

PYTHON BOOTCAMP

Lezione 1



INFO

- ▶ 16/06 dalle 9.00 alle 13.00
- ▶ 17/06 dalle 9.00 alle 13.00
- ▶ 18/06 dalle 9.00 alle 13.00
- ▶ 19/06 dalle 9.00 alle 13.00
- ▶ Indirizzo email: pietro.musoni@unipr.it

COME ACCEDERE AI PC

Username: da **vpipoli01** a **vpipoli20** (in base al numero della postazione)

Password: **vpipoli**

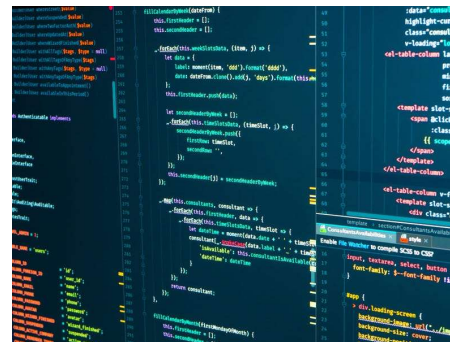
Dominio: **UNIPR**

Pool: **Aula Informatica**

PROVATE AD ACCEDERE

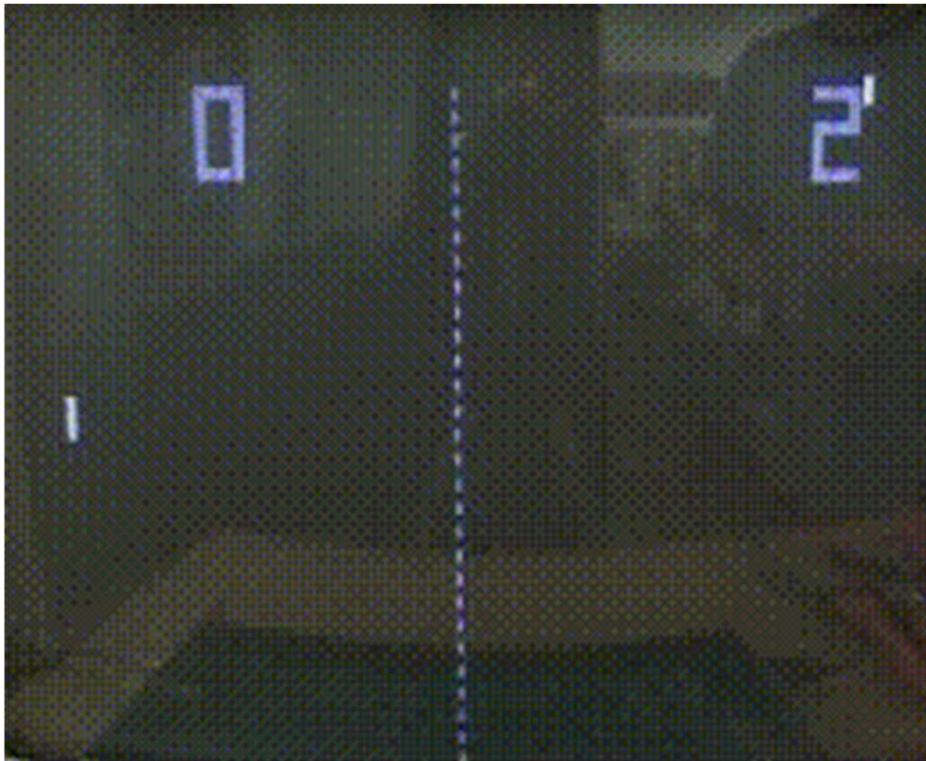
OBIETTIVI

- ▶ Imparare a studiare un problema e sviluppare un algoritmo
- ▶ Tradurre un algoritmo in un programma in python
- ▶ Imparare i costrutti base di python
- ▶ ...



OBIETTIVI

Creare il gioco Pong da zero!



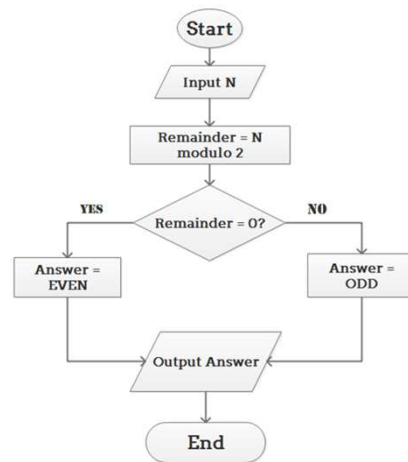
Cosa vuol dire programmare?

- ▶ Si parte con un problema
- ▶ Devo risolvere il problema in modo automatico: uso un programma
- ▶ Per scrivere un programma devo prima scomporre il problema in sottoproblemi più semplici.
- ▶ Traduco la mia soluzione con un linguaggio comprensibile dal computer!



RISOLVERE UN PROBLEMA

Algoritmo



Division

Algorithm 2: Division

```
1 function divide (x, y);  
   Input: Two  $n$ -bit integers  $x$  and  $y$ , where  $y \geq 1$   
   Output: The quotient and remainder of  $x$  divided by  $y$   
2 if  $x = 0$  then  
3   return  $(q, r) = (0, 0)$   
4 else  
5   set  $(q, r) = \text{divide}(\lfloor \frac{x}{2} \rfloor, y)$ ;  
6    $q = 2 \times q, r = 2 \times r$ ;  
7   if  $x$  is odd then  
8      $r = r + 1$   
9   end  
10  if  $r \geq y$  then  
11     $r = r - y, q = q + 1$   
12  end  
13  return  $(q, r)$   
14 end
```

- Come una ricetta, una sequenza ordinata di istruzioni per raggiungere uno scopo
- Devo scomporre un procedimento complesso in passaggi elementari

