构造函数和原型重名的属性

如果函数本身有这个属性，就用自己的，如果没有就用原型原型链上面的同名属性，如果原型链上还是没有，则返回 null。

```
function Foo() {
    Foo.a = function() { console.log(1) }
    this.a = function() { console.log(2) }
}

Foo.prototype.a = function() { console.log(3) }
Foo.a = function() { console.log(4) }

Foo.a();
let obj = new Foo();
obj.a();
Foo.a();
```

函数定义时，里面的具体内容不管它，把自己当作 JS 引擎去执行代码，函数在执行时才去理解函数内部的代码。

理解原型和原型链，如果函数本身有这个属性，就用自己的，如果没有就用原型原型链上面的同名属性。

1. 执行 `Foo.a()` 时，前面定义过 Foo 的 a 属性是一个函数，于是执行这个函数，输出 4。
2. 执行 `let obj = new Foo()`，Foo 的 a 属性被赋值为其他函数了。且 Foo 函数的实例对象（obj）的 a 属性赋值了一个函数。
3. 执行 `obj.a()` 时，前面已经个这个对象赋值了 a 属性，于是执行 `this.a` 函数，输出 2。
4. 执行 `Foo.a()` 时，第 2 步中说 Foo 的 a 属性被赋值为其他函数了，执行这个函数，输出 1。