Working with Basic Types

Harit Himanshu
@harittweets

Overview

Introduction to Scala Types

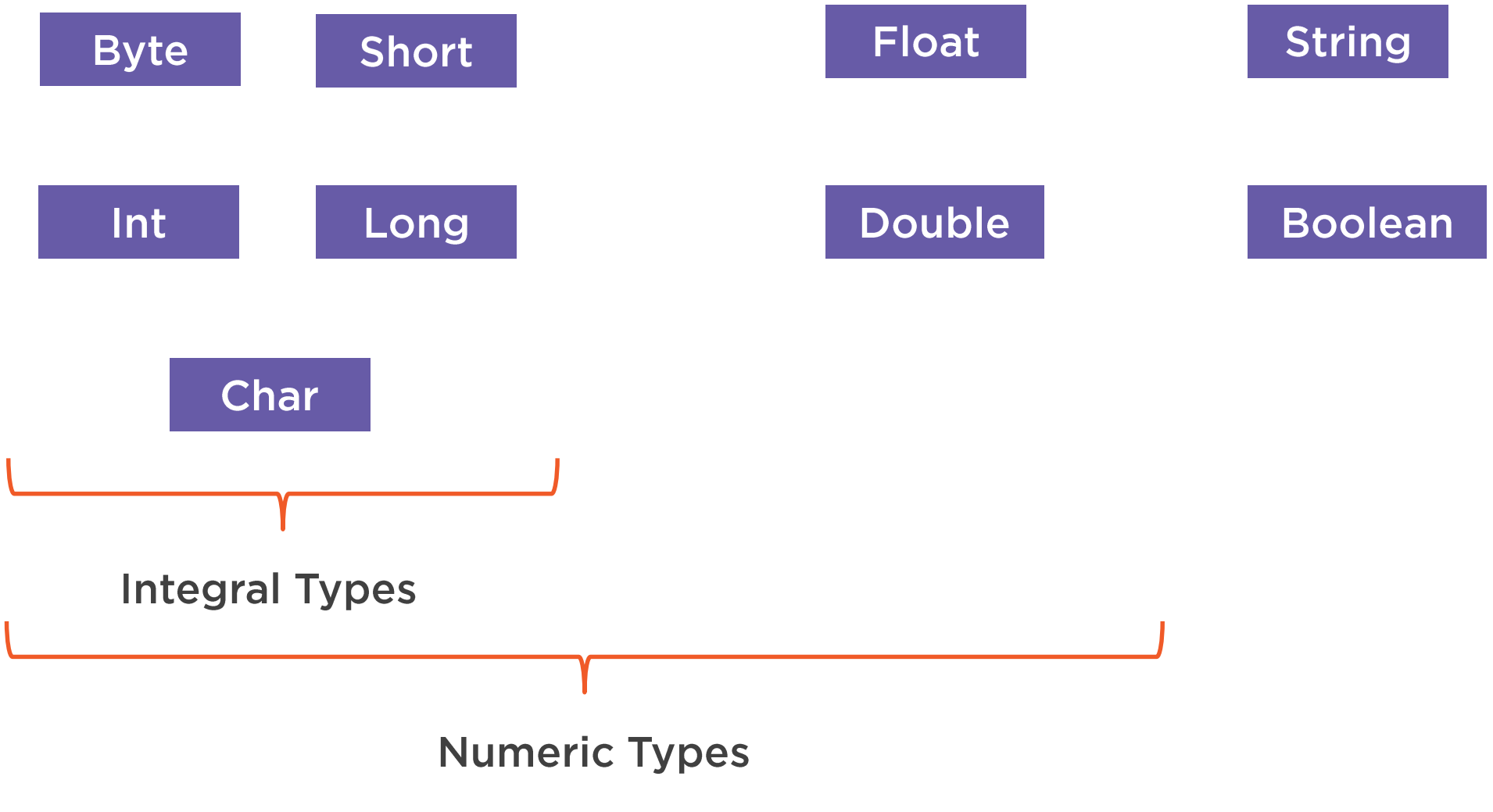
Embedding Scala Expressions in String Literals