RISOLVERE UN PROBLEMA

Algoritmo

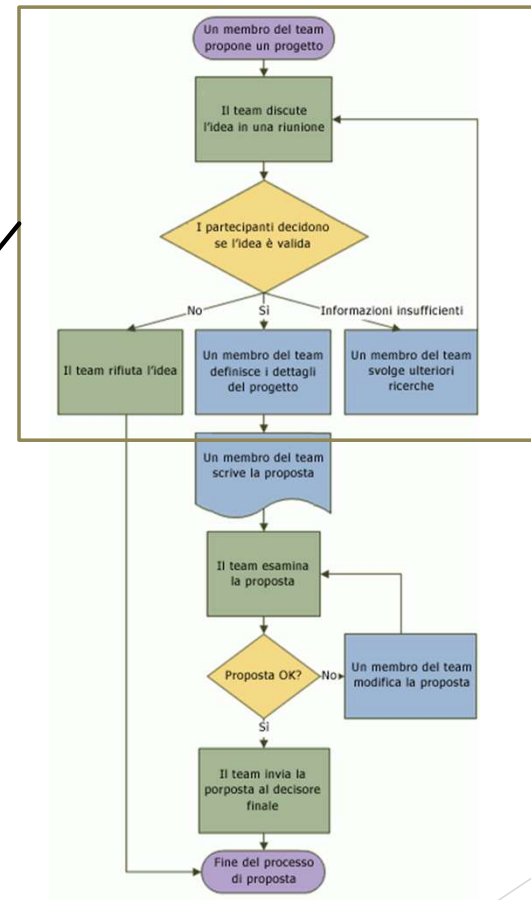
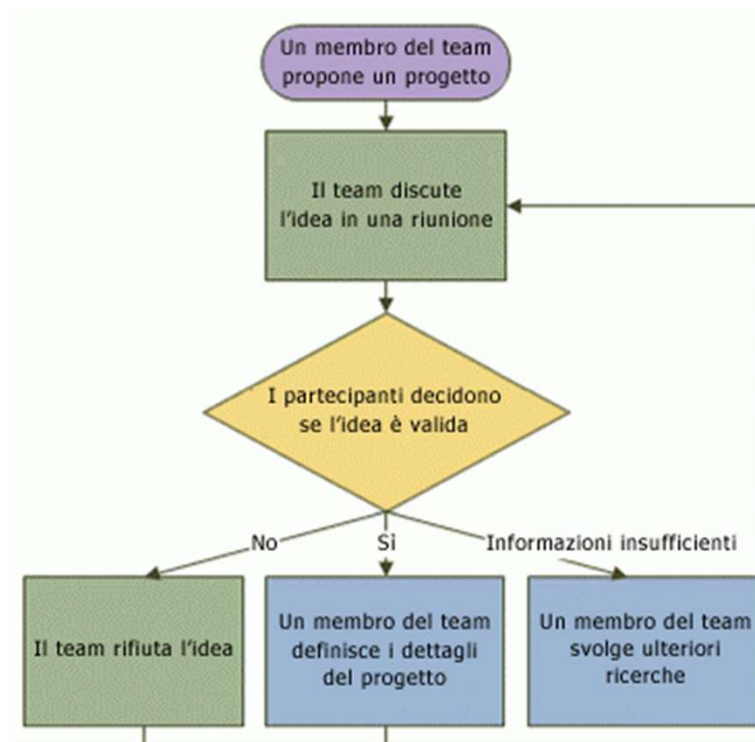


Programma

```
1 function [M,N,new,p2p,p2pCFD,times,params] = compute_CMH_mesh_transfer(M, N, KM, kN, hands_radius, params)
2
3 %% prepare MESH struct
4 disp('  prepare MESHES')
5 tINI = tic;
6 times = [];
7
8 landmarks = [M.Lm N.Lm];
9 M1 = MESH('Src',M.VER,M.TRIV);
10 M2 = MESH('Tar',N.VER,N.TRIV);
11
12 %k = findK(M);
13
14
15 if(params.Mipcd)
16     for k_nn = 6:params.knnMax
17         params.knn = k_nn;
18
19         if params.fixedRadius
20             L1 = lb_basis_pcd_radius(M, 200, params.knn);
21             %L1 = lb_basis_pcd_radiusVarKnn(M, 200, params.knn);
22         elseif params.mean32
23             L1 = lb_basis_pcd_mean32(M, 200, params.knn);
24         else
25             L1 = lb_basis_pcd(M, 200, params.knn);
26         end
27
28         if (sum(diff(L1.evecs(:,2))) < - 0.00001) || sum(diff(L1.evecs(:,2))) > 0.00001
29             %if isFound(L1.evecs)
30             break;
31         else
32             fprintf('-----constant functions \n');
33         end
34     end
35 end
36
37 L1.pcd = 1;
```

Traduco i passaggi dell'algoritmo in un linguaggio specifico, comprensibile dalla macchina!

Esempio: Diagramma di flusso



Esempio: Diagramma di flusso

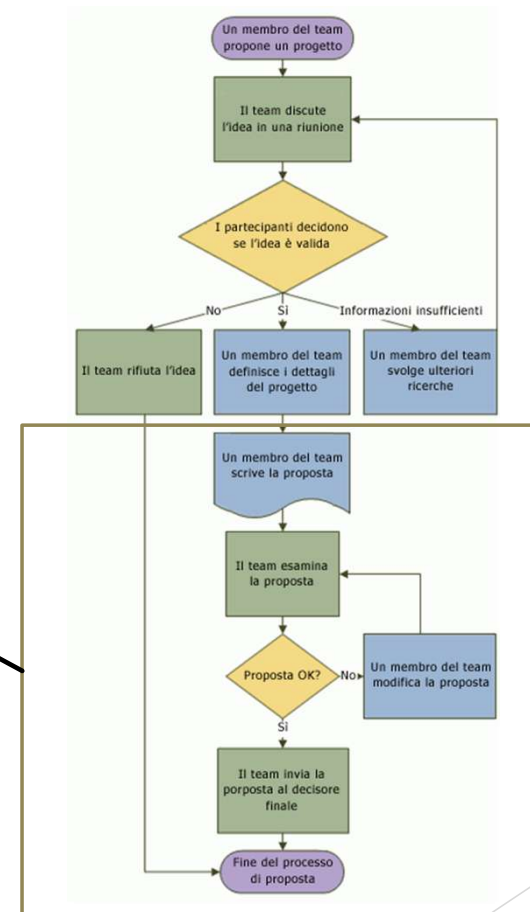
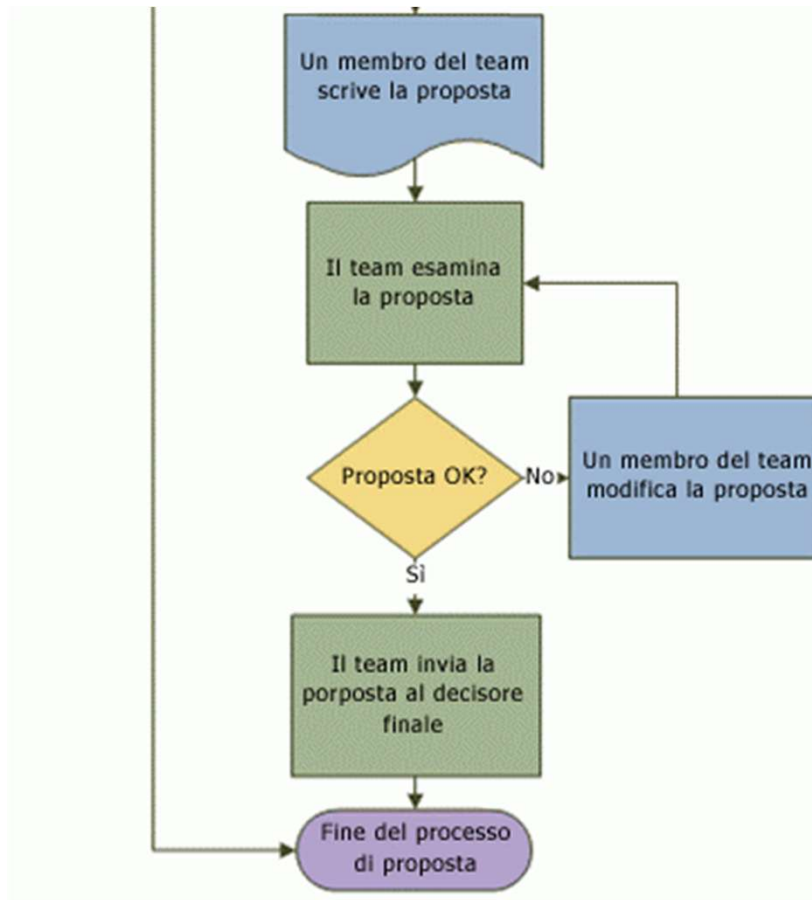


Diagramma di flusso: esempio

Ingredienti e strumenti (dati)

