

原型链系统

一个神秘的指针

- `__proto__`，指向着一个对象，我们称为原型对象
 - `Object.setPrototypeOf(obj, prototype)`
 - `Object.getPrototypeOf(object)`
 - [`Object.prototype.isPrototypeOf\(\)`](#)
 - [`Object.prototype.hasOwnProperty\(\)`](#)
 - [`Object.create\(\)`](#)
- 原型对象还会有着自己的原型对象，形成原型链系统
 - 问题：最终的原型对象是什么。
- 它会引导着 `.` 运算符，从而形成ES的继承体系。

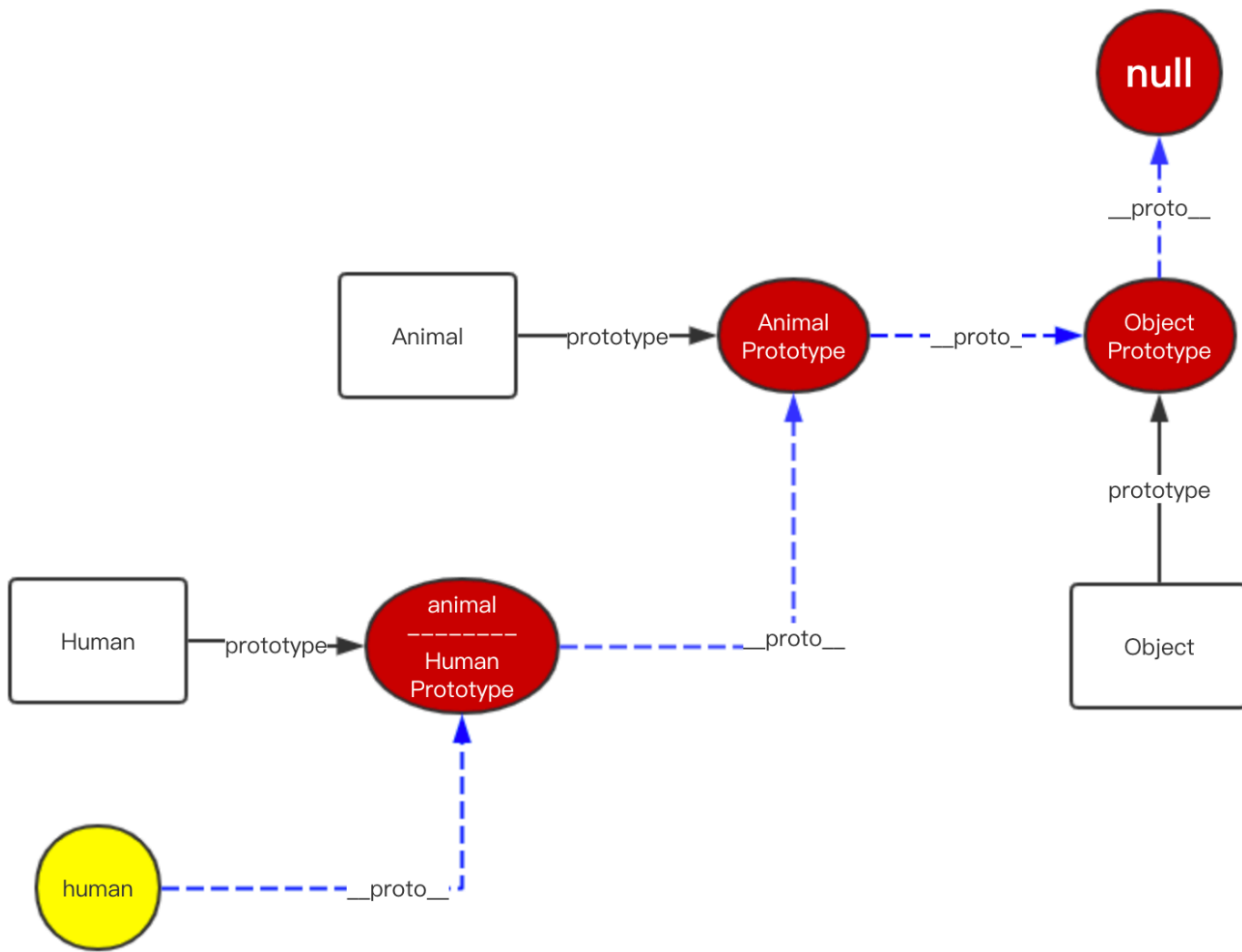
注意：

- 多个对象共享原型对象
- 与in遍历的关系
- 与prototype指针的关系

应用

- 创造“类”的概念
- 继承
- Instanceof运算符的真正含义

按图完成任务



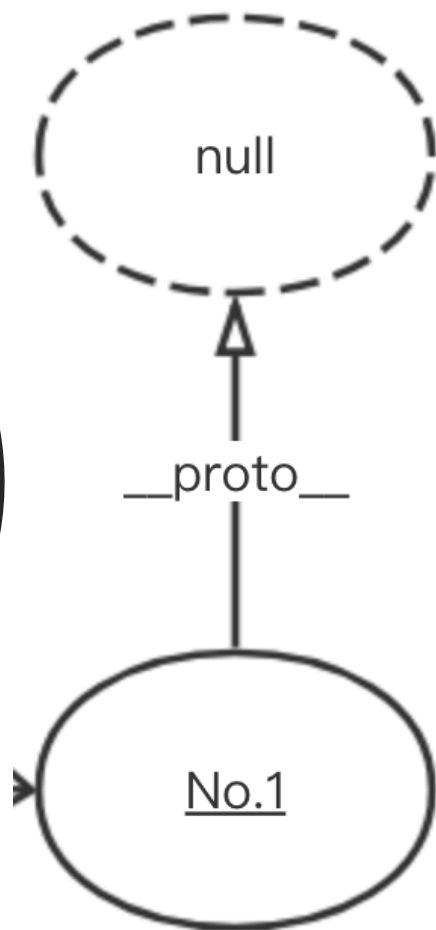
创世界

混沌



