[**深入理解JS原型与原型链**](https://www.cnblogs.com/fzgt/p/12732940.html)

getData(){

    /\*

    \* javascript原型

    \*/

*function* person(*name*,*sex*,*age*){

        this.name = name

        this.sex = sex

        this.age = age

        this.firstname = *function*(){

            return this.name + '------' + this.sex + '------' + this.age

        }

    }

*var* persona = new person('小明','男',28)

// 构造新对象

console.log(persona)

// 所有的函数对象都有prototype属性----从构造函数指向原型对象

console.log(*person*.prototype)

//所有的原型对象都有constructor属性，从原型对象指向构造函数

console.log(*person*.prototype.constructor)

//所有的对象都有\_\_proto\_\_属性----该属性从该对象指向该对象的原型对象

console.log(persona.\_\_proto\_\_ === *person*.prototype)

// 调用构造器中的方法

    console.log(persona.firstname())

    // \*/

    /\*

     \* javascript原型链

     \*/

*function* Foo1(){

        this.name1 = '1'

    }

*function* Foo2(){

        this.name2 = '2'

    }

*Foo2*.prototype = new Foo1()

*function* Foo3(){

        this.name = '3'

    }

*Foo3*.prototype = new Foo2()

*var* foo3 = new Foo3()

    console.dir(foo3.\_\_proto\_\_.\_\_proto\_\_)

}

**js中console.log()和console.dir()的区别**

**console.log()**会在浏览器控制台打印出信息

**console.dir()**可以显示一个对象的所有属性和方法