- 1/2 etto di burro, 1/2 etto di farina, 1/2 litro di latte, 1/2 cucchiaino di sale, 5 grammi di pepe e noce moscata
- 1 casseruola, 1 setaccio, 1 coltello, 1 cucchiaio, 1 fornello

Preparazione (azioni)

- setaccia la farina
- metti la farina setacciata nella casseruola con 50 cc di latte freddo
- **se** il burro è in un pezzo unico **allora** taglia a pezzetti il burro
- aggiungi alla casseruola il burro e 300 cc di latte
- accendi il fuoco
- metti la casseruola sul fuoco
- **finché** non bolle **ripeti** mescola
- **se** vuoi piccante
 allora finché non piccante **ripeti** aggiungi pepe
 altrimenti aggiungi noce moscata
- aggiungi 150 cc di latte
- fai cuocere per 20 minuti

Diagramma di flusso: esempio

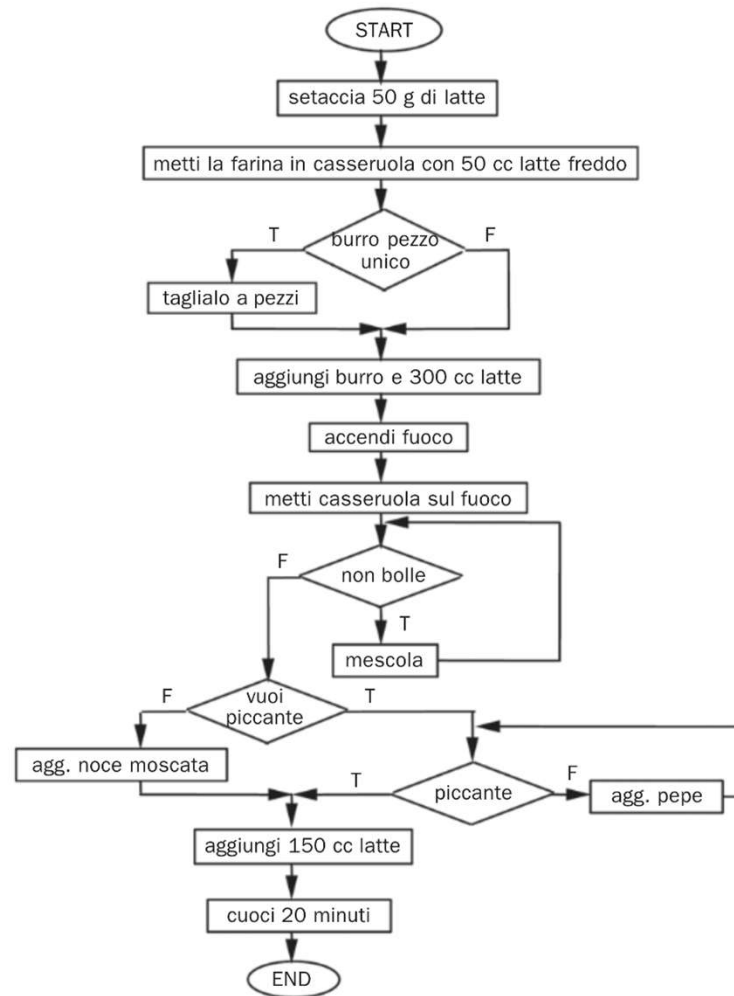
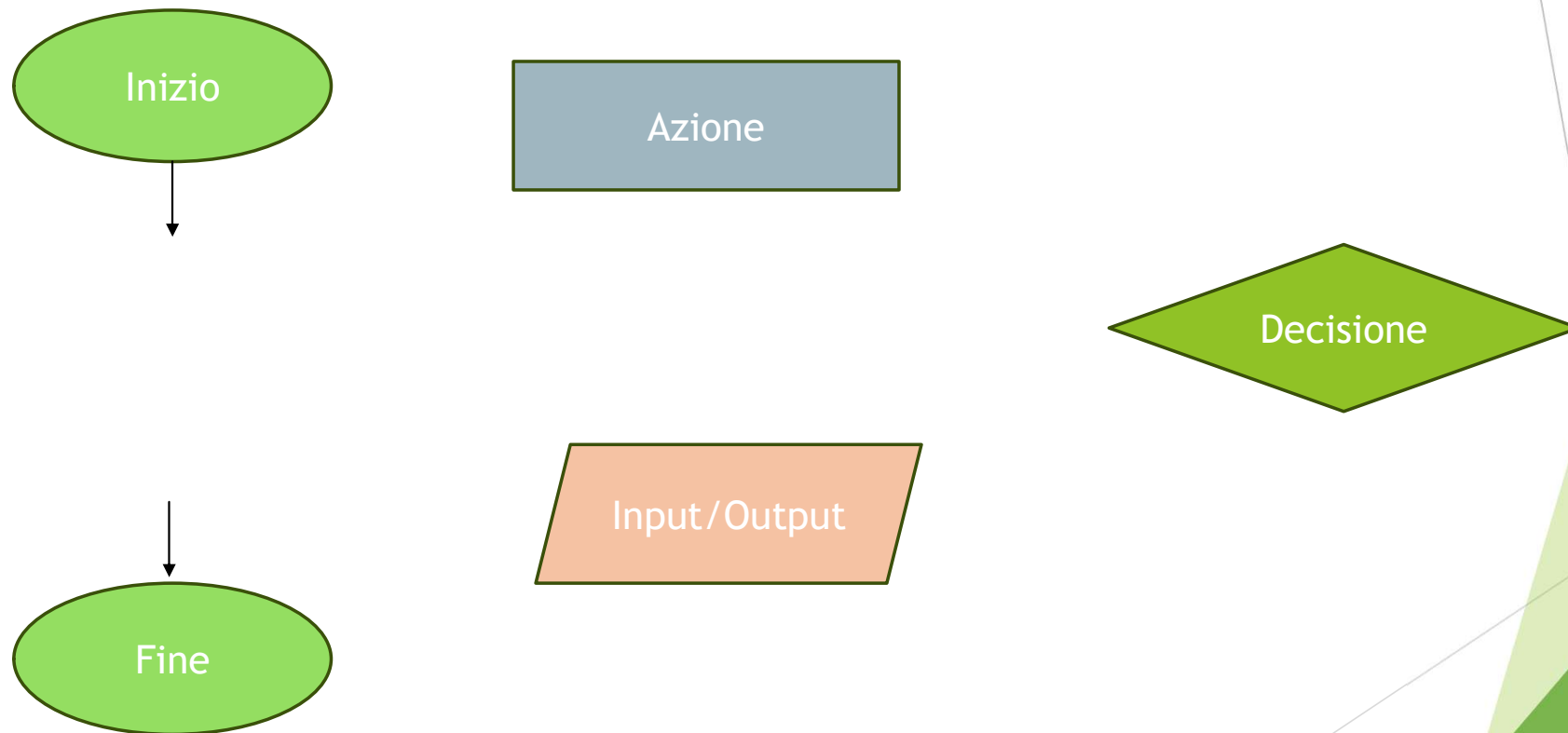
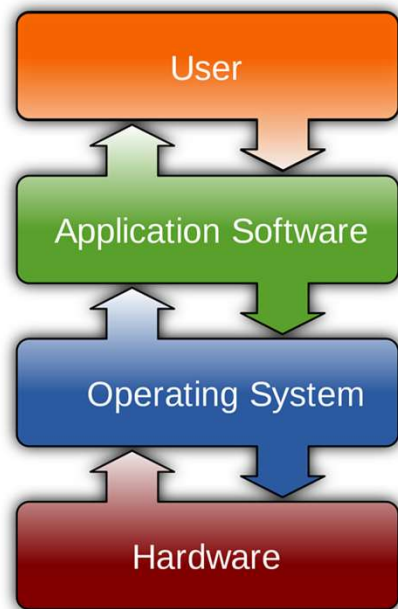


