

*Paola Celio | Pietro Corsi | Sergio Lins*

**PYTHON**

***BASICS***



## Lessons November

Date	Time	
November 2	16:00 – 18:00	Computer basic knowledge
November 5	18:00 – 20:00	Computer basic knowledge
November 9	16:00 - 19:00	Programming language introduction
November 12	18:00 – 20:00	Programming language introduction
November 16	16:00 - 19:00	Data Types, Variables, Conditions
November 19	18:00 – 20:00	Data Types, Variables, Conditions
November 23	16:00 - 19:00	Funzioni
November 26	18:00 – 20:00	Funzioni
November 30	16:00 – 19:00	



Lessons  
December

Date	Time
December 4	18:00 – 20:00
December 7	16:00 - 19:00
December 11	18:00 – 20:00
December 14	16:00 - 19:00
December 18	18:00 – 20:00
December 21	16:00 - 19:00

# *Lesson 6* – Data Types, Variables, Conditions

## INTRODUZIONE

La volta scorsa abbiamo iniziato a parlare dei tipi di variabile in Python (concetto che può essere opportunamente allargato alla programmazione in generale). Abbiamo anche detto che esistono delle parole riservate vorrei farvi un elenco in modo che non abbiate dubbi in futuro:

False, class, finally, is, return, None, continue, for, lambda, try, True, def, from, nonlocal, while, and, del, not, global, with, as, elif, if, or, yield, assert, else, import, pass, break, except, in raise.

Non occorre imparare a memoria questo elenco. Nella maggior parte degli ambienti di sviluppo, le parole chiave vengono evidenziate con un diverso colore; se cercate di usarne una come nome di variabile, ve ne accorgete subito.

# Lesson 6 – Data Types, Variables, Conditions

## INTRODUZIONE

Un'**espressione** è una combinazione di valori, variabili e operatori. Un valore è considerato già di per sé un'espressione, come pure una variabile, per cui quelle che seguono sono tutte delle espressioni valide (supponendo che alla variabile *n* sia già stato assegnato un valore):

```
>>> 42
```

```
42
```

```
>>> n
```

```
17
```

```
>>> n + 25 42
```

# Lesson 6 – Data Types, Variables, Conditions

## INTRODUZIONE

Quando scrivete un'espressione al prompt dei comandi, l'interprete la **valuta**, cioè trova il valore dell'espressione. Nell'esempio di prima,  $n$  ha valore 17 e  $n + 25$  ha valore 42. Un'**istruzione** è una porzione di codice che l'interprete Python può eseguire e che ha un qualche effetto, come creare una variabile o mostrare un valore.

```
>>> n = 17
```

```
>>> print(n)
```

La prima riga è un'istruzione di assegnazione che dà un valore alla variabile  $n$ . La seconda è un'istruzione di stampa che mostra a video il valore di  $n$ .

# *Lesson 6* – Data Types, Variables, Conditions

## INTRODUZIONE

Quando scrivete un'istruzione, l'interprete la **esegue**, cioè fa quello che l'istruzione dice di fare. In linea generale, le istruzioni, a differenza delle espressioni, non contengono valori.





# Lesson 6 – Data Types, Variables, Conditions

## INTRODUZIONE

Abbiamo poi parlato dei dati numerici ed illustrato le possibili operazioni vorrei ricordare che quasi tutte le operazioni base sono note (somma, sottrazione, moltiplicazione, divisione) per chi non è nell'ambito scientifico abbiamo introdotto ad esempio il concetto di resto di una operazione

```
>>> a = 5
```

```
>>> b = 2
```

```
>>> print(a%b)
```

```
1
```

Dove ovviamente % è l'operatore che ci dà il resto dell'operazione. Ma questo lo imparerete con la pratica dovete solo sapere che potete avere la parte intera ed il resto quando vi serve basterà cercarlo.

# *Lesson 6* – Data Types, Variables, Conditions

## INTRODUZIONE

Altri concetti sono stati introdotti

- `type()`, `isinstance()`, `isnumeric()` per ispezionare il tipo di dati di un valore e la sua adattabilità per l'uso o la conversione ad un certo tipo di dati numerici.
- Possiamo usare ad esempio operatori matematici per realizzare operazioni matematiche di base su dati numerici.
- E possiamo usare il tipo di dato `float` per rappresentare valori che includono valori frazionari che necessitano quindi di cifre dopo la virgola (nella notazione inglese la virgola è sostituita dal punto)

# *Lesson 6* – Data Types, Variables, Conditions

## INTRODUZIONE

```
print(type('7'))
```

```
print(type(7))
```

```
print(type(7.1))
```

```
print(isinstance('7', str))
```

```
print(isinstance(7, int))
```

```
print(isinstance(7.1, float))
```

```
print(isinstance(7, str))
```

```
print(isinstance('7', int))
```

```
print(isinstance('7.1', float))
```



# Lesson 6 – Data Types, Variables, Conditions

## INTRODUZIONE

E' fondamentale capire che il tipo di dato è parte del valore. Il tipo di dato non è parte della variabile che vorreste utilizzare per accedere al valore. Una variabile può accedere ad ogni valore indipendentemente dal tipo di dato del valore. Per essere più chiari scrivete il seguente codice:

```
>>> x = 'a string'  
>>> print(type(x))  
>>> x = 7  
>>> print(type(x))  
>>> x = False  
>>> print(type(x))
```



```
<class 'str'>  
<class 'int'>  
<class 'bool'>
```

# Lesson 6 – Data Types, Variables, Conditions

## INTRODUZIONE

Cosa avete imparato nella lezione scorsa:

- La funzione `type()` restituisce il tipo di dato di un valore specifico.
- La funzione `isinstance()` vi permette di controllare se un valore è un'istanza di uno specifico tipo di dato.
- Il tipo di dato `float` è per valori numerici che contengono valori frazionari che sono rappresentati come numeri dopo la "virgola" decimale.
- I valori hanno tipi di dati ma le variabili no. Una variabile punta banalmente ad un valore e voi potete puntare ad ogni valore di ogni tipo di dato.

# Lesson 6 – Data Types, Variables, Conditions

Python installation: <https://opensource.com/article/17/10/python-101>

[https://raw.githubusercontent.com/AllenDowney/ThinkPythonItalian/master/thinkpython\\_italian.pdf](https://raw.githubusercontent.com/AllenDowney/ThinkPythonItalian/master/thinkpython_italian.pdf)

[https://www.thinkwithgoogle.com/intl/it-it/strategie/ricerca-google/tendenze-ricerca-retail/?gclid=Cj0KCQiA48j9BRC-ARIsAMQu3WQtKdayyfJDhuz03leAdWwOQ7m0tmuidhH4wejb9P0a0Jlb19TuJ5QaAlaiEALw\\_wcB](https://www.thinkwithgoogle.com/intl/it-it/strategie/ricerca-google/tendenze-ricerca-retail/?gclid=Cj0KCQiA48j9BRC-ARIsAMQu3WQtKdayyfJDhuz03leAdWwOQ7m0tmuidhH4wejb9P0a0Jlb19TuJ5QaAlaiEALw_wcB)

<https://docs.microsoft.com/en-us/learn/modules/python-datatypes-numeric-operations/3-exercise-isnumeric>