

ISTRUZIONI PYTHON

STAMPA = `print()`

STRINGA = `"..."`

TIPO VARIABILE = `type(variabile)`

ISTRUZIONE ASSEGNAZIONE = `=`

PAROLE RISERVATE A PYTHON = `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`

Elevamento a potenza = `**`

N.B. Nella divisione quando entrambi gli operandi sono numeri interi il risultato è sempre un numero intero

COMMENTO = `#`

VARIABILE INTERA = `int`

VARIABILE DECIMALE = `float`

VARIABILE STRINGA = `str`

FILE FUNZIONI MATEMATICHE = `import math`

N.B. Per chiamare una funzione di un modulo dobbiamo specificare il nome del modulo che la contiene e il nome della funzione separati da un punto. ES. `altezza = math.sin(angolo)`

FUNZIONE = `def NOME (LISTA_DEI_PARAMETRI):`

`ISTRUZIONI`

RICHIAMARE FUNZIONE = `NOME (LISTA_DEI_PARAMETRI)`

OPERATORE MODULO (RESTO DELLA DIVISIONE) = `%`

ESPRESSIONE BOOLEANA = un'espressione che `e vera ha valore 1, un'espressione falsa ha valore 0

`==` → **confronta** due valori e produce un risultato di tipo booleano

`!=` → diverso

N.B. `=` `e un operatore di assegnazione e `==` un operatore di confronto.

OPERATORI LOGICI = `and`, `or` e `not`

ISTRUZIONE CONDIZIONALE = `if CONDIZIONE1:`

`print "....."`

`elif CONDIZIONE2:`

`print "....."`

`else CONDIZIONE3:`

`print "....."`

N.B. Non c'è alcun limite al numero di istruzioni elif ma è possibile inserire un'unica istruzione else che deve essere l'ultima dell'elenco.

Return= termina l'esecuzione di una funzione prima di raggiungerne la fine

INSERIMENTO DA TASTIERA= input ()

FUNZIONE PRODUTTIVA= return variabile (alla fine della funzione)

Ciclo di esecuzione di un'istruzione while:

1. Valuta la condizione controllando se essa è vera (1) o falsa (0).
2. Se la condizione è falsa esci dal ciclo while e continua l'esecuzione dalla prima istruzione che lo segue.
3. Se la condizione è vera esegui tutte le istruzioni nel corpo del while e torna al passo 1.

OPERATORE PORZIONE= nomestringa[indice] (restituisce una lettera della stringa)

N.B. I conteggi partono da 0 e non da 1 e per accedere al primo carattere di una stringa dobbiamo richiedere il numero 0.

NUMERO CARATTERI STRINGA= len(nomestringa)

SELEZIONE PORZIONE STRINGA= nomestringa[n:m] (n= posizione dell'inizio della porzione della stringa m=fine)

LISTA= serie ordinata di valori, ognuno identificato da un indice. Definita da elementi racchiusi tra []

N.B. La sintassi per l'accesso agli elementi di una lista è la stessa dei caratteri di una stringa.

N.B. Se un indice ha valore negativo il conteggio parte dalla fine della lista.

In = controlla se un valore è presente in una lista

OPERAZIONI LISTE=

- L'operatore + concatena le liste;
- L'operatore * ripete una lista un dato numero di volte.

del nomelista[indice]= rimuove un elemento da una lista.

DIZIONARI= possono usare qualsiasi tipo di dato immutabile come chiave(indice). Il dizionario vuoto è indicato da {}. Un elemento è detto coppia chiave-valore. ES: {'one': 'uno', 'two': 'due'}

del nomedizionario[chiave]= rimuove una coppia chiave-valore.

Modificare un valore associato= nomedizionario['chiave']=nuovovalore

len(dizionario)= numero di coppie chiave-valore

nomedizionario.keys()= elenco chiavi dizionario

nomedizionario.value()= elenco valori dizionario

nomedizionario.items()= elenco chiave-valore

nomedizionario.has_key(chiave)= ritorna vero(1) se la chiave è presente e falso(0) se non lo è

Aprire un file= variabile= open("nomefile", "modo")

Modo=

- w= apriamo il file in scrittura(file vuoto) → Inserire dati nel file= nomefile.write("testodainserire")→ Chiusura file= nomefile.close()
- r=apriamo il file in lettura→quanti caratteri deve leggere=printnomevariabile.read(numcaratteri)

print nomefile.readline()=legge la prima riga

print nomefile.readlines()=legge le altre righe sottoforma di lista

Aprire un file esterno= f = open("/usr/share/dict/words","r") specificare da destra verso sinistra il nome del file e da dove proviene

float('inf')= valore infinito

"testo in minuscolo".upper()="testo in maiuscolo"

"testo in maiuscolo".lower()="testo in minuscolo"

Nomelista.sort()= mette gli elementi in ordine crescente

Funzioni utili per liste:

<code>list.sort()</code>	ordina la lista
<code>list.index("value")</code>	restituisce l'indice dove viene trovata la prima occorrenza di "value"
<code>list.append("value")</code>	aggiunge l'elemento "value" alla fine della lista

Nomestringa.isdigit () = controlla se la stringa è formata da sole cifre