Diagramma di flusso per un algoritmo

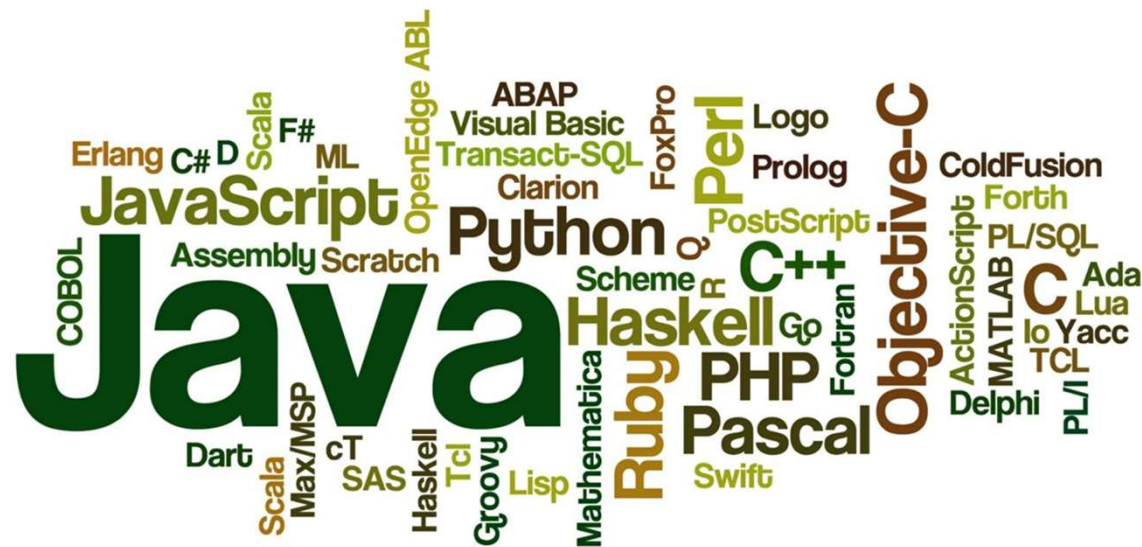


ALGORITMO

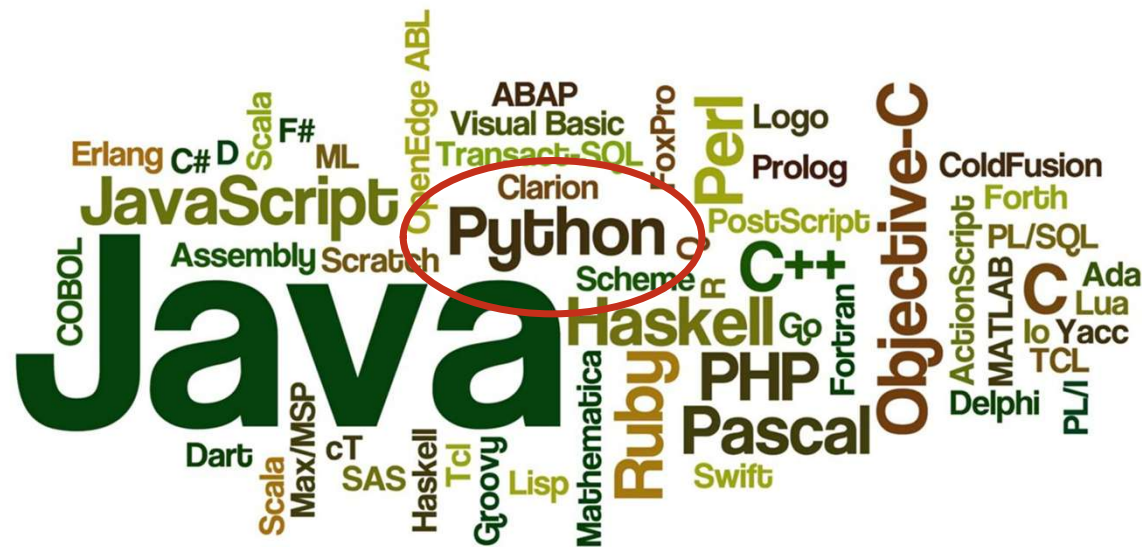


- ▶ Interazione fra uomo e la macchina
- ▶ Serve un linguaggio di interazione

LINGUAGGIO DI PROGRAMMAZIONE

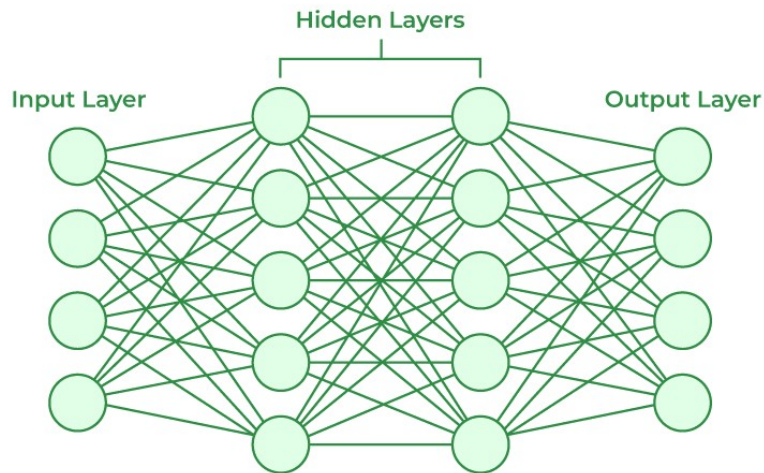


LINGUAGGIO DI PROGRAMMAZIONE

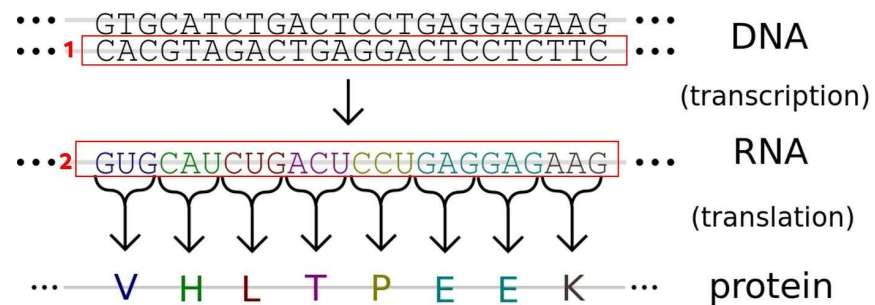


perché PYTHON?