Using Methods as Operators

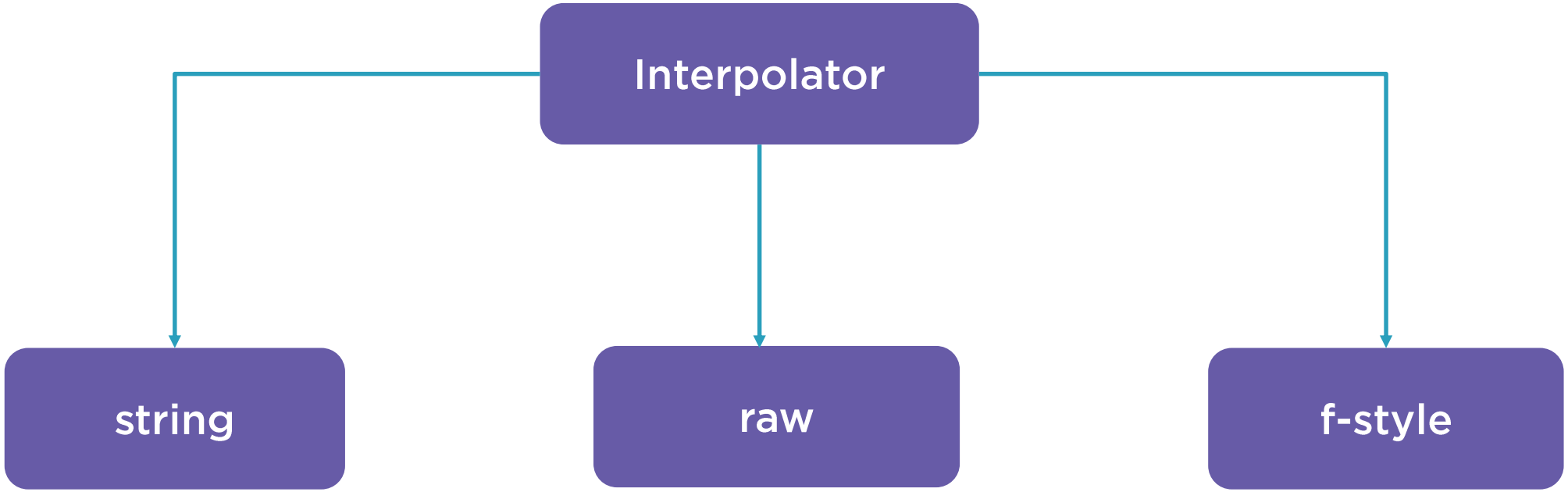
Understanding Scala Class Hierarchy

Project Demo

Fundamental Scala Types



Scala Interpolators



```
val v = 10.23
println(s"$v")
```

◀ string interpolator for variable

```
class object O(val v: Int)
val o = new O(10)
println(s"${o.v}")
```

