The Scala Interpreter

# Install Scala.

* curl -s "https://get.sdkman.io" | bash
* source "$HOME/.sdkman/bin/sdkman-init.sh"
* sdk install scala
* scala -version

# Run "Hello World"

## Create a Scala File

* nano Hello.scala

## Scala Code

* object Hello {

def main(args: Array[String]): Unit = { println("Hello, Scala!")

}

}

## Save and Exit Nano

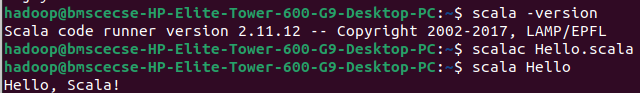
* Ctrl + O to write the file
* Enter to confirm the filename
* Ctrl + X to exit

## Compile the Scala Program

* scalac Hello.scala

## Run the Program

* scala Hello



## Experiment with Scala Basics

* scala

## Data types & Variables

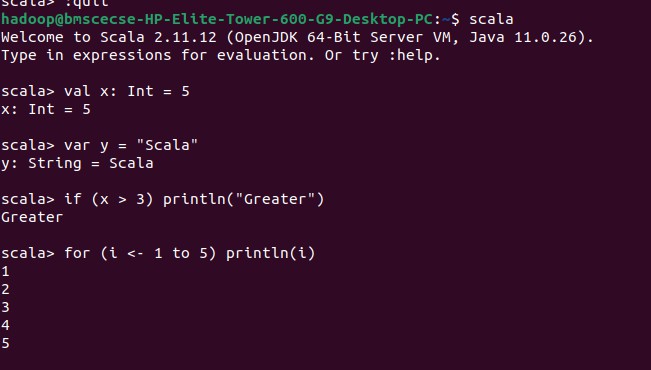
* val x: Int = 5
* var y = "Scala"

## Operators & Conditionals

* if (x > 3) println("Greater")

## Loops

* for (i <- 1 to 5) println(i)

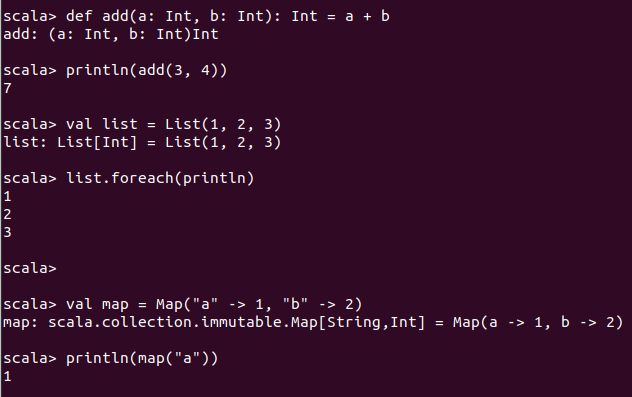


## Work with Functions

* + def add(a: Int, b: Int): Int = a + b
  + println(add(3, 4))

## Collections

* + val list = List(1, 2, 3)
  + list.foreach(println)
  + val map = Map("a" -> 1, "b" -> 2)
  + println(map("a"))



## Object-Oriented Programming

* + class Person(name: String) {

def greet() = println(s"Hello, $name")

}

* + val p = new Person("Alice") p.greet()

## Advanced Features

* **Traits**
  + trait Greeter { def greet(): Unit

}

* + class EnglishGreeter extends Greeter { def greet() = println("Hello")

}

* + val g = new EnglishGreeter()
  + g.greet()

## Pattern Matching

* + def describe(x: Any): String = x match { case 1 => "One"

case "two" => "Two"

case \_ => "Something else"

}

