Dichiarazione variabili:

var stockPrice: Double = 100.0

Metodi

Method Declarations

Method *definitions*

- start with the def keyword,

* followed by optional argument lists,
* a colon character (:)
* and the return type of the method,
* an equals sign (=),
* and finally the method body. Methods are implicitly *declared* “abstract”
* NamedParameters nell invocazione
* Nested definitions

Type inference:

Il compilatore e’ intelligente abbastanza da riuscire ad intuire il tipo delle variabili.

When Explicit Type Annotations Are Required

In practical terms, you have to provide explicit type annotations for the following

situations:

1. A variable declaration, unless you assign a value to the variable (e.g., val name =

"Programming Scala")

2. All method parameters (e.g., def deposit(amount: Money)...)

3. Method return values in the following cases:

a. When you explicitly call return in a method (even at the end)

b. When a method is recursive

c. When a method is overloaded and one of the methods calls another; the

*calling* method needs a return type annotation

d. When the inferred return type would be more general than you intended, e.g.,

Any

**TUPLES:**

N variabili associate tra loro. Sorta di struct.

These literal “groupings” are instantiated as scala.TupleN instances, where N is the

number of items in the tuple.   
  
The Scala API defines separate TupleN classes for N between 1 and 22, inclusive.

Tuple instances are immutable, *first-class* values, so you can assign

them to variables, pass them as values, and return them from methods

Packages:

Non c’e’ relazione con paths.

Possibli annidate packages e classi nello stesso file.

Import:

Import singoli

Import tutto il package .\_

Import alcune classi del package {}

Import locali a metodi o blocchi.  
 Import parziali con alias.