Cheat sheet per il linguaggio Scala

Documentazione online

https://docs.scala-lang.org/
https://docs.scala-lang.org/cheatsheets/
https://jaxenter.com/cheat-sheet-complete-guide-scala-136558.html
http://overapi.com/static/cs/Scala_Cheatsheet.pdf

Import

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
```

Commenti

```
// come in C++, Java, ecc.
/* commento esteso
   su più linee */
```

Variabili e costanti

```
var x = "mutabile"   // attenzione alla 'r' in var
val y = "immutabile"
val z : BigInt = 43
```

Ennuple

```
val q = (10, 3.4, "ciao!")
println(q._1) // 10
```

Array

```
val arr = Array(5,10,20)
val lst = List(5,10,20) // immutabile
println((arr.length, arr.head, arr.last, arr(2)))
println((lst.length, lst.head, lst.last, lst(2)))
arr(0) = 4
lst(0) = 4 // ERRORE (lst è immutabile)
```

Stringhe

```
var s = "    Una stringa    "
s.length // 18
s.trim // "Una stringa"
val s2 = """
una stringa su più
righe"""
"rosso,nero,blu".split(",") //-> Array(rosso, nero, blu)
"ciao " ++ "mondo!" //-> "ciao mondo!"
"result = %d".format(40+2) //-> "result = 42"
val result = 40+2
s"la risposta è $result" //-> "la risposta è 42"
```

Operazioni su vettori

```
val arr = List(5,10,20,17)
arr.map(x=>x*2) //-> List(10, 20, 40, 34)
arr.map(x=>x*2).reduce((x,y)=>x+y) //-> 104
arr.map(_*2).reduce(_+_) // notazione abbreviata
arr.filter(_%2==0) //-> List(10, 20)
List(("uno", 1),("tre",3),("due",2)).maxBy(_._2) //-> ("tre",3)
```

Funzioni

```
def triplica(x:Int) : Int = x*3
triplica(6) // -> 18
def norm(x:Double, y:Double) : Double = {
      val r2 = x*x + y*y
      return Math.sqrt(r2)
}
norm(3,4) //-> 5.0
```

Strutture di controllo

```
for(i <- 1 to 100) { println(i) }
if (x % 2 == 0) { println("pari") }
else { println("dispari")  }
var x = 1; while (x < 100) { x *= 2 }
```

Pattern matching

```
val value : Any = (1,2)
val result = value match {
      case 0 => 1000
      case 1 => 1001
      case x:Int => x
      case (x:Int,y:Int) => x*x+y*y
      case _ => -1
} //-> 5
value match {
      case 0 => println("zero")
      case _ => println("Non zero")
} //-> stampa "Non zero"
Array((1,2),(3,4)).map { case (x,y) => x*x+y*y }
//-> Array(5,25)
var arr = Array(1,2,(3,5), "ok")
arr.map {
      case x:Int => x
      case (x:Int, y:Int) => x+y
      case _ => 0
} //-> Array(1, 2, 8, 0)
```

Conversioni

```
math.sqrt(10).toInt //-> Int = 3
"42".toInt //-> Int = 42
```

Operatori di concatenazione

```
"ciao," ++ " mondo!" //-> "Ciao, mondo!"
Array(1,2,3) ++ Array(4,5,6) //-> Array(1,2,3,4,5,6)
Array(Array (1,2,3), Array (4,5,6)).flatten
List(1,2,3) ::: List(4,5,6) // Solo con List
0 :: List(1,2,3) //->List(0,1,2,3); solo con List
```

Case class

```
case class Persona(name:String, age:Int)
val p = Persona("Silvia", 42) // Persona(Silvia, 42)
p.name //=> Silvia
p match { case Persona(x,y) => println(s"Ciao, $x!")}
// ->stampa "Ciao, Silvia!"
```

Metodo principale

```
object HelloWorld {
      def main(args: Array[String]): Unit = {
            println("Ciao mondo!")
      }
}
```