

## Week 1

Call by value first resolves the “values” before calling the function, while call by name first calls the function, gets the results and then resolves them.

## Week 2

**Take a function as a parameter:**

```
def sum(f: Int => Int, a: Int, b: Int): Int =  
  if a > b then 0  
  else f(a) + sum(f, a + 1, b)
```

**We can also generate functions using functions:**

```
def sum(f: Int => Int): (Int, Int) => Int =  
  def sumF(a: Int, b: Int): Int =  
    if a > b then 0  
    else f(a) + sumF(a + 1, b)  
  sumF
```

```
def sumInts = sum(x => x)  
def sumCubes = sum(x => x * x * x)  
def sumFactorials = sum(fact)  
sumCubes(1, 10) + sumFactorials(10, 20)
```

Generalization:

It's the same as creating a function that takes the n-1 arguments, and one takes the last one:

```
def f(ps1)...(psn-1) = (psn => E)
```

## Types

```
Type = SimpleType | FunctionType  
FunctionType = SimpleType '=>' Type  
| '( ' [ Types ] ') ' '= > ' Type  
SimpleType = Ident  
Types = Type { ' , ' Type }
```