◀ string interpolator for property access

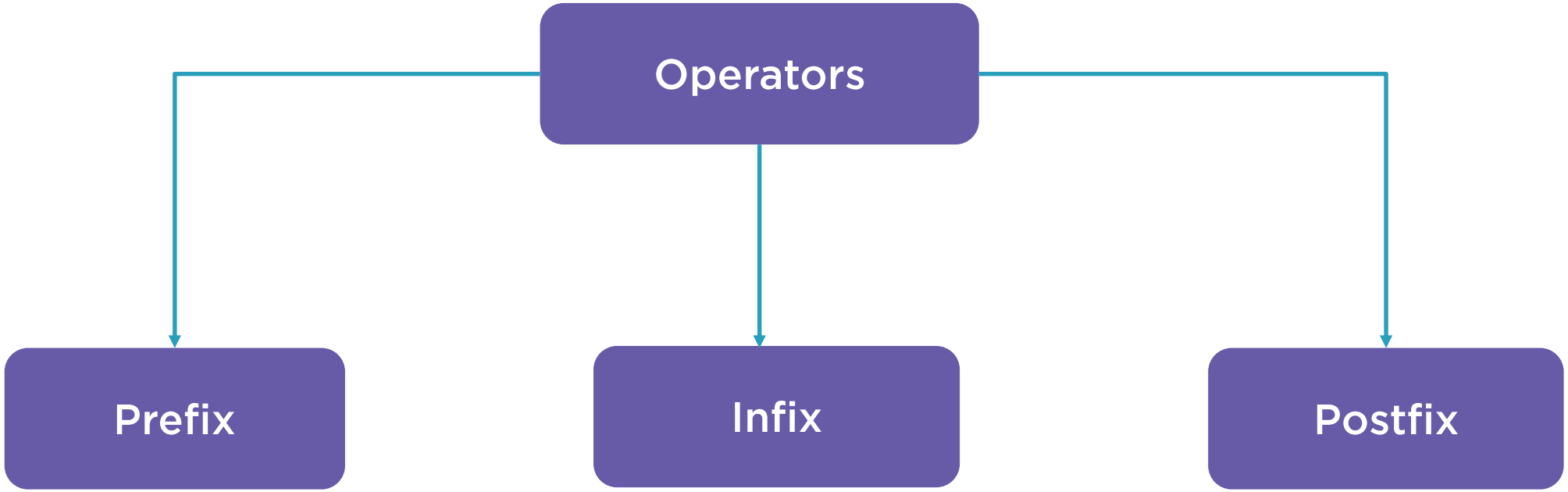
```
val v = "\\\\\\"
println(raw"$v")
```

◀ raw interpolator

```
println(f"${math.e}%.5f")
```

◀ f-style interpolator

Operators



! value

a + b

i ++

Creating your own prefix operator

def unary_<symbol>

+, - , !, ~

def unary_+

def unary_-

def unary_!

def unary_~

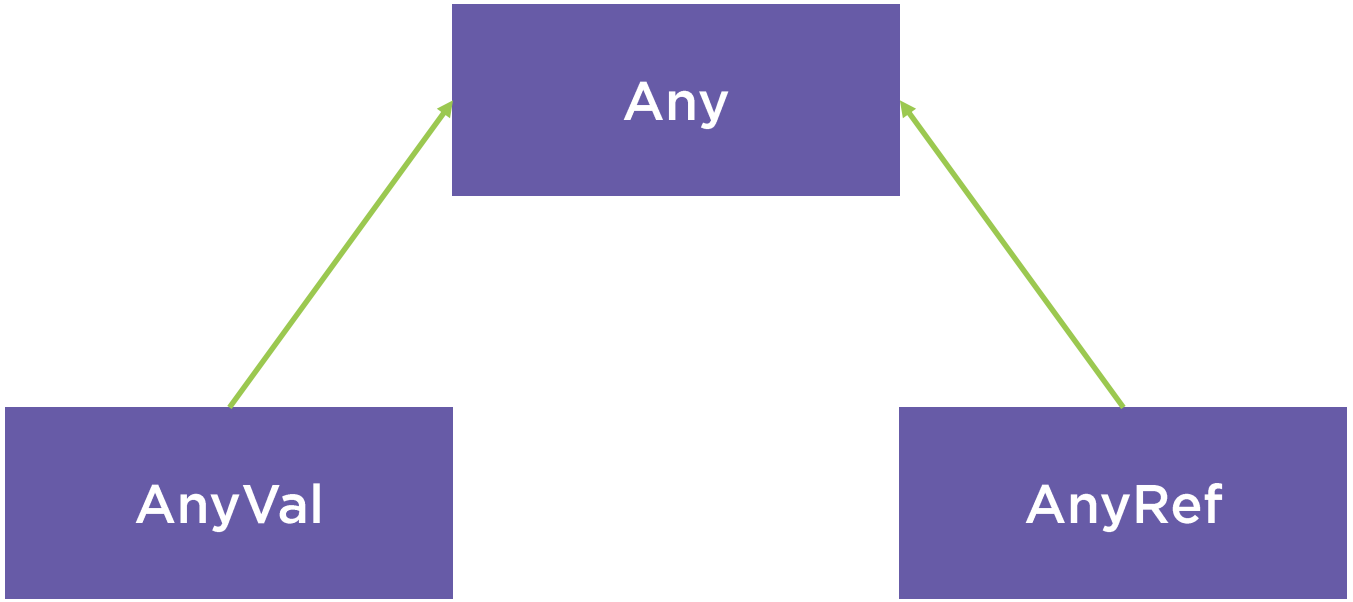
Scala Class Hierarchy

Any

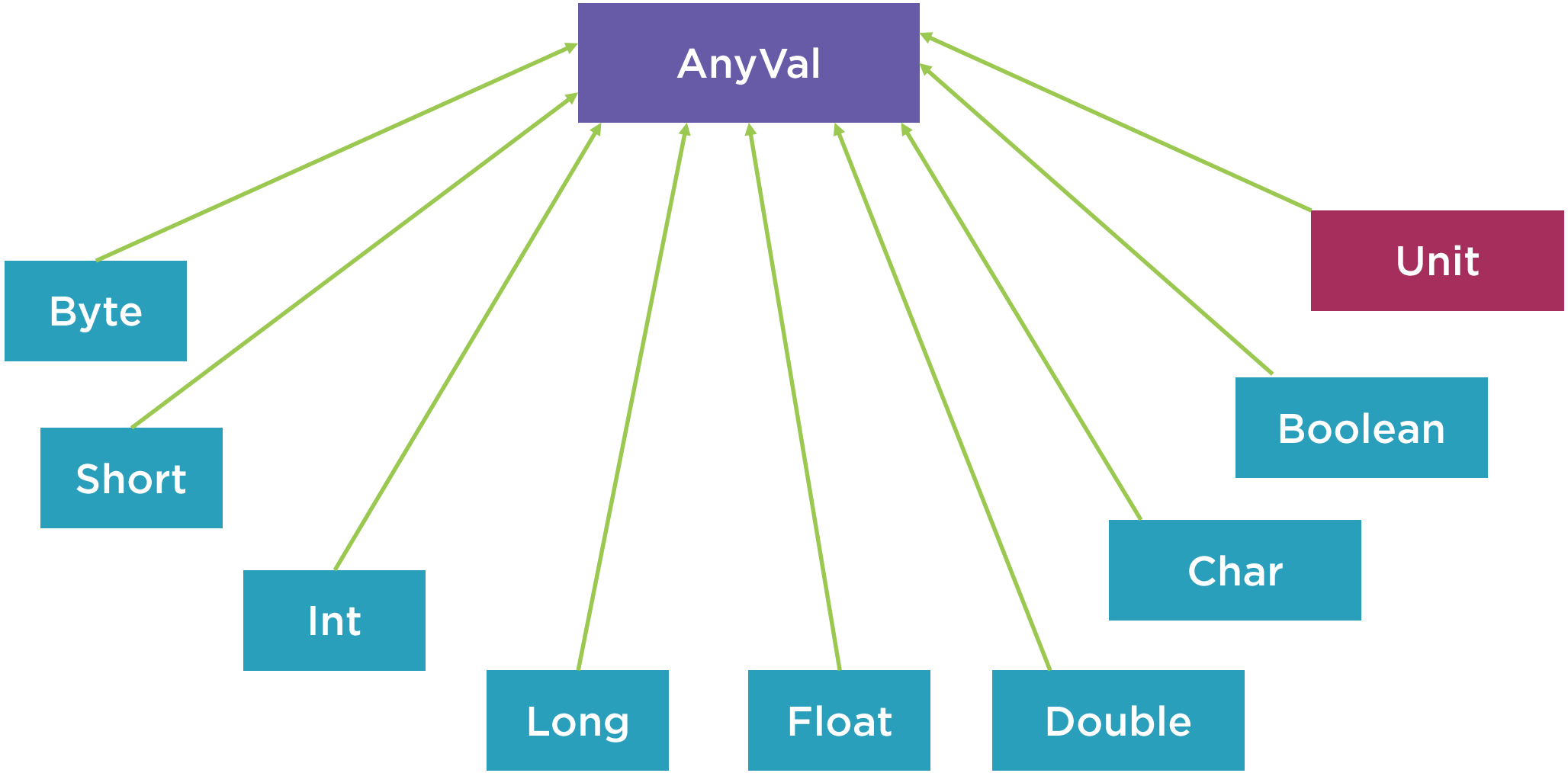
Universal Methods

```
final def ==(that: Any): Boolean
final def !=(that: Any): Boolean
def equals(that: Any): Boolean
def ##: Int
def hashCode: Int
def toString: String
```

Any's Subclasses



AnyVal Class



Scala Unit Type

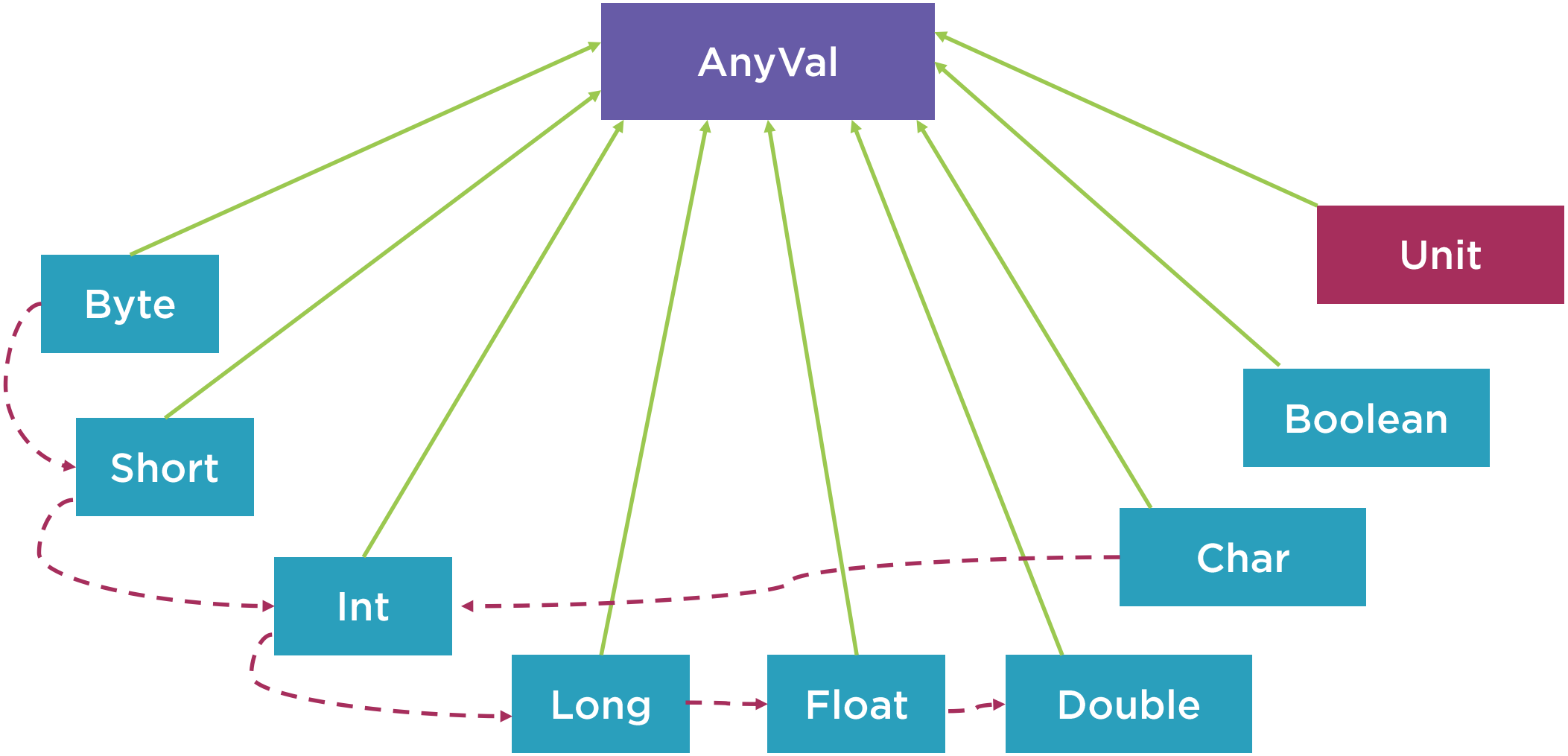
Unit

Does not return a result

Single instance with value ()

Represents the side-effect

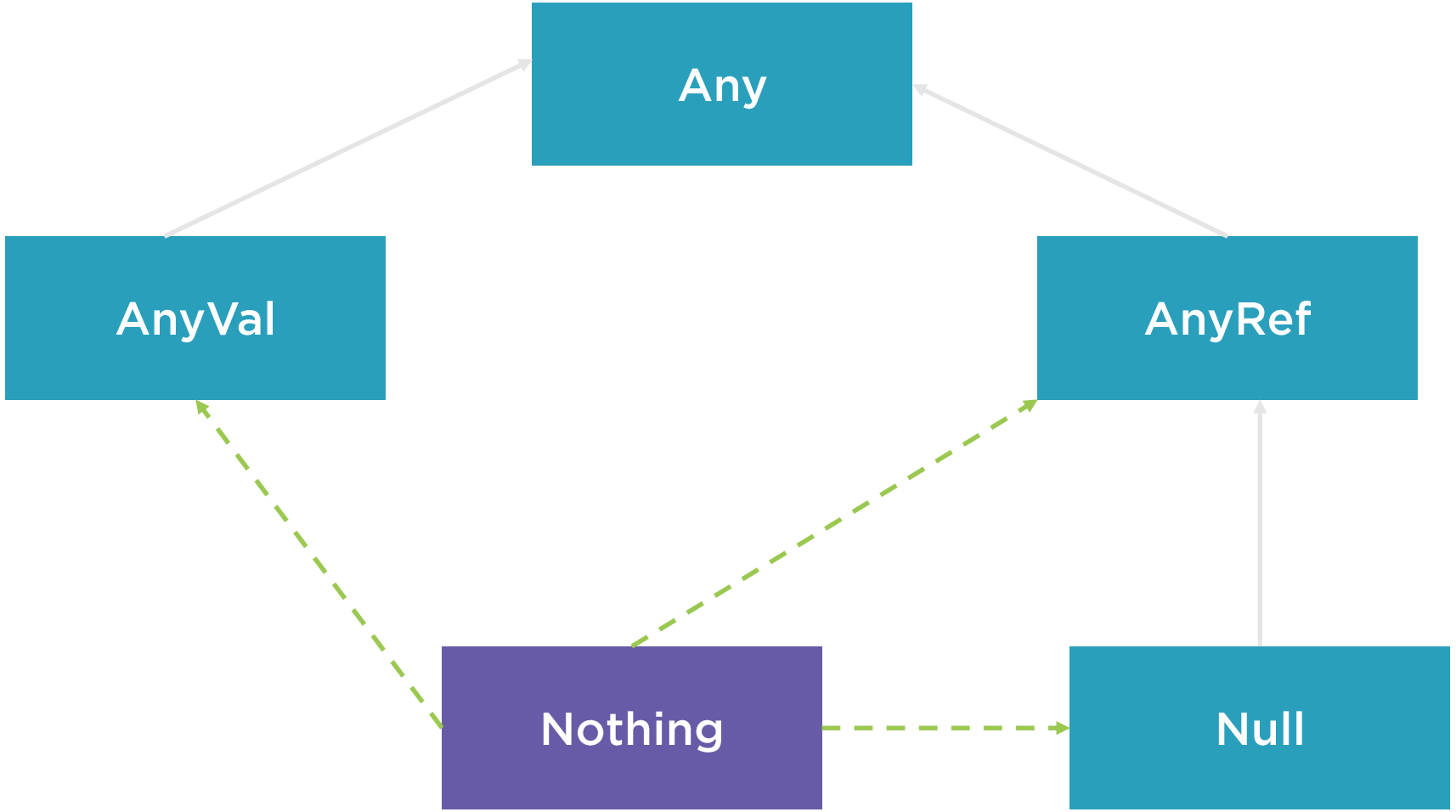
AnyVal with Implicit Conversions



AnyRef Class

Base class for all Classes in Scala

Scala Nothing Type

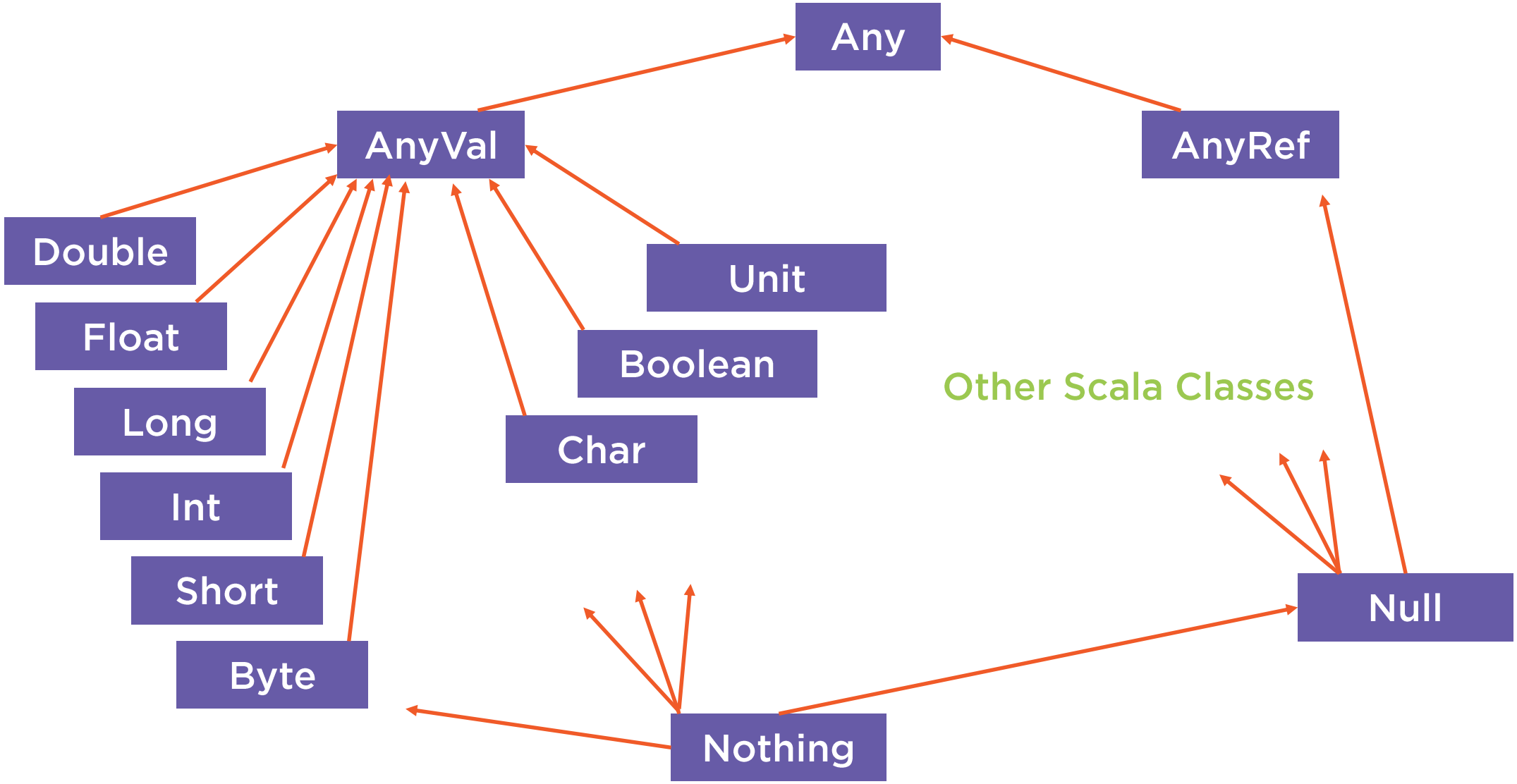


```
def error(messsage: String): Nothing = throw new RuntimeException(message)
```

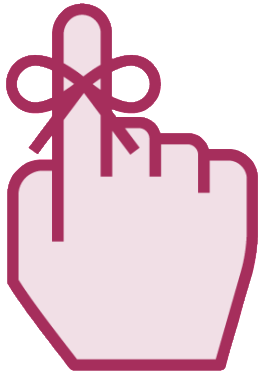
What's the Use Case for Nothing Type?

Signal abnormal termination

Scala Class Hierarchy



Creating Your Own Value Class



Rules to Remember

Subclass AnyVal Class

single val parameter representing underlying runtime representation

Can define defs

No vals, vars, nested traits, classes or objects

Demo

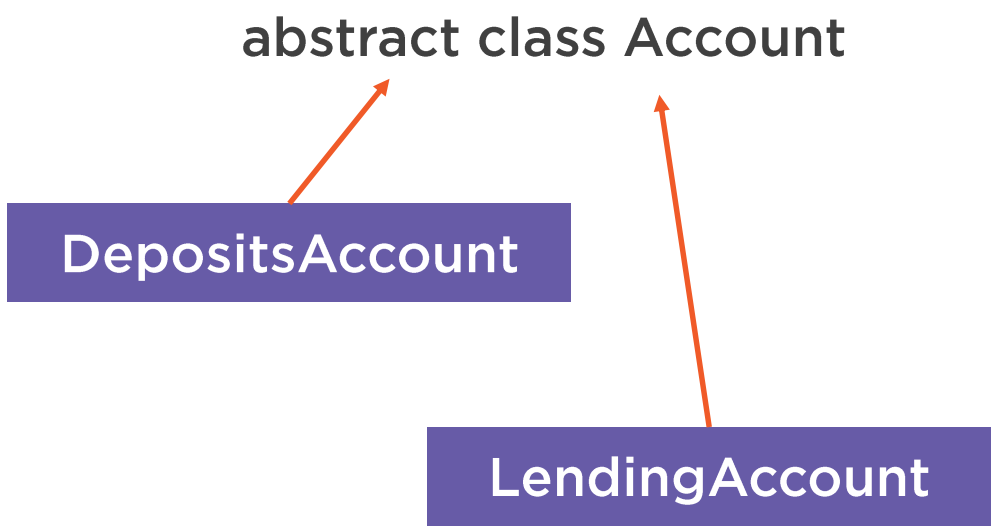
- Using *fields* instead of *class parameters*
- Creating the Account hierarchy
- Using Preconditions
- Creating the Email Type
- Creating the Dollars Value Class
- Revisiting the use of *Methods* as *Operators*
- Changing the *Application* entry point

Account Hierarchy Refactor

Before

abstract class Account

After



Summary

Introduction to Scala Types

Embedding Scala Expressions in String Literals

Using Methods as Operators

Understanding Scala Class Hierarchy

Project Demo