SCALA

Types

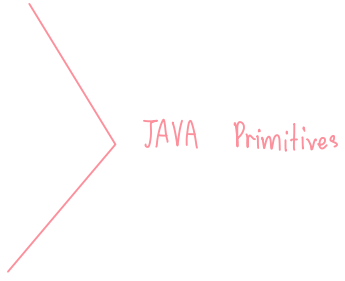
Boolean

Byte Short Int Long

Char

Float Double

String



Unit — Void

Null

Nothing (subtype of others (include "no")

Any

AnyRef (Any to Ref's type)

SCALA Basics

LATER

Functional Programming

Partially Applied

```
f = sum(1, 2, _)      <- will need to specify later
:
f(2)
```

Impure

SCALA variables is modifable within scope.

```
add = (x: Int) => {
    n = x+n        <- n is modifed
    n
}
```

Tail Recursion — Call function itself without extra work

```
retur x * fac(x-1)   <-->   ... fac(x, x*acc)
```

Not Tail — Tail

Higher Order
- Take Function as args
- Return Function

Currying — Functions as Return Value

```
add(x: Int): Int => Int = {
    return (y: Int) => x+y
}
```

With Special Syntax `sum(f: Int => Int)(a: Int, b: Int): Int =>` — pass f, where f is, return sum as