Reti Neurali (IA)



Operazioni su stringhe (bioinformatica)



Computer Vision



Web server

python

web server


SHELL

- ▶ Operazioni singole (posso usarla come una calcolatrice)
Scrivo un'espressione e ottengo subito il risultato
- ▶ Funziona con tutti i comandi di python

Shell

```
>>> 4 * 3  
12  
>>> 6 + 5  
11  
>>> 7 / 2  
3.5
```


AMBIENTE: IDLE



The image shows a screenshot of the IDLE Shell 3.8.10 window. The window has a title bar with the text "IDLE Shell 3.8.10" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window displays the following text: "Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>> |". The status bar at the bottom right of the window shows "Ln: 3 Col: 4".

```
IDLE Shell 3.8.10
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [M
SC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>> |
Ln: 3 Col: 4
```

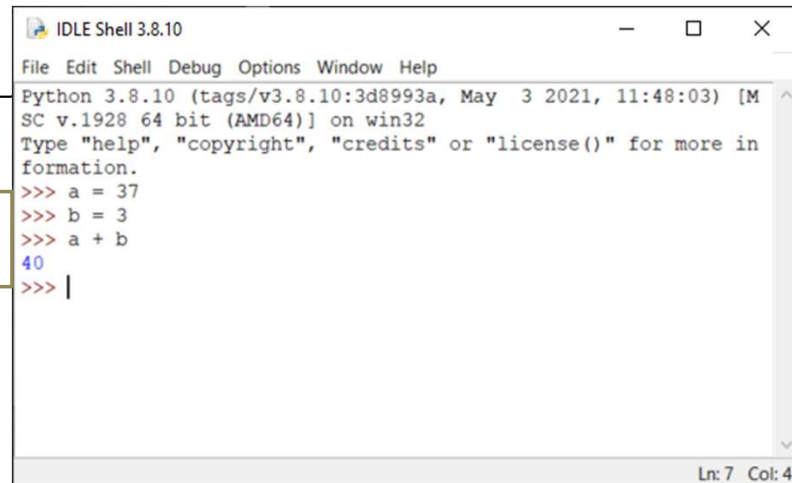
AMBIENTE: IDLE

Versione di
python in uso

Operazioni

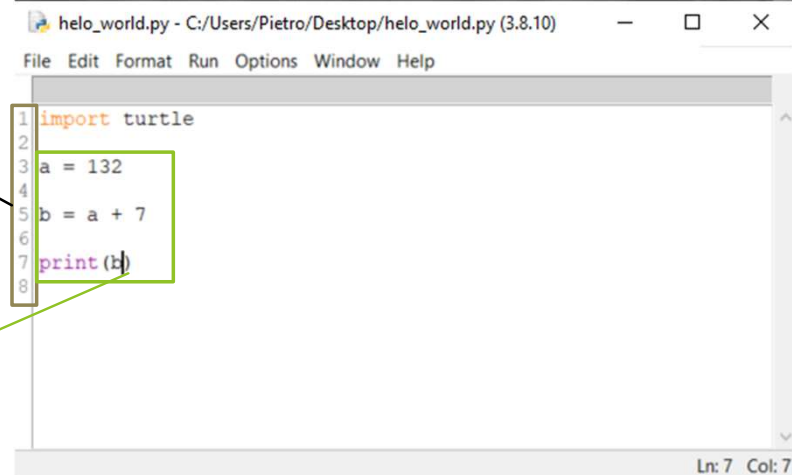
Numero della
riga, molto utile!

operazioni



```
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>> a = 37
>>> b = 3
>>> a + b
40
>>> |
```

Shell



```
1 import turtle
2
3 a = 132
4
5 b = a + 7
6
7 print(b)
8
```

Script file

AMBIENTE: IDLE

Versione di
python in uso

Operazioni

Shell

Shell:

- esegui un comando alla volta
- Utile per provare nuovi comandi

Script:

- i comandi sono eseguiti tutti in sequenza
- Necessario per task complessi

Script file

operazioni

The screenshot shows the IDLE Python environment. The top window is the 'IDLE Shell 3.8.10' window, which contains the Python interpreter's startup message and a series of commands: `>>> a = 37`, `>>> b = 3`, and `>>> a + b`. The bottom window is the 'Script file' window, which contains a script with the following code: `1 b = a + 7`, `2`, `3`, `4 print(b)`, and `5`. The status bar at the bottom of the Script file window indicates 'Ln: 7 Col: 7'.

```
IDLE Shell 3.8.10
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 37
>>> b = 3
>>> a + b
40

Script file
1 b = a + 7
2
3
4 print(b)
5
Ln: 7 Col: 7
```

PROVIAMO A USARE QUESTI STRUMENTI

- ▶ **IDE:** Integrated Development Environment
- ▶ Noi useremo IDLE

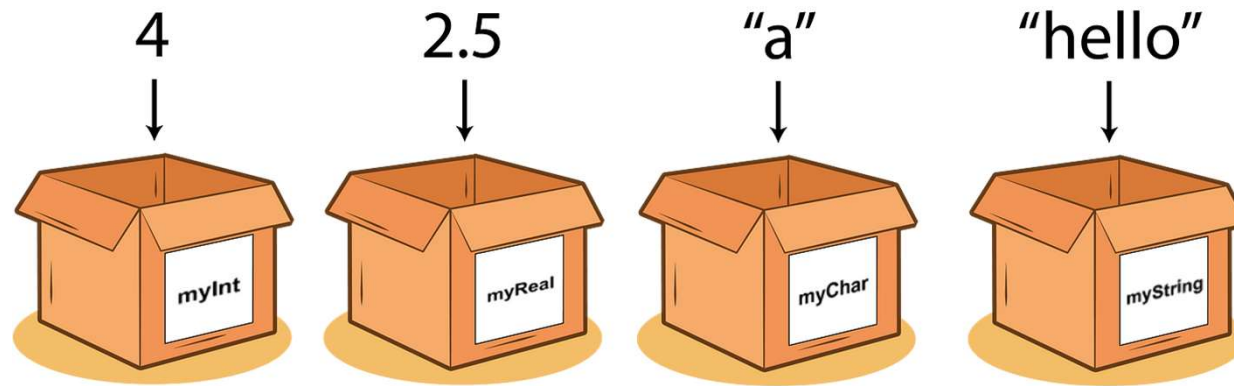
TIPI

- ▶ **Numeri:** int (valori interi), float (numeri con la virgola)
1, -5, 7.4, 3.141593, 2300329131413...
- ▶ **Stringhe:** (lettere, parole, frasi, lunghi testi...)
'Pietro' 'Pippo' 'Once upon a time...'
- ▶ **Booleani:** (usati per le condizioni logiche)
True | False



VARIABILI

Quando voglio conservare un valore!



