PYTHON

Note:

* **input:** ricezione di dati da tastiera, da file o da altro dispositivo.
* **output:** scrittura di dati su video, su file o trasmissione ad altro dispositivo.
* **matematiche:** esecuzione di semplici operazioni matematiche, quali l'addizione e la sottrazione.
* **condizionali:** controllo di alcune condizioni ed esecuzione della sequenza di istruzioni appropriata.
* **ripetizione:** ripetizione di un'azione, di solito con qualche variazione.

Esempi :

print 1+1

2

Print “ciao”

Ciao

Per scrivere i numeri non servono le “

**Glossario**

Soluzione di problemi

il processo di formulare un problema, trovare una soluzione ed esprimerla.

Linguaggio ad alto livello

un linguaggio di programmazione tipo Python che è progettato per essere facilmente leggibile e utilizzabile dagli esseri umani.

Linguaggio di basso livello

un linguaggio di programmazione che è progettato per essere facilmente eseguibile da un computer; è anche chiamato "linguaggio macchina" o "linguaggio assembly".

Portabilità

caratteristica di un programma di poter essere eseguito su computer di tipo diverso.

Interpretare

eseguire un programma scritto in un linguaggio di alto livello traducendolo ed eseguendolo immediatamente, una linea alla volta.

Compilare

tradurre un programma scritto in un linguaggio di alto livello in un programma di basso livello come preparazione alla successiva esecuzione.

Codice sorgente

un programma di alto livello prima di essere compilato.

Codice oggetto

il risultato ottenuto da un compilatore dopo aver tradotto il codice sorgente.

Eseguibile

altro nome per indicare il codice oggetto pronto per essere eseguito.

Script

programma memorizzato in un file, solitamente destinato ad essere interpretato.

Programma

serie di istruzioni che specificano come effettuare un'elaborazione.

Algoritmo

processo generale usato per risolvere una particolare categoria di problemi.

Bug

errore in un programma (detto anche "baco").

Debug

processo di ricerca e di rimozione di ciascuno dei tre tipi di errori di programmazione.

Sintassi

struttura di un programma.

Errore di sintassi

errore in un programma che rende impossibile la continuazione dell'analisi del codice (il programma non può quindi essere interpretato interamente o compilato).

Errore in esecuzione

errore che non è riconoscibile finché il programma non è stato eseguito e che impedisce la continuazione della sua esecuzione.

Eccezione, errore runtime

altri nomi per indicare un errore in esecuzione.

Errore di semantica

errore nel programma che fa ottenere risultati diversi da quanto ci si aspettava.

Semantica

significato di un programma.

Linguaggio naturale

ognuno dei linguaggi parlati evoluti nel tempo.

Linguaggio formale

ognuno dei linguaggi che sono stati progettati per scopi specifici, quali la rappresentazione di idee matematiche o programmi per computer (tutti i linguaggi per computer sono linguaggi formali).

Token

uno degli elementi di base della struttura sintattica di un programma analogo alla parola nei linguaggi naturali.

Parsing

esame e analisi della struttura sintattica di un programma.

Istruzione di stampa

istruzione che ordina all'interprete Python di scrivere un valore sullo schermo.

Le stringhe appartengono al tipo string e gli interi al tipo int.

>>> type("Hello, World!")   
<type 'string'>   
>>> type(17)   
<type 'int'>

I numeri con il punto decimale appartengano al tipo float: questi numeri sono rappresentati in un formato chiamato **virgola mobile** o **floating-point**.

>>> type(3.2)   
<type 'float'>

Qunado i numeri sono racchiusi nelle “ sono delle stringhe come le parole mentre se non hanno i “ sono numeri

Python ha 28 parole riservate alla struttura che non possono essere usate nelle variabili :

and      continue  else      for      import    not      raise   
assert   def       except    from     in        or       return   
break    del       exec      global   is        pass     try   
class    elif      finally   if       lambda    print    while

L'istruzione print invece stampa il valore dell'espressione, che nel caso delle stringhe corrisponde al loro contenuto. Le virgolette sono quindi rimosse.

Segni :

+ = addizione

- = sottrazione

/ = divisione

\* = moltiplicazione

\*\* = potenza

* **Parentesi**: hanno il più alto livello di precedenza e possono essere usate per far valutare l'espressione in qualsiasi ordine. Dato che le espressioni tra parentesi sono valutate per prime, 2\*(3-1) dà come risultato 4, e (1+1)\*\*(5-2) dà 8. Puoi usare le parentesi per rendere più leggibile un'espressione come in (minuti\*100)/60, anche se questo non influisce sul risultato.
* **Elevamento a potenza**: ha la priorità successiva così 2\*\*1+1 fa 3 e non 4, e 3\*1\*\*3 fa 3 e non 27.
* **Moltiplicazione** e **Divisione** hanno la stessa priorità, superiore a somma e sottrazione. 2\*3-1 dà 5 e non 4, e 2/3-1 fa -1, e non 1 (ricorda che la divisione intera 2/3 restituisce 0).
* **Addizione** e **Sottrazione**, anch'esse con la stessa priorità.
* Gli operatori con la stessa priorità sono valutati da sinistra verso destra, così che nell'espressione minuti\*100/60, la moltiplicazione è valutata per prima, ottenendo 5900/60, che a sua volta restituisce 98. Se le operazioni fossero state valutate da destra a sinistra il risultato sarebbe stato sbagliato: 59\*1=59.