

Functions



Function structure

Defining functions

```
def myFunction(x: Int, y: String): String = x + " " + y
```

parameter name parameter type return type
function name parameter body / implementation
 • an expression
 • can be a block

Calling functions

```
val result = myFunction(2, "hello scala")
```

function name actual parameter values
 this is an expression!

Takeaways

The Unit type

- used for side effects
- instruction-like functions

```
def aVoidFunction(x: Int): Unit
```

```
aVoidFunction(2)  
println("hello world")
```

prints to console

Nested functions

- in blocks of code
- scoping

Basic recursion

```
def isPrime(n: Int): Boolean = {  
    def isPrimeAux(t: Int): Boolean =  
        if (t <= 1) true  
        else (n % t != 0) && isPrimeAux(t - 1)  
  
    isPrimeAux(n / 2)  
}
```

Scala rocks

