# 原型模式

1. **基本概念**

### **1、定义：**

原型（Prortotype）模式的定义：用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

简单来说，原型模式就是从一个对象在创建另外一个可定制的对象，而且不需要知道任何创建的细节。

一般在需要大量复制某些内容的时候进行使用比较方便。

### 2、组成环节：

原型类（Prototype）：声明了一个克隆自身的接口，可以是具体原型类的父类。

具体原型类（Concrete Prototype）：原型类的子类，让一个原型克隆自身从而创建一个新的对象。

## 二、go语言中实现

##### 1、首先定义一个Prototype的原型类

type Prototype interface {  
 Clone() Prototype  
}

##### 2、定义一个ConcretePrototype具体对象类

type ConcretePrototype struct {  
 name string  
 age int  
}

##### 3、为具体对象类的clone方法进行实现

func (c ConcretePrototype) Clone() Prototype {  
 d := c  
 return d  
}

##### 4、通过main函数实现

func main() {  
 var c = &ConcretePrototype{name: "xiaoming", age: 18}  
 d := c.Clone()  
 ds, ok := d.(ConcretePrototype)  
 fmt.Println(ds.age, ds.name, ok)  
}

##### 5、输出结果

**18 xiaoming true**