null

大佬诞生



null是最原始对象

No.1的_proto_指针，指向它，

表示“继承”它

万物工厂Object

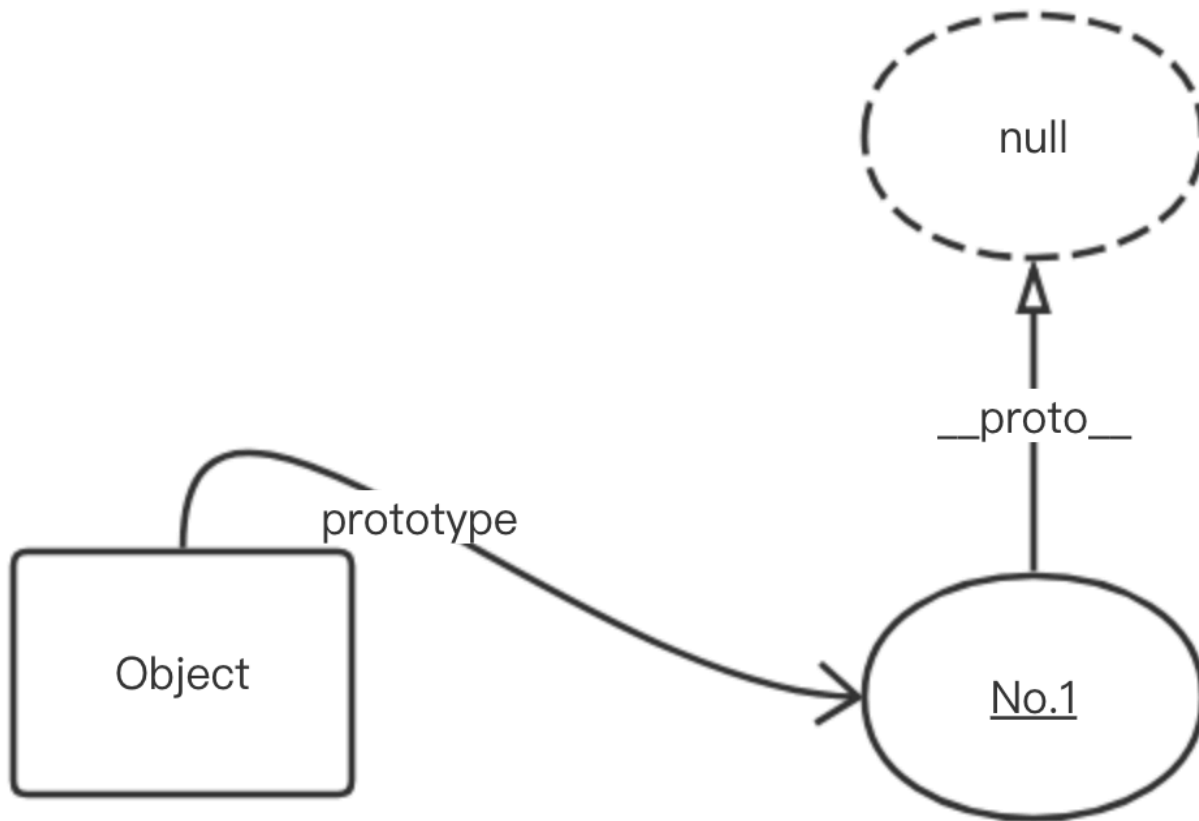
`new Object()`产生对象

该对象的`__proto__`会指向No.1

为什么？.....

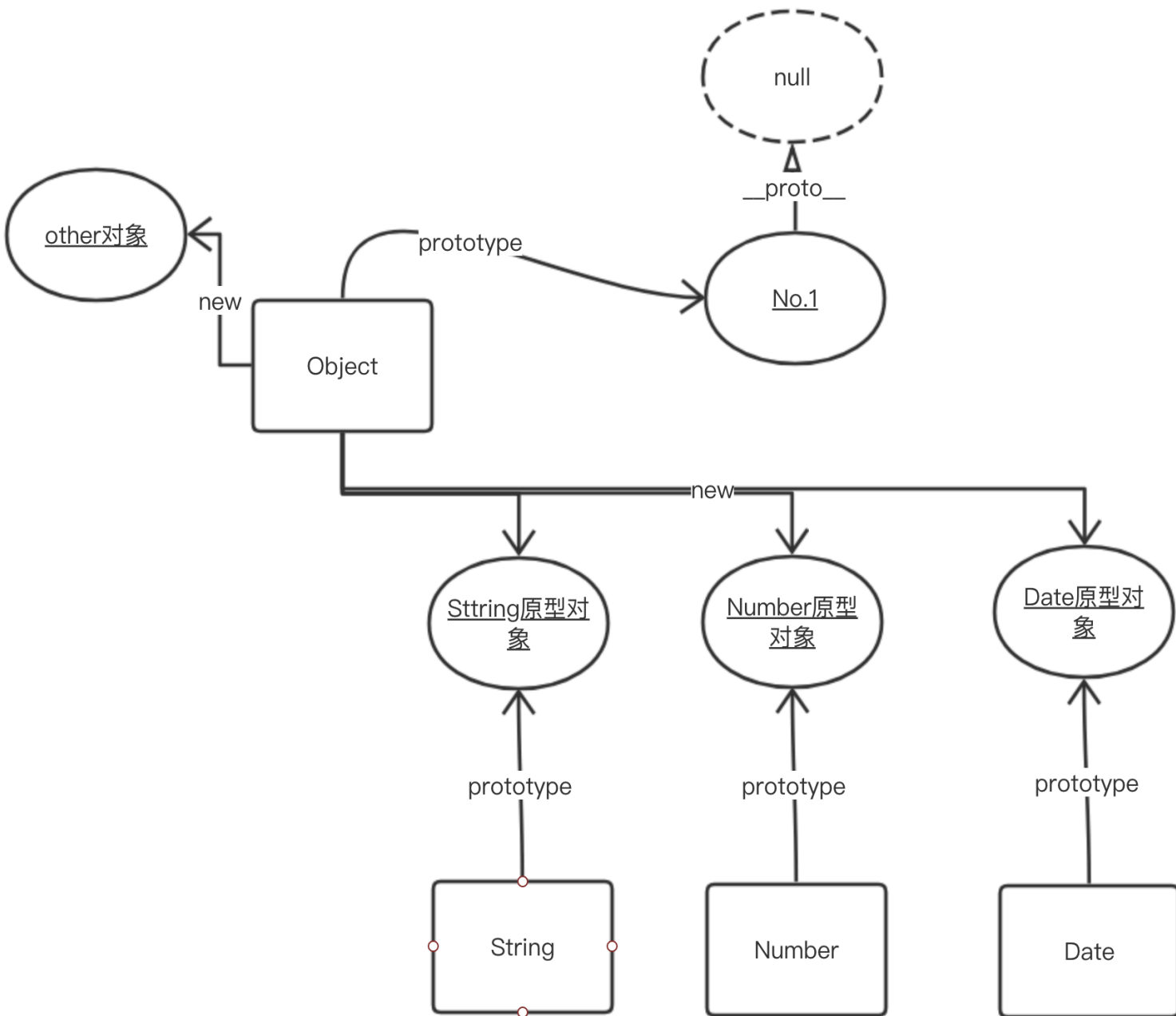
prototype表示：工厂指向的对象，将成为新对象的模板！！！！

No.1也被叫做“原型对象”



