# 策略模式

要实现某一个功能有多种方案可以选择，这些方案被隐藏在算法之后。 比如 一个压缩文件的程序，既可以选择 zip 算法，也可以选择 gzip 算法。 这些算法灵活多样，而且可以随意互相替换。这种解决方案就是策略模式。

策略模式的定义是：定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换。

## 应用场景

策略模式指的是定义一系列的算法，并且把它们封装起来。因此对于一种业务场景，流程是一样的，但是具体规则会随着某些因素的 不同而不同，我们就可以使用策略模式。，只要这些业务规则指向的目标一致，并且可以被替换使用，我们就可以 用策略模式来封装它们。

一个典型的例子就是表单校验。

## 实现

前端有一个人，是谁我就不说了，一直在鼓吹表编程，所谓的表编程其实本质上就是策略模式。

比如如下代码就是策略模式，也就是表编程：

const mapper = {  
 A() {  
 return 'you got A';  
 },  
 B() {  
 return 'you got B';  
 },  
 C() {  
 return 'you got C';  
 },  
 D() {  
 return 'you got D';  
 }  
}  
  
function getTextByLetter(letter) {  
 return mapper[letter]();  
}  
getTextByLetter('A'); // you got A  
getTextByLetter('C'); // you got C