

# Class 2 Homework: Hadoop Review Part 2, Scala

New York University

**Summer 2017**



# Homework: Explore Scala Numerics

---

## A. Readings

1. Please read Chapter 1 in the class text, "Learning Spark."

# Homework: Explore Scala-Spark Variables

---

## B. Explore Scala-Spark Variables

Complete the following steps and submit your code on NYU Classes together with screen shots showing the intermediate results as you interactively enter the Scala commands in the shell.

1. In a terminal window, start the Scala Spark Shell: `$ spark-shell`
2. Create an immutable variable named `exchangeRate` with explicit type `Double` and assign to it the value `0.88`.
3. Create an immutable variable named `dollars` with explicit type `Int` and assign to it the value `100.00`.
4. Correct step 3 to get rid of the error.
5. Create a mutable variable named `euros` with *implicit* type `Double` initialized to zero.
6. Assign to 'euros' the result of converting `dollars` to `euros` using `exchangeRate` as the conversion factor.
7. Assign to `dollars` a new value: `500`
8. Note the error in step 7. Fix the error in step 7 and set `dollars` to `500`.
9. Now set `dollars` to `500.00`. You should see an error because `dollars` expects an `Int`, not a `Double`.
10. Create a new mutable variable, `eurosInt`, of type `Int` and assign to it `0`.
11. Assign to `eurosInt` the result of converting `dollars` to euros using `exchangeRate`.
12. Use `toInt` with `exchangeRate` to remove the error in step 11.
13. What is the result in step 12? Is it a useful result? What happened?

# Homework: Explore Scala-Spark Computation

---

## C. Explore Scala-Spark Computation

In the previous exercise you worked with integers and doubles. Notice that using `toInt` may not give the expected result.

1. Try using `toInt` in a different way to achieve the desired `eurosInt` result of **440**.

2. Use `getClass` to verify the types of the three variables.

3. Output the result using the `println` command:

```
println("$" + dollars + " = " + eurosInt + " Euros")
```

4. Enter: `27/3.0` and note the result variable name, e.g. `res3`.

Use the result variable in an expression: `res3 * 2`

5. Assign the value `22.5` to `res3` - why didn't this work?

6. Import `scala.math.pow` and raise 2 to the third power.

7. Import `scala.math.sqrt` and take the `sqrt` (square root) of **64**.

# Homework: Explore Scala-Spark Strings

---

## D. Explore Scala-Spark Strings

Complete the following steps and submit your Scala code on NYU Classes together with screen shots showing the intermediate results as you interactively enter the Scala commands in the shell.

1. Create an immutable variable called `record` and assign to it the following string:

```
2017-01-08:10:00:00, 12345678-aaaa-1000-gggg-000111222333, 58, TRUE, enabled, disabled, 37.819722,-122.478611
```

2. Use `record.length` to determine the number of characters in `record`.

3. Use the `contains` method to search for the word "disabled" in `record`: `record.contains( "search term" )`

4. Use `indexOf` to find the index of the first occurrence of "16" in `record`.

5. Convert `record` to lower case using `toLowerCase` and then use chaining with `indexOf` to find the start of substring "true".

6. Verify that step 5. did not modify the variable named `record`.

7. Create a new variable called `record2` and assign to it the contents of `record`.

8. Test whether `record == record2`

9. Set `record2 = "no match"`

10. Test whether `record == record2`