- ▶ Metto un'etichetta a un valore per poterlo riutilizzare in seguito
- ▶ Quando creare una variabile?
 - ▶ Quanto so che mi servirà un valore in seguito
 - ▶ Quando devo fare operazioni in più passaggi

Assegnamento

Nome con cui voglio chiamare la variabile

Valore che voglio dare alla variabile

nome_variabile = valore

Operatore di assegnamento
(N.B. non confondere con ==)

Shell

```
>>> base = 3
>>> altezza = 5
>>> area = base * altezza
>>> print(area)
15
>>> area = 2
>>> print(area)
2
```

numero

Risultato operazione

Ri-assegnamento (perdo il valore precedente)

OPERAZIONI

- ▶ Operazioni su numeri (int e float): +, -, *, /, **, %, //
- ▶ Comparazioni: <, >, <=, >=, ==
- ▶ Espressioni logiche: or, and, not

OPERAZIONI

Shell

```
>>> 7 / 4
```

```
1.75
```

```
>>> 7 // 4
```

```
1
```

```
>>> 7 % 4
```

```
3
```

```
>>> 2 ** 8
```

```
256
```

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

```
>>> a < 3
```

```
False
```

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

```
True
```

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

```
True
```

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

```
True
```

Tengo solo la parte intera

Restituisce il resto della divisione intera

Elevamento a potenza

Attenzione a usare >= e <=

OPERAZIONI

Shell

```
>>> a = 5  
>>> a == 5 or a < 5  
True  
>>> a == 5 and a > 3  
True  
>>> not True  
False  
>>> not a == 5  
False
```


OPERAZIONI

A	B	A AND B	A OR B	NOT A
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE

STRINGHE

- ▶ Una stringa è una successione di caratteri alfanumerici
 - ▶ Lettere singole (comprese le cifre)
 - ▶ Parole
 - ▶ Frasi, testi lunghi...
 - ▶ Caratteri speciali. ES `'\n'` carattere per andare a capo

Attenzione!!! 234 è diverso da `'234'`

STRINGHE

- ▶ Python ha molte funzioni per gestire le stringhe
 - ▶ `+` : Concatena stringhe
 - ▶ `split(',')` : suddivide tutta la stringa in sottostringhe separate dal carattere indicato
 - ▶ `replace('b','c')` : Sostituisce ogni 'b' con 'c' all'interno di una stringa

STRINGHE

Shell

```
>>> stringa1 = 'Sono'  
>>> stringa2 = 'Pietro'  
>>> print(stringa1 + stringa2)  
SonoPietro
```

Come faccio?

STRINGHE

Shell

```
>>> stringa1 = 'Sono'
>>> stringa2 = 'Pietro'
>>> print(stringa1 + stringa2)
SonoPietro
>>> stringa2 = ' Pietro'
>>> stringa1 + stringa2
Sono Pietro
```

→ Ora lo spazio c'è!

STRINGHE

Shell

```
>>> stringa_prova = 'Sono andato nel palazzo'  
>>> stringa_prova.replace('nel', 'sul')  
Sono andato sul palazzo  
>>> stringa_prova  
Sono andato nel palazzo
```

STRINGHE

Shell

```
>>> stringa_prova = 'Sono andato nel palazzo'
>>> stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
>>> stringa_prova
Sono andato nel palazzo
>>> stringa_prova = stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
```


STRINGHE

Shell

```
>>> stringa_prova = 'Sono andato nel palazzo'
>>> stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
>>> stringa_prova
Sono andato nel palazzo
>>> stringa_prova = stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
>>> stringa_prova
Sono andato sul palazzo
```

STRINGHE

Shell

```
>>> stringa_prova = 'Sono andato nel palazzo'
>>> stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
>>> stringa_prova
Sono andato nel palazzo
>>> stringa_prova = stringa_prova.replace('nel', 'sul')
Sono andato sul palazzo
>>> stringa_prova
Sono andato sul palazzo
>>> stringa_prova = stringa_prova.replace('sul', 'dietro al')
Sono andato dietro al palazzo
```

STRINGHE lower e upper

- ▶ Nelle stringhe il maiuscolo e il minuscolo sono valori differenti!
- ▶ Posso usare *nome_stringa.lower()* e *nome_stringa.upper()* per portare rispettivamente la stringa tutta minuscola e tutta maiuscola

Shell

```
>>> stringa = 'hello, world'
>>> stringa.upper()
'HELLO WORLD'
>>> STRINGA = 'HELLO, WORLD'
>>> STRINGA.lower()
'hello, world'
```

OPERAZIONI

Esempi espressioni logiche:

- ▶ 1 è maggiore di -2?
- ▶ 0 è compreso tra -3 e 3?
- ▶ x non è compreso tra 3 e 6?

In Python

- ▶ `1 > -2`
- ▶ `0 > -3 and 0 < 3`
- ▶ `not (x < 6 and x > 3)`

INPUT/OUTPUT

- ▶ Posso richiedere che l'utente inserisca un input
- ▶ L'input inserito viene salvato come una variabile dal programma
- ▶ Con `print()` posso 'stampare' a video una stringa, comprese le mie variabili

Attenzione !!! Qualsiasi input viene salvato come una stringa!

Shell

```
>>> mio_input = input('Come ti chiami? ')
Come ti chiami? Pietro
>>> print('Ah, ciao ', mio_input, '!')
Ah, ciao Pietro !
```

INPUT/OUTPUT: esempio

Script

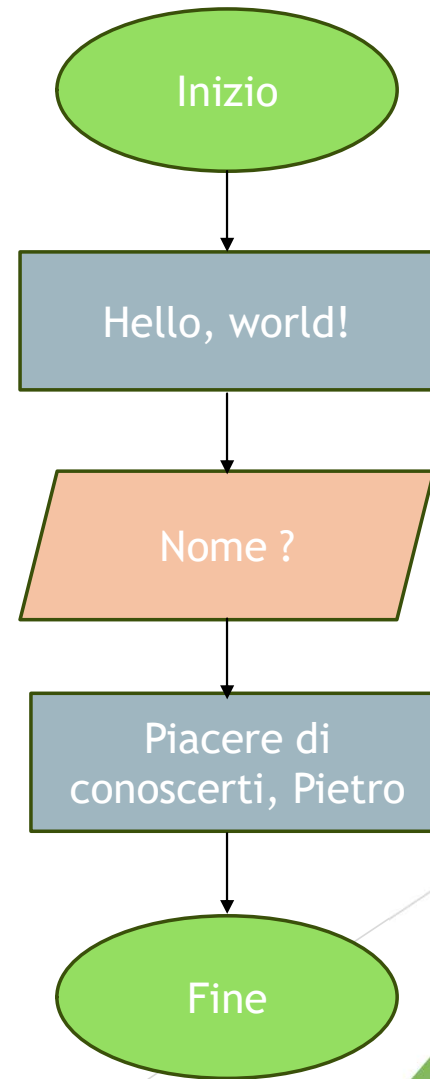
```
print('Hello, world!')  
  
nome = input('Come ti chiami? ')  
  
print('Piacere di conoscerti, ', nome)
```

- ▶ Fare sempre attenzione: lo spazio ' ' è un carattere, se lo vogliamo inserire dobbiamo esplicitarlo nelle stringhe che usiamo
- ▶ 'nome' è la variabile in cui salvo l'input dell'utente
- ▶ Fare *run* sullo script e poi vedere cosa succede sulla *shell*

INPUT/OUTPUT: esempio

Script

```
Print('Hello, world!')  
  
nome = input('Come ti chiami? ')  
  
print('Piacere di conoscerti, ', nome)
```

