

极客大学前端进阶训练营

程劭非 (winter)

前手机淘宝前端负责人

重学前端

JavaScript

Types

- Number
- String
- Boolean
- Object
- Null
- Undefined
- Symbol

Object



三只一模一样的鱼，其实是三个对象。

Object



其中一只鱼发生了状态改变，失去了尾巴。

其它两只鱼并不受到影响。

因此，当我们在计算机中描述这三只鱼时，**必须把相同的数据存储三份。**

Object



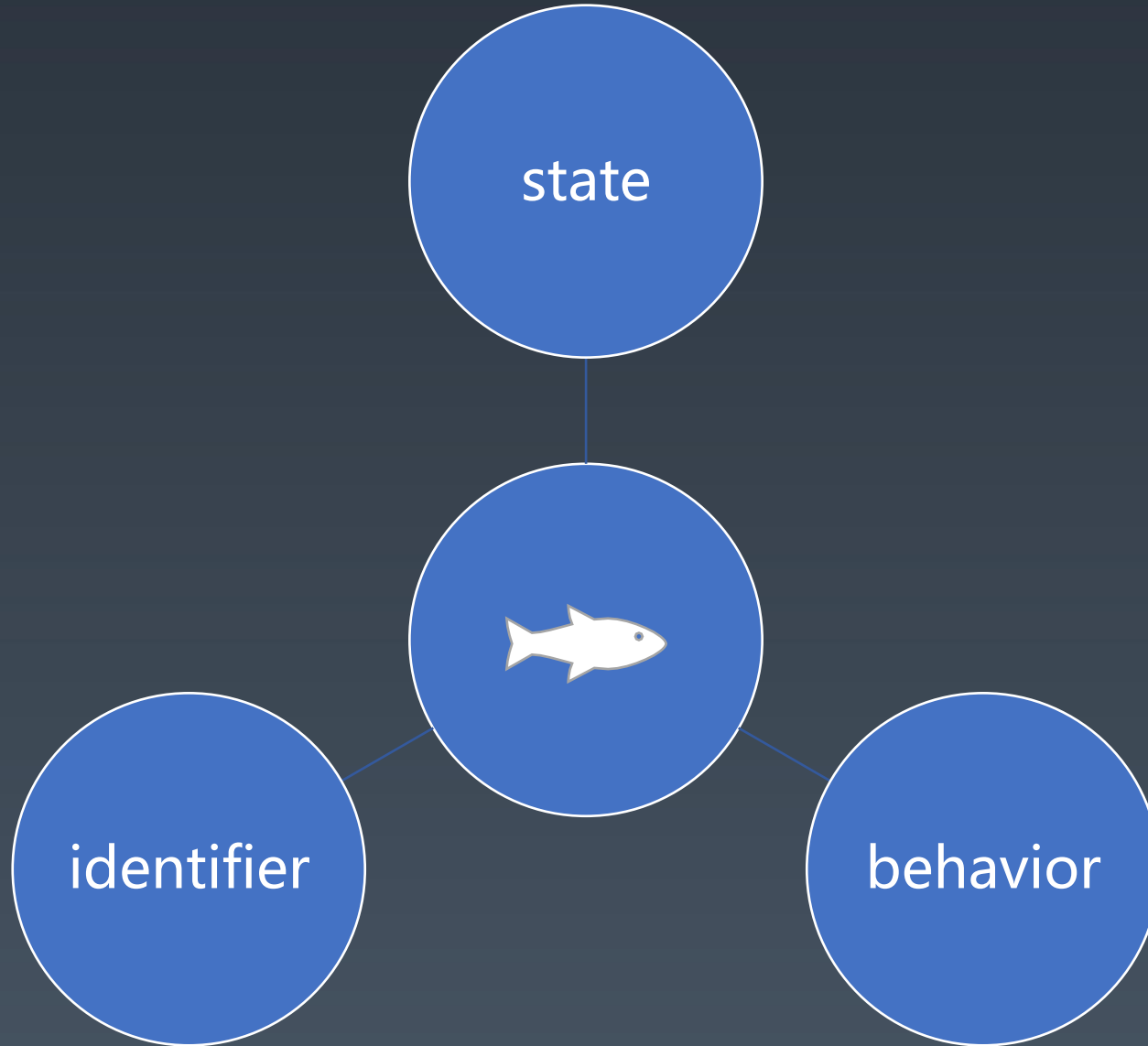
