```
def                    String    Double
    switch
        case "apples"     140
        case "origins"     223



def                  Double          String
Double =
    switch
        case "NY"             2
        case "FL"             3



val                  withTax _  Double  "NY"
val                  locallyTaxed price "apples"
```

部分施用(Partial application)

- 提前代入一部分参数值，使一个带有多个参数的函数得以省略部分参数，从而转换为一个参数数目较少的函数

- 让函数先作用于一些参数，经过部分的求解，结果返回一个由余下参数构成签名的函数

```
def price(product: String) : Double = {
    switch match {
        case "apples" => 140
        case "origins" => 223
    }
}

def withTax(cost: Double, state: String) :
Double = {
    switch match {
        case "NY" => cost * 2
        case "FL" => cost * 3
    }
}

val locallyTaxed = withTax(_: Double, "NY")
val costOfApples = locallyTaxed(price("apples"))
```

引用透明性(Referential Transparent)