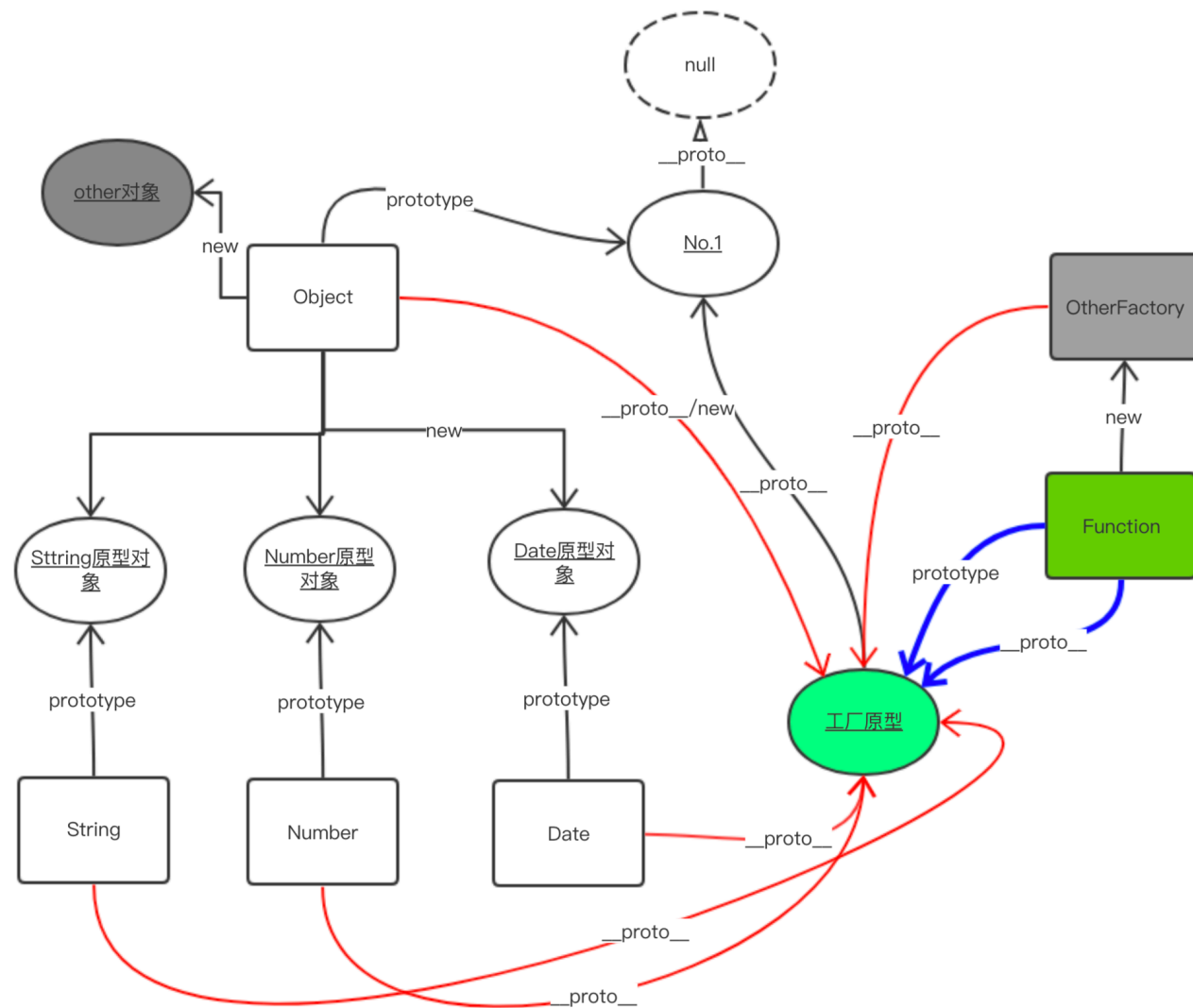
关于工厂的思考

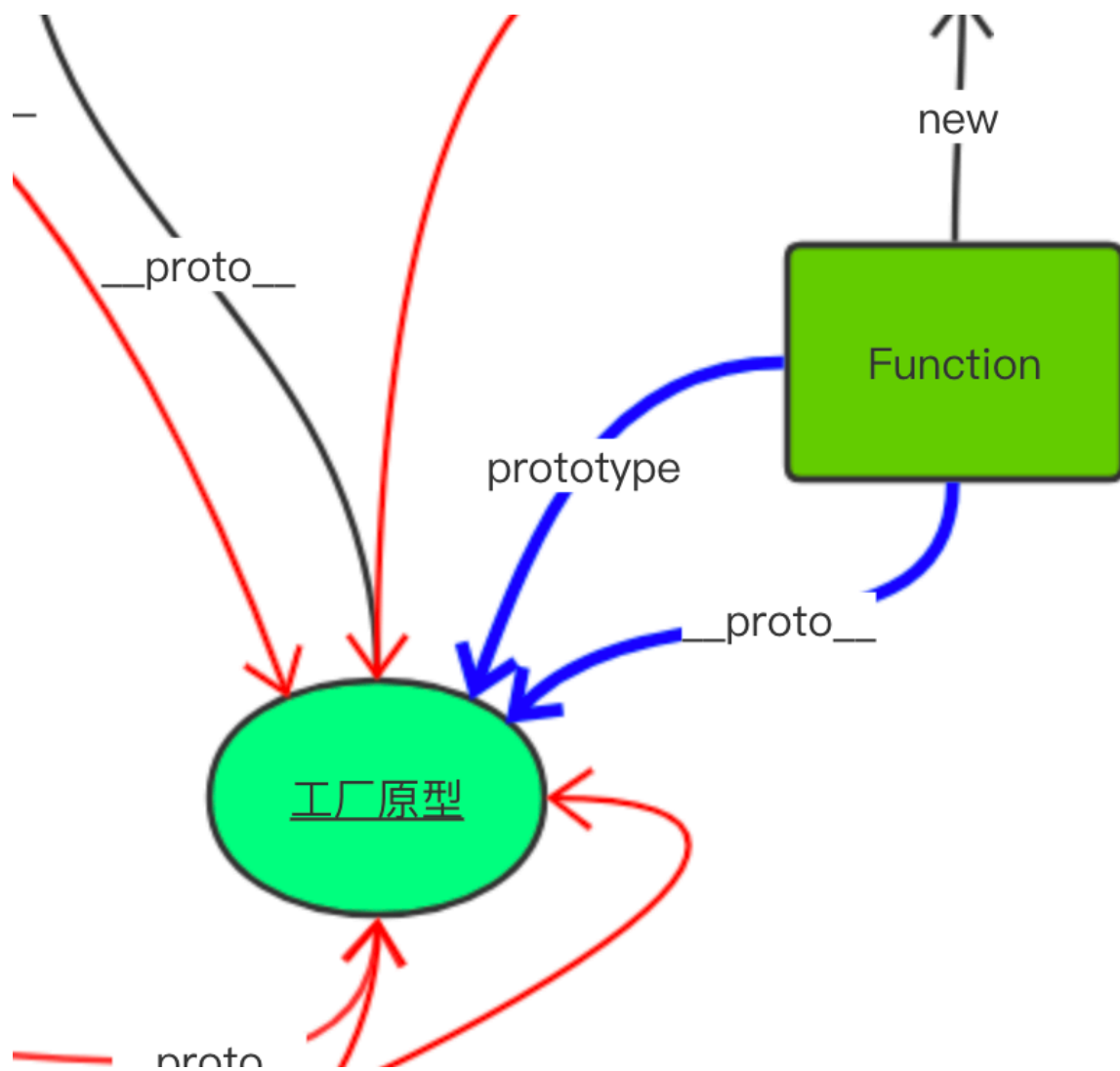
- 对象要分类，所以单
- 现在要有String, N
- 嗯，创造它们的原型
- 然后，有了它们的工
- 现在可以new Strir
- 但有更多我们自己需



需要工厂的 工厂 (Function)

现在可以new OtherFactory()了!





Function 也是工厂

Function.prototype == Function.prototype.__proto__