是数组？
- ◆ 手写一个简易的 jQuery ，考虑插件和扩展性
- ◆ class 的原型本质，怎么理解？

如何准确判断一个变量是数组

- ◆ `a instanceof Array`

class 的原型本质

- ◆ 原型和原型链的图示
- ◆ 属性和方法的执行规则