所以任何一个对象都是唯一的，这与它本身的状态无关。

所以，即使状态完全一致的两个对象，也并不相等。

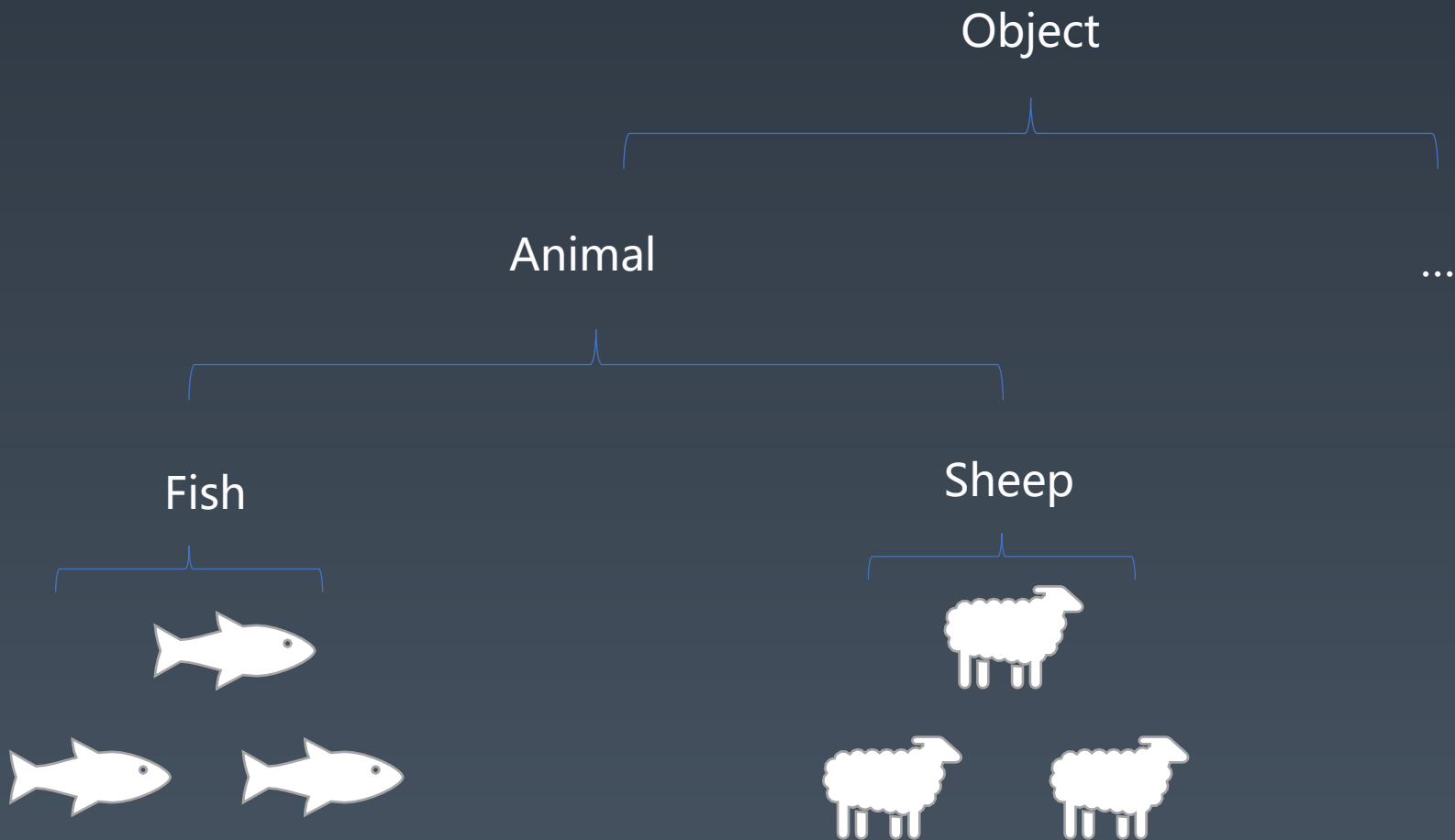
我们用状态来描述对象。

我们状态的改变即是行为。

Object



Object—Class



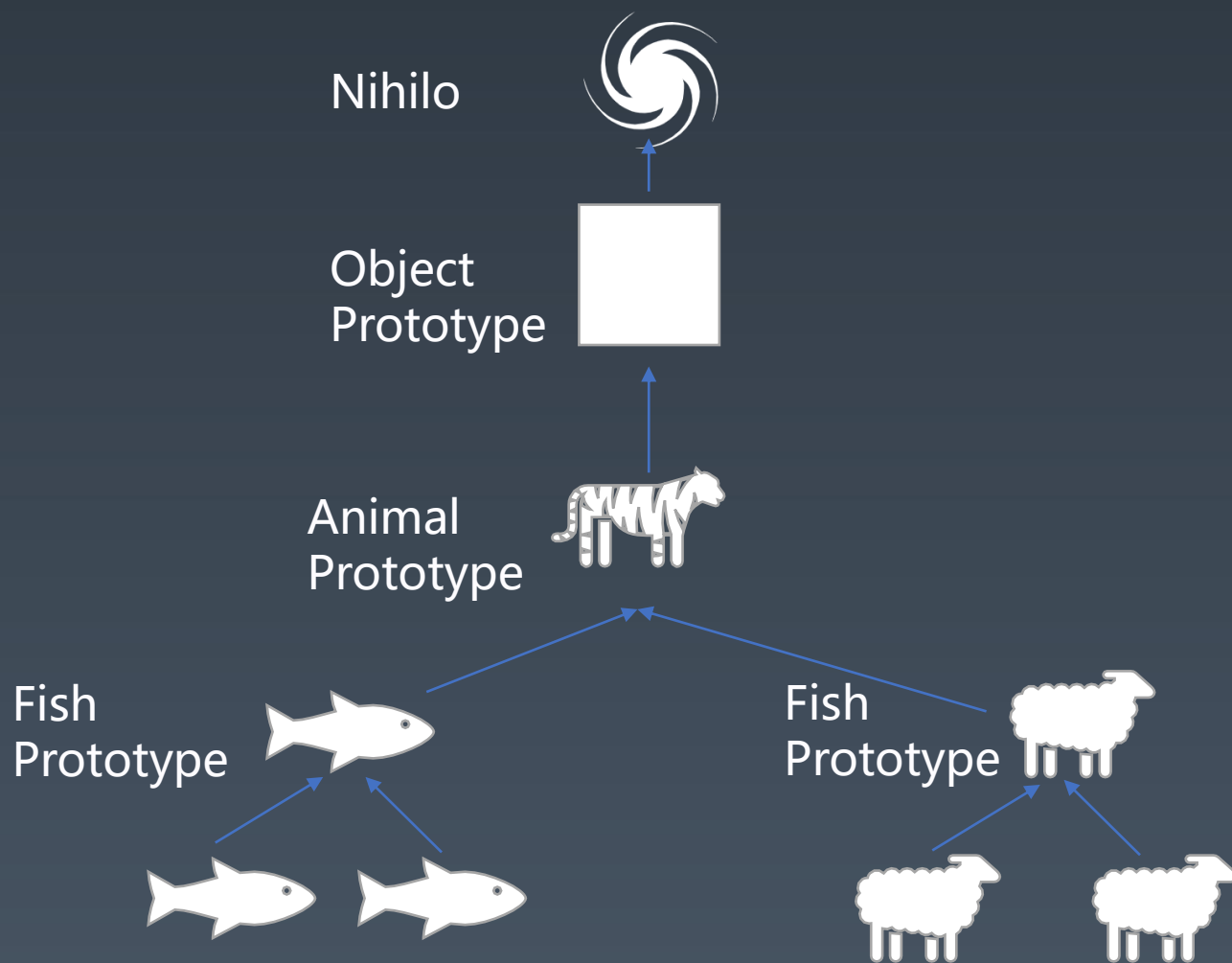
类是一种常见的描述对象的方式。

而“归类”和“分类”则是两个主要的流派。

对于“归类”方法而言，多继承是非常自然的事情。如C++

而采用分类思想的计算机语言，则是单继承结构。并且会有一个基类Object。

Object—Prototype



原型是一种更接近人类原始认知的描述对象的方法。

我们并不试图做严谨的分类，而是采用“相似”这样的方式去描述对象。

任何对象仅仅需要描述它自己与原型的区别即可。

Object Exercise

- 狗 咬 人
- “咬” 这个行为该如何使用对象抽象？

Object Exercise

```
class Dog {  
    bite(human) {  
        //.....  
    }  
}
```