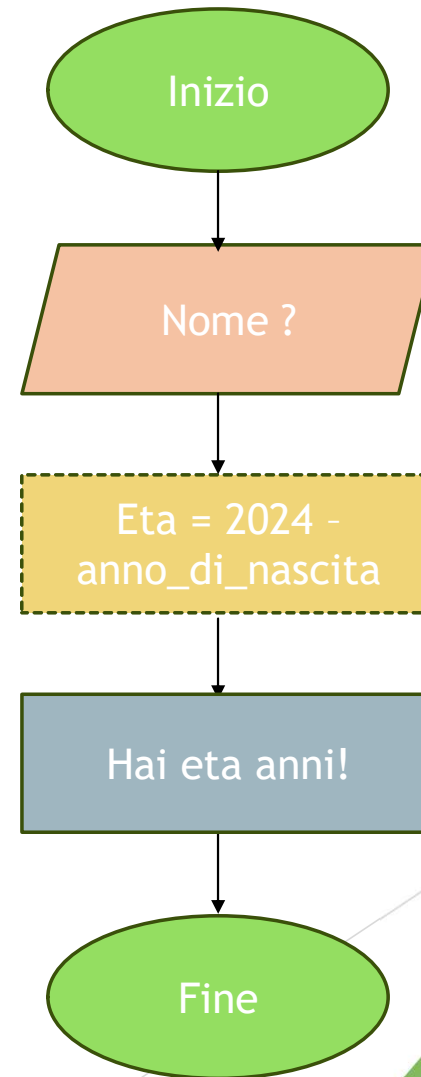


INPUT/OUTPUT: esempio

Script

```
anno_di_nascita = input('In che anno sei nato? ')\neta = 2024 - anno_di_nascita\nprint('Hai ', eta, ' anni!')
```

- ▶ Purtroppo non funziona...
- ▶ La variabile in input è una stringa, non un numero!
- ▶ Non posso fare la sottrazione tra un numero e una stringa!

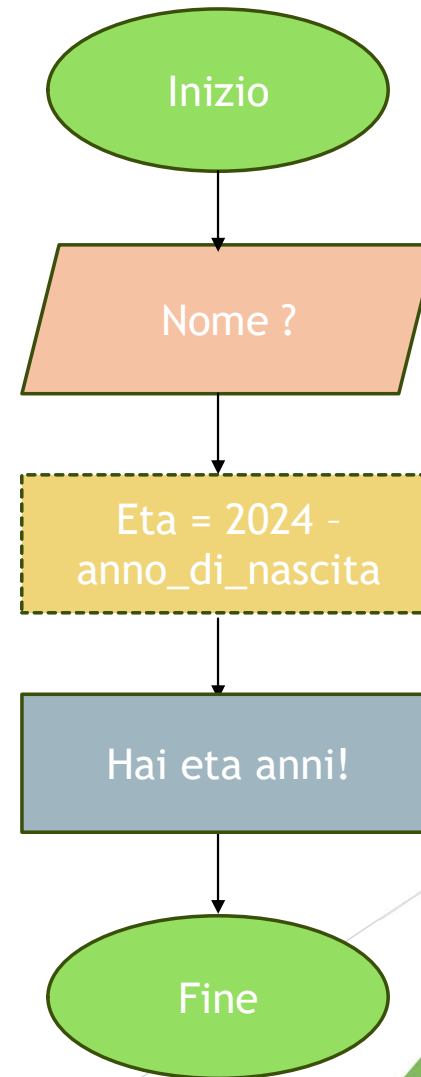


INPUT/OUTPUT: esempio

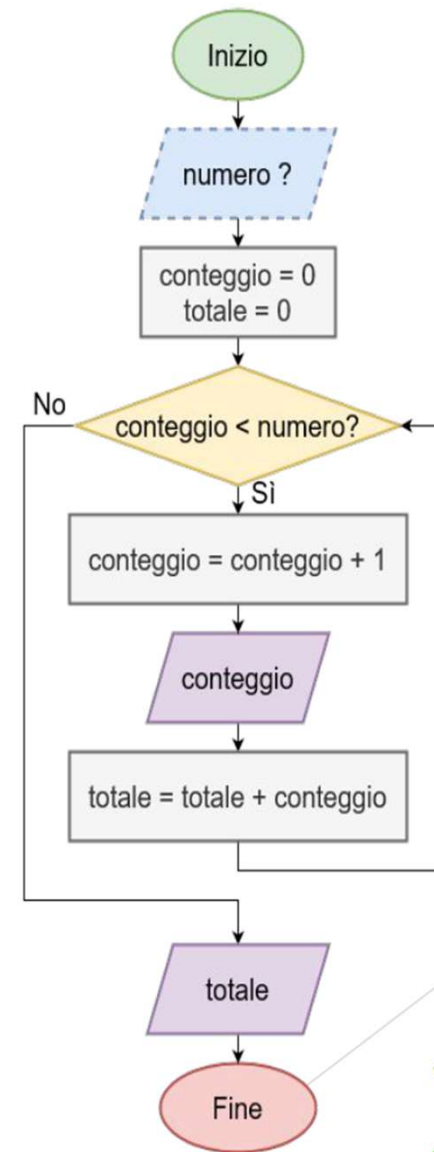
Script

```
anno_di_nascita = input('In che anno sei nato? ')\nanno_di_nascita = int(anno_di_nascita)\n\neta = 2024 - anno_di_nascita\n\nprint('Hai ', eta, ' anni!')
```

- Uso `int()` per cambiare il tipo da stringa a intero
- Questa operazione si chiama cast, come `int()`, c'è anche `float()` per i numeri con la virgola

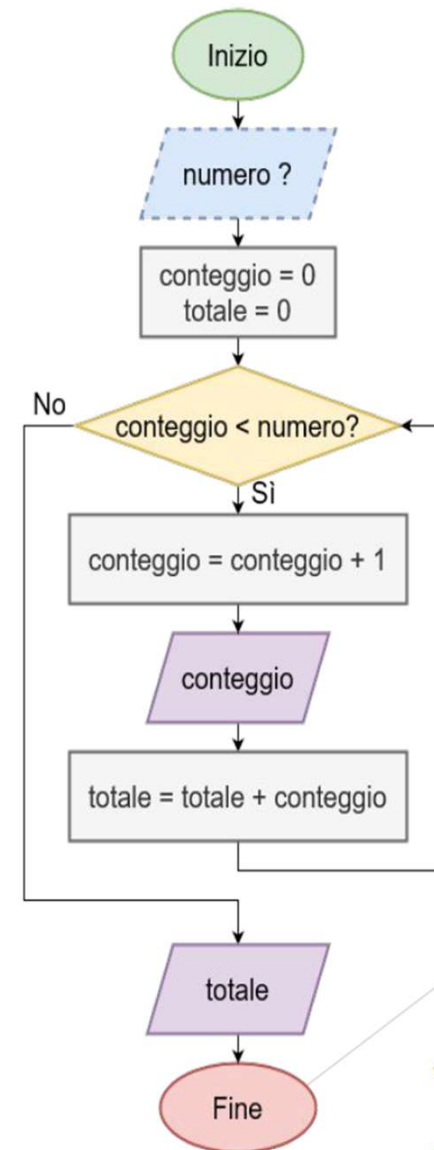


Esempio Diagramma di Flusso



Esempio Diagramma di Flusso

- Cosa fa questo diagramma?
- Chiede un input
- Crea due variabili, una per tenere il conto e una per tenere il totale della somma
- Cosa indica il totale?

