

FUNZIONE

PRINT()

STAMPARE VALORI - STRINGHE

PRINT("CIAO") = CIAO

s = "CIAO"
PRINT(s) = CIAO

INPUT()

INSERIRE UN VALORE DA TASTIERA
IN UNA VARIABILE

NAME = INPUT("INSERISCI IL TUO NOME -> ")
STRINGA STAMPATA IN CONSOLE

FLOAT()

CONVERTIRE UNA STRINGA INSERITA DA TASTIERA
IN UN NUMERO DECIMALE OCL.

ETA = INPUT("QUANTI ANNI HAI")
ANNI = FLOAT(ETA)

- VARIABILI

INT = NUMERO INTERO

LIST = LISTE (SEQUENZA MUTABILE ORDINATA)

FLOAT = NUMERO CON VIRGOLA

TUPLE = TUPLE ("IMMUTABILE")

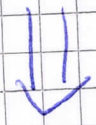
STR = TESTO (STRINGA)

DICTIONARY = DIZIONARI (ASSOCIA CHIAVO - VALORE)

BOOL = VERO o FALSO

IF (SE)

GRUPPO DI CONDIZIONI O SEGUITE



```

IF CONDIZIONE 1:
    ...
ELSE:
    ...

```

SE LA CONDIZIONE E' VERA

SE LA CONDIZIONE E' FALSA

ELIF: CI CONSENTI DI INSERIRE PRIMA DELL'ELSE

```

IF CONDIZIONE 1:
    ...
ELSE IF CONDIZIONE N:
    ...
ELSE:
    ...

```

RANDOM

IMPORT RANDOM

1) v = RANDOM.RANDRANGE (0, 100)

2) v = RANDOM.RANDOM() 0-1

WHILE (CICLO)

IN ALTERNATIVA ALLA RICORSIONE

DEF CONTROLLOSCIA(N):

```
WHILE N > 0:
    PRINT N
    N = N - 1
```

CONDIZIONE

SE E' VERA CONTINUA L'ESecuzione
SE FALSA ESCI DAL CICLO

```
PRINT ("VIA")
```

BREAK = FUNZIONE CHE INTERRUPE IL CICLO E NE ESCI

FOR (CICLO)

SI UTILIZZA PRINCIPALMENTE CON OGGETTI ITERABILI (LISTE, TUPLE, DIZIONARI)

```
S = [1, 2, 3, 4] OPERATORE LOGICO (N IN S)
```

```
FOR N IN S:
```

```
    PRINT ("IL QUADRATO DI", N, "E'", N^2)
```

VARIABILE CHE SI INCREMENTA

RANGE() = FUNZIONE CHE PERMETTE DI SPECIFICARE UN VALORE INIZIALE O UNO FINALE

```
LIST (RANGE(5, 10)) = [5, 6, 7, 8, 9]
```

```
FOR N IN RANGE(2, 6) N = (2, 3, 4, 5)
```

```
    PRINT ("IL QUADRATO E'", N^2)
```

LISTE []

LISTA = [⁰"c", ¹"i", ²"a", ³"o"]

SLICING

LISTA [1:3] -> ["i", "a", "o"]

LEN (LUNGHEZZA)

LEN(LISTA) = 4

OPERAZIONI SULLE LISTE

1) APPEND

LISTA.APPEND(ELEMENTO) = AGGIUNGO ELEMENTO ALLA FINE

2) EXTEND

LISTA.EXTEND(SERIE) = ESTENDO LA LISTA AGGIUNGENDO ALLA FINE LA SERIE

3) INSERT

LISTA.INSERT(INDICE, ELEMENTO) = AGGIUNGO L'ELEMENTO IN POSIZIONE INDICE

4) POP

LISTA.POP() = RIMOVO E RESTITUISCO L'ULTIMO ELEMENTO

5) REMOVE

LISTA.REMOVE(ELEMENTO) = RIMOVO ELEMENTO DALLA LISTA

6) SORT

LISTA.SORT() = ORDINA IN MODO CROSCENTE LA LISTA

1) REVERSE

LISTA . REVERSE () = INVERTE L'ORDINE DEGLI ELEMENTI

2) COPY

LISTA . COPY () = CREA E RITORNA UNA COPIA DELLA LISTA

3) CLEAR

LISTA . CLEAR () = RIMOVI TUTTI GLI ELEMENTI

LISTE COLLE PILE / CODE

LIFO (LAST - IN, FIRST - OUT)

- PER AGGIUNGERE UN ELEMENTO IN UNA LISTA USARE APPEND ()
 - PER OTTENERE UN ELEMENTO ALLA SOMMA SI USA POP ()
 ↳ SENZA INDICE

FIFO (FIRST - IN, FIRST - OUT)