Object Exercise

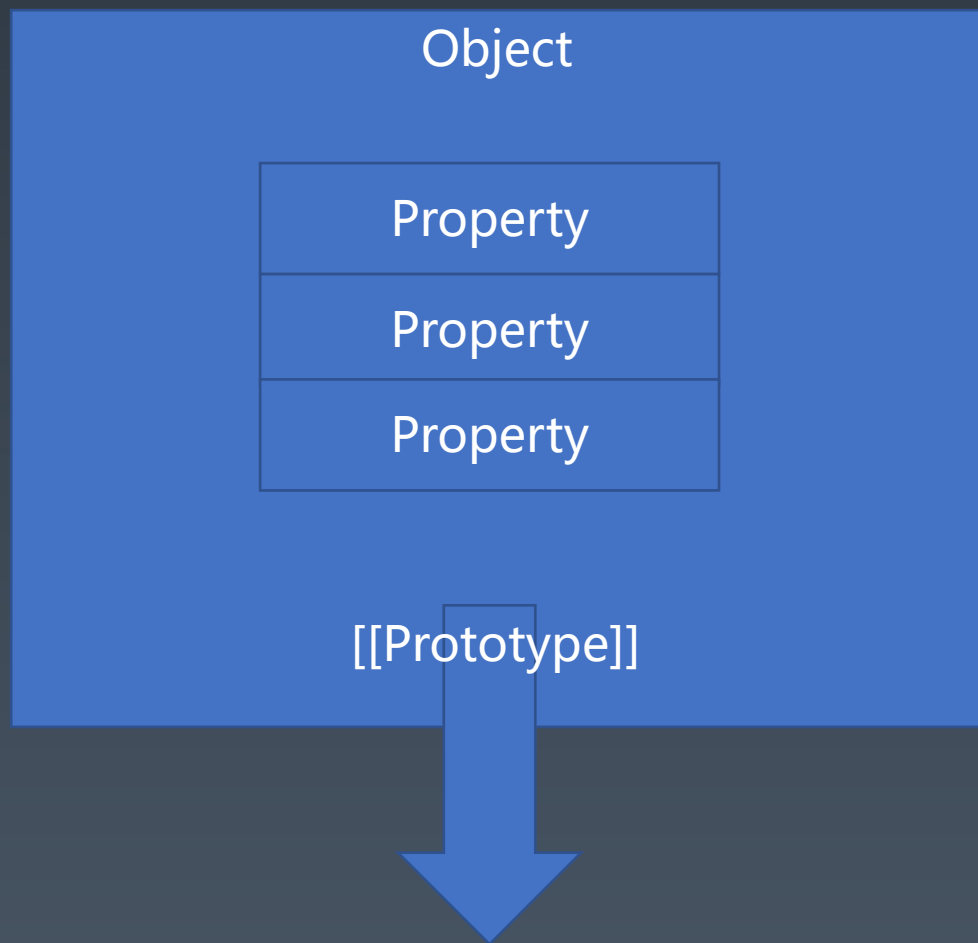
```
class Human {  
    hurt(damage) {  
        //.....  
    }  
}
```



