Partial Functions n Currying

In [1]:
```
def modN(n:Int, x:Int) = ((x % n ) == 0)
```

```
defined function modN
```

In [2]:
```
modN(2,5)
```

```
res1: Boolean = false
```

In [3]:
```
modN(2,4)
```

```
res2: Boolean = true
```

In [3]:
```
modN(2)
```

```
Main.scala:25: not enough arguments for method modN: (n: Int, x: Int)B
oolean.
Unspecified value parameter x.
modN(2)
    ^
```

In [4]:
```
def modN(n:Int=2, x:Int) = ((x%n)== 0)
```

```
defined function modN
```

In [4]:
```
modN(5) //default arg can not be first arg, but last arg
```

```
Main.scala:25: not enough arguments for method modN: (n: Int, x: Int)B
oolean.
Unspecified value parameter x.
modN(5)
    ^
```

In [5]:
```
modN(x=5)
```

```
res4: Boolean = false
```

In [6]:
```
modN(x=4)
```

```
res5: Boolean = true
```

In [7]:
```
def modN(n:Int) (x:Int)= ((x%n) == 0)
```

```
defined function modN
```

In [8]:
```
modN(2)(5)
```

```
res7: Boolean = false
```

In [9]:
```
modN(2)(4)
```

```
res8: Boolean = true
```

In [9]:
```
modN(2)
```

```
Main.scala:25: missing arguments for method modN in class $user;
follow this method with `_' if you want to treat it as a partially app
lied function
modN(2)
    ^
```

In [10]:
```
List(4,5,6).map(modN(2))//default it is first arg first and go on ->posit
```

```
res9: List[Boolean] = List(true, false, true)
```

In [11]:
```
List(4,5,6).map(modN(n=2))
```

```
res10: List[Boolean] = List(true, false, true)
```

In [11]:
```
List(4,5,6).map(modN(x=2))
```

```
Main.scala:25: not found: value x
List(4,5,6).map(modN(x=2))
                     ^
```

In [12]:
```
List(4,5,6).map(modN(_)(x=2))
```

```
res11: List[Boolean] = List(false, false, false)
```

In [13]:
```
List(4,5,6).map(modN(_)(x=12))
```

```
res12: List[Boolean] = List(true, false, true)
```

In [ ]: