MODULI, MATH, RANDOM

[INTRODUZIONE](#_heading=h.uhaabotfee8j)

[TERMINI CHIAVE](#_heading=h.tyjcwt)

[COSA FA IL CODICE? [10 minuti]](#_heading=h.3dy6vkm)

[COSA FA IL CODICE? [10 minuti]](#_heading=h.zi7lqpus96ni)

[DOMANDE [10 minuti]](#_heading=h.e6yqhulijc8j)

[TROVA GLI ERRORI [5 minuti]](#_heading=h.4zojj7y8ty94)

[ESERCIZI PRATICI](#_heading=h.3j2qqm3)

# INTRODUZIONE

I **moduli**, conosciuti anche come **librerie**, sono dei file che contengono un insieme di funzioni aggiuntive implementate da altri (o da noi stessi) che possiamo importare (e quindi poi utilizzare) nel nostro codice. Python include già una lista di moduli standard (anche conosciuti *come standard library*), ma è anche possibile scaricarne o definirne di nuovi.

In questa lezione vedremo i moduli *math* e *random.*

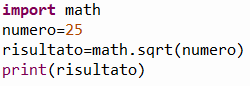
# TERMINI CHIAVE

|  |  |  |
| --- | --- | --- |
| **COMANDO / FUNZIONE / TERMINE** | **SIGNIFICATO** | **ESEMPIO** |
| import *libreria* | **comando** per **importare** librerie del linguaggio. | **import** math |
| math | **modulo** contenente funzioni matematiche |  |
| sqrt(x) | **funzione** per calcolare la radice quadrata di x | math.**sqrt**(25)  => 5 |
| numero pseudocasuale | numero prodotto da un algoritmo che simula la generazione di un numero casuale |  |
| random | **modulo** per la generazione di numeri pseudocasuali |  |
| random() | **funzione** che genera un numero pseudocasuale decimale compreso tra 0 e 1 | random.random()  => 0.701728341232 |
| randint(a,b) | **funzione** che genera un numero pseudocasuale intero compreso tra a e b | random.randint(2,100)  => 56 |
| Altre funzioni matematiche utili (predefinite, non necessitano dell’importazione di un modulo): | | |
| round(x,n) | **funzione** per arrotondare x ad n cifre decimali | **round**(12.43562,2)  => 12.44 |
| abs(x) | **funzione** che ritorna il valore assoluto di x | **abs**(-5)  => 5 |

# COSA FA IL CODICE? [10 minuti]

Lavoro in coppia

Di seguito c’è del codice scritto in Python:



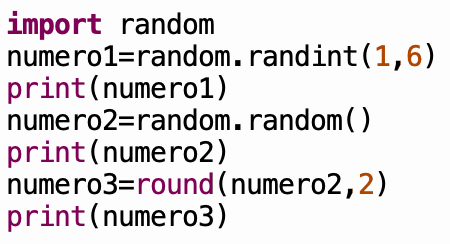
|  |
| --- |
| Secondo voi cosa fa il codice quando viene eseguito? |
| Fa la radice quadrata del numero 25 |
| Ora scrivete il codice in VSCode (salvate con nome *15\_Math.py*) e controllate se si comporta come previsto. In caso contrario, descrivere cosa accade di diverso. |
|  |

# 

# COSA FA IL CODICE? [10 minuti]

Lavoro in coppia

Di seguito c’è del codice scritto in Python:



|  |
| --- |
| Secondo voi cosa fa il codice quando viene eseguito? |
| “random.randint” genera un numero casuale tra 1 e 6, “random.random” genera un numero casuale decimale tra 1 e 0, “round” arrotonda il risultato della variabile “numero2” |
| Ora scrivete il codice in VSCode (salvate con nome *16\_Random.py*) e controllate se si comporta come previsto. In caso contrario descrivete cosa accade di diverso. |
|  |

# DOMANDE [10 minuti]

Lavoro in coppia

* Nei due esempi precedenti quanti e quali moduli abbiamo visto?
* Qual è la sintassi per richiamare una funzione all’interno di un modulo? Fai un esempio.
* A cosa serve la funzione round()? Fa parte di un modulo oppure è una funzione predefinita?
* Che differenza c’è tra la funzione random() e la funzione randint()?

# TROVA GLI ERRORI [5 minuti]

Lavoro in coppia

|  |
| --- |
| In questo programma ci sono 4 errori, riesci ad individuarli? Evidenziali con un colore. |
| import random  import math  n1=random.randint(10,40,2)  n2=math.sqrt(36  n3=random()  risultato=n1+n2+n3  risultato\_arrotondato=random.round(risultato,3)  print(risultato\_arrotondato) |

# 

# 

# ESERCIZI PRATICI

1. Scrivi un programma che stampi un numero (decimale) compreso tra 0 e 10. *[17\_random0\_10.py]*
2. Fai ora in modo che il numero random (decimale) sia compreso tra 0 e 100. *[18\_random0\_100.py]*
3. Scrivi un programma che simula il lancio di un dado a 20 facce (i numeri che possono uscire vanno da 1 a 20). *[19\_dado\_20\_facce.py]*

PS: i dadi a 20 facce sono molto usati nei giochi di ruolo!

1. Scrivi un programma che chieda all’utente l’area di un quadrato e stampi il lato arrotondato alla prima cifra decimale. *[20\_lato\_quadrato.py]*
2. Scrivi un programma che calcoli la radice quadrata di un numero intero pseudocasuale compreso fra 10 e 1000 e stampi il risultato arrotondato alla terza cifra decimale. *[21\_radice\_pseudocasuale.py]*