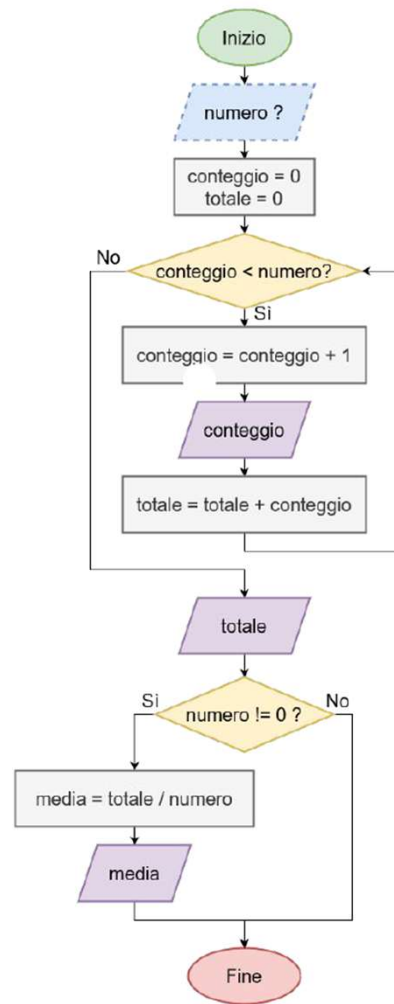


Esempio Diagramma di flusso

- ▶ Se volessi fare la media dei numeri sommati precedentemente?

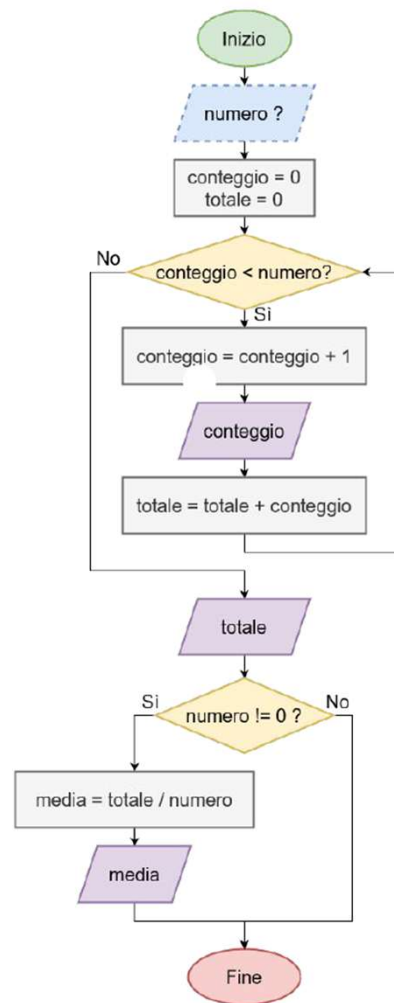


Esempio Diagramma di flusso



- Se volessi fare la media dei numeri sommati precedentemente?

Esempio Diagramma di flusso



- Se volessi fare la media dei numeri sommati precedentemente?
- Controllo che il numero sia diverso da 0
- Divido il totale per il numero
- Stampo il risultato

Esercizi

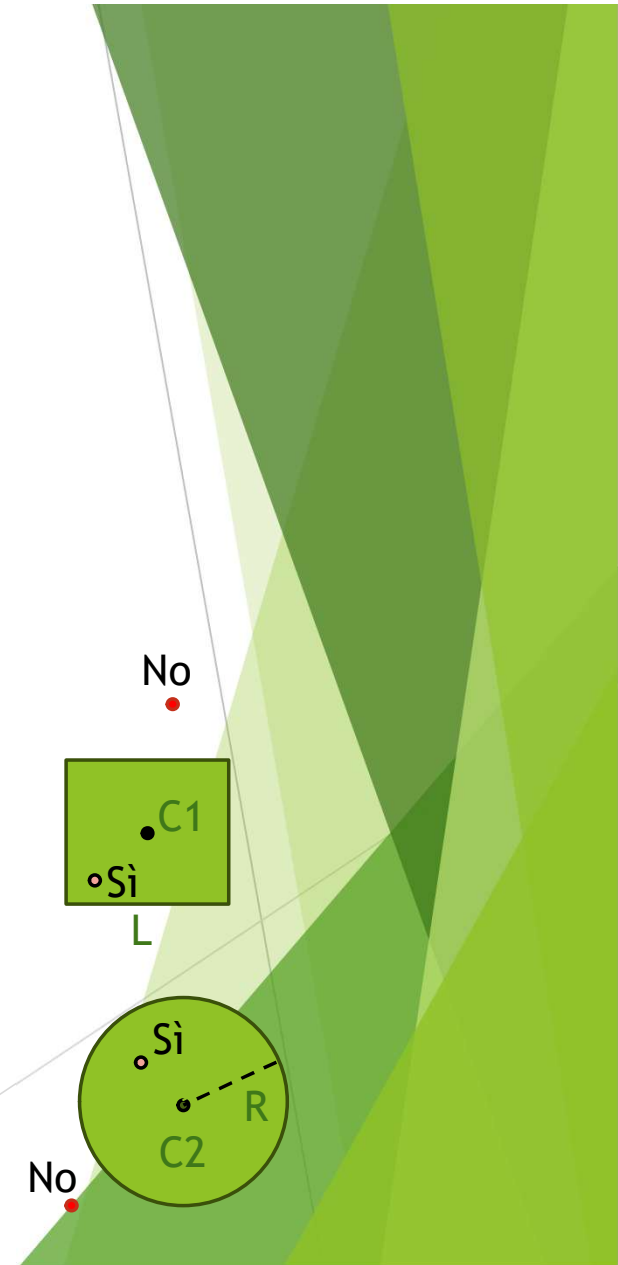
- 1. Se voglio ottenere questo risultato che codice dovrò scrivere?

Shell

```
...codice...
MichelaRossi (nome e cognome da input)
...codice...
Michela Rossi
...codice...
Michela Rossi sta seguendo il bootcamp di python
...codice...
Michela Rossi, una mia amica, sta seguendo il
bootcamp di python.
...codice...
Michela Rossi, una mia amica, sta seguendo il
bootcamp di python.
Meglio stare attenti alle stringhe!
```

Esercizi

- ▶ Calcolare il valore assoluto di un numero in input arbitrario. Fare il diagramma di flusso di questo programma
- ▶ Calcolo il prodotto di due numeri (usando l'addizione). Fare il diagramma di flusso di questo programma.
- ▶ Controllare se un punto nello spazio 2D (coordinate x, y) risiede all'interno di un quadrato (dati il centro del quadrato $C1(C1_x, C1_y)$ e la lunghezza del lato L). Richiedere i valori x, y e $C1_x, C1_y$ da input. Fare il diagramma di flusso per questo programma
- ▶ Controllare se un punto nello spazio 2D (coordinate x, y) risiede all'interno di un cerchio (dati il centro del cerchio $C2(C2_x, C2_y)$ e la lunghezza del raggio R). Richiedere i valori x, y e $C2_x, C2_y$ da input. Fare il diagramma di flusso per questo programma



Esercizi

- ▶ Controllare se un punto nello spazio 2D (coordinate x , y) risiede all'interno di un triangolo (dati i tre punti che compongono il triangolo (x_1, y_1) , (x_2, y_2) , (x_3, y_3)). Richiedere i valori da input. Fare il diagramma di flusso per questo programma
- ▶ Voglio che l'utente inserisca due numeri. Moltiplico questi due numeri e poi li stampo sulla shell. (i numeri inseriti possono essere con la virgola). Fare il diagramma di flusso e poi scrivere il codice corrispondente
- ▶ Fare la divisione tra due numeri inseriti e stampare solo la parte intera del risultato ('la parte intera della divisione tra ... e ... è: ...'), poi andare a capo e stampare il resto della divisione. ('Il resto della divisione tra ... e ... è: ...'). Fare il diagramma di flusso e poi scrivere il codice corrispondente