我们不应该受到语言描述的干扰。

在设计对象的状态和行为时，我们总是遵循
“行为改变状态” 的原则。

Object in JavaScript

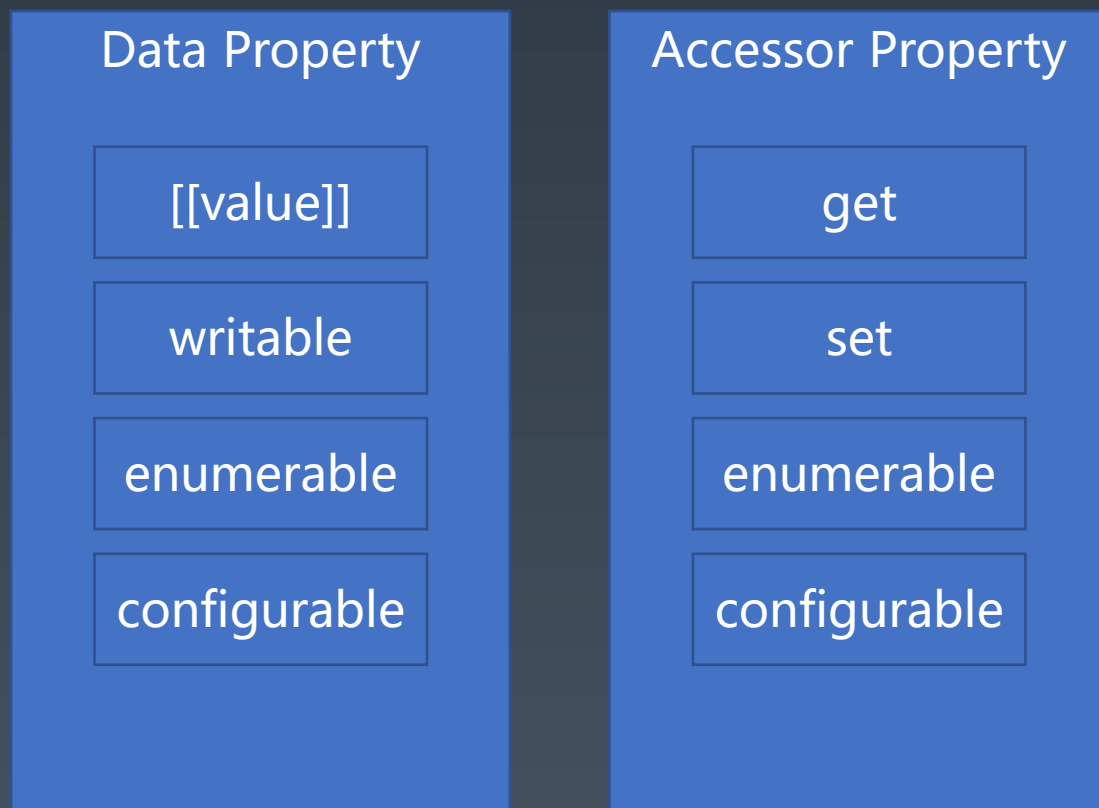


在JavaScript运行时，原生对象的描述方式非常简单，我们只需要关心原型和属性两个部分。

Object in JavaScript



Object in JavaScript

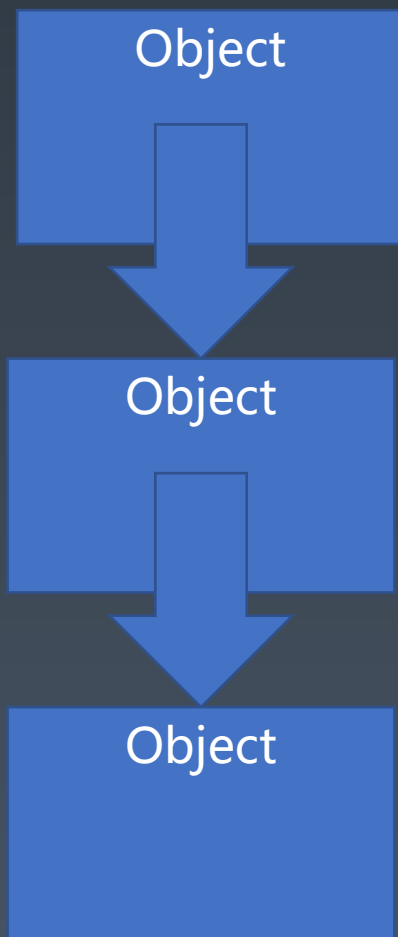


JavaScript用属性来统一抽象对象状态和行为。

一般来说，数据属性用于描述状态，访问器属性则用于描述行为。

数据属性中如果存储函数，也可以用于描述行为。

Object in JavaScript



当我们访问属性时，如果当前对象没有，则会沿着原型找原型对象是否有此名称的属性，而原型对象还可能有原型，因此，会有“原型链”这一说法。

这一算法保证了，每个对象只需要描述自己和原型的区别即可。

Object API/Grammar

- `{}` . `[]` `Object.defineProperty`
- `Object.create` / `Object.setPrototypeOf` / `Object.getPrototypeOf`
- `new` / `class` / `extends`
- `new` / `function` / `prototype`

Function Object

Object
[[call]]

前面讲述了JavaScript中的一般对象。

但JavaScript中还有一些特殊的对象，比如函数对象。

除了一般对象的属性和原型，函数对象还有一个行为[[call]]。

我们用JavaScript中的function 关键字、箭头运算符或者Function构造器创建的对象，会有[[call]]这个行为。

我们用类似 f() 这样的语法把对象当做函数调用时，会访问到[[call]]这个行为。

如果对应的对象没有[[call]]行为，则会报错。

Special Object

Array
[[length]]

Object.prototype
[[setPrototypeOf]]

.....

Host Object

Object
[[call]]
[[construct]]

```
foo() = 2;  
delete x;
```

THANKS! |  极客大学