- PER AGGIUNGERE SEMPRE APPEND ()
 - PER OTTENERE L'ELEMENTO IN TESTA SI USA POP (0)
 ↳ INDICE

TUPLE

SEQUENZA IMMUTABILE DI OGGETTI
↓
FISSO

ASSOMIGLIANO ALLE LISTE

Indice 0 1 2 3
T = ("A", "B", "C", "D") → CON UN SOLO ELEMENTO T = "A"

FUNZIONE TUPLE

T = TUPLE ("CIAO")
PRINT (T) } ("C", "I", "A", "O")

- SUPPORTA LO SLICING

PRINT T [1:3] → ("I", "O")

METODO SPLIT

DIVIDE IN 2 ELEMENTI UNA LISTA

EMAIL = "PAIS. MARCO @ GMAIL.COM"

NOME, DOMINIO = EMAIL . SPLIT("@")

↓ CARATTERE PER DIVIDERE

NOME
DOGLI

VARIABILI
ELEMENTI

PRINT NOME → PAIS. MARCO @

PRINT DOMINIO → GMAIL.COM

FUNZIONE ZIP

- RICEVE 2 O + SEQUENZE E LE ABBINA

S = "A BC"

T = [0, 1, 2]

ZIP (S, T) → [("A", 0), ("B", 1), ("C", 2)]

DIZIONARI

EN-IT ["ONE"] = "UNO"
↑
CHIAVE

↑
VALORE

POSSIBILI UNA RELAZIONE TRA
CHIAVE - VALORE
↓
INDICE

EN-IT {} → DIZIONARIO VUOTO

~~EN-IT~~ EN-IT = DICT() → FUNZIONE

EN-IT = { "ONE": "UNO", "TWO": "DUE", "THREE": "TRE" }
↑
CHIAVI
↑
VALORE

ERRORS

- VISUALIZZAZIONE GENERATA SO LA CHIAVE NON È PRESSIONE

PRINT EN-IT ["FOUR"]

ERRORE

FUNZIONE LEN

LEN EN-IT → 3

OPERATORI IN

INFORMA SE "QUALCOSA" COMPARE COME CHIAVE

"ONE" IN EN-IT → TRUE

"CART" IN EN-IT → FALSE

ISTOGRAMMA

- TORNARE PER INDICARE UN INSISTENZA DI CONTATORI

DEF ISTOGRAMMA (S)

D = DICT()

FOR C IN S

IF C NOT IN D
D[C] = 1

* ELSE
D[C] += 1

RETURN D

FOR SU DIZIONARIO

• UN FOR CHE SCORRE SULLO CHIAMI

METODI SUI DIZIONARI:

1) ITEMS ()
D.items() → RESTITUISCE GLI ELEMENTI DI D COME TUPLE

2) KEYS ()
D.keys() → RESTITUISCE LE CHIAVI

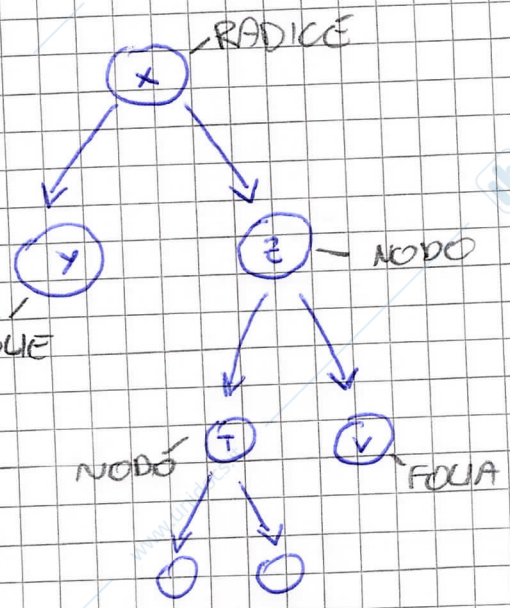
3) VALUES ()
D.values() → RESTITUISCE I VALORI

4) GET ()
D.get(CHIAVE, V) → RESTITUISCE IL VALORE DI CHIAVE SE PRESENTE ALTRIMENTI V O NONE

5) POP ()
D.pop(CHIAVE, V) → RIMUOVE E RESTITUISCE IL VALORE CORRISPONDENTE A CHIAVE ALTRIMENTI IL V

6) UPDATE ()
D.update(D2) → AGGIUNGE GLI ELEMENTI DI D2 A D

ALBERO



PROFONDITA = DISTANZA DI UN NODO DALLA RADICE

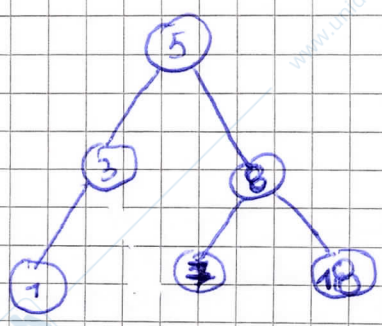
h DI UN ALBERO = DISTANZA TRA LA RADICE E L'ULTIMO NODO

LIVELLO = INSIEME DI NODI ALLA STESSA PROFONDITA

GRADO D'USