

Scala Cheat Sheet

@codecentric

Variables

```
var x      = "mutable"
val y      = "immutable"
lazy val z = "lazy"
```

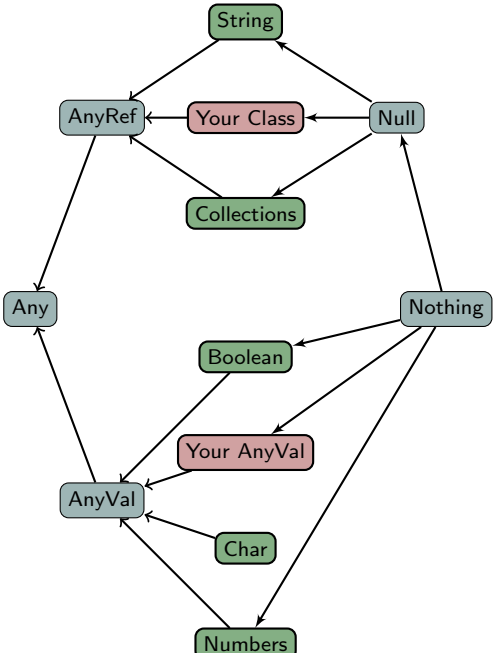
Methods

```
def id(arg: Char): Char = arg
def add(x: Int, y: Int): Int = x+y
def add(x: Int)(y: Int): Int = x+y
def twice[A](f: => A) = { f; f }
```

Function Types

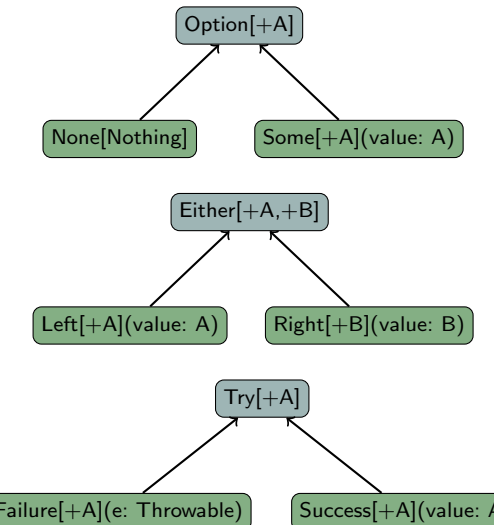
```
() => A      === Function0[A]
Unit => A    === Function1[Unit,A]
A => B      === Function1[A,B]
(A,B) => C   === Function2[A,B,C]
```

Types



```
graph TD
    String --> AnyRef
    String --> Null
    YourClass --> AnyRef
    YourClass --> Null
    Null --> AnyRef
    Null --> Collections
    Null --> Nothing
    AnyRef --> Any
    AnyRef --> AnyVal
    Collections --> AnyVal
    Nothing --> Boolean
    Nothing --> YourAnyVal
    Nothing --> Char
    Nothing --> Numbers
    Boolean --> AnyVal
    YourAnyVal --> AnyVal
    Char --> AnyVal
    Numbers --> AnyVal
```

Collections



```
graph BT
    None[None[Nothing]] --> Option[Option[+A]]
    Some[Some[+A](value: A)] --> Option
    Left[Left[+A](value: A)] --> Either[Either[+A,+B]]
    Right[Right[+B](value: B)] --> Either
    Failure[Failure[+A](e: Throwable)] --> Try[Try[+A]]
    Success[Success[+A](value: A)] --> Try
```

Either

- `getOrElse`
- `fold`
- ...

Either

- `getOrElse`
- `fold`
- ...

Either

- `getOrElse`
- `fold`
- ...

Either

- `getOrElse`
- `fold`
- ...

Either

- `getOrElse`
- `fold`
- ...

Try

- `getOrElse`
- `fold`
- ...

Futures

Timeouts

- `import scala.concurrent.duration._`
- `Await.result(future, 42.seconds)`

Implicits

- implicit conversions
- implicit arguments

Type Parameters

Type Hierarchy


Text 2

IntelliJ

Reformat

Extract Method

Rename



```
Ctrl + Alt + L
Ctrl + Alt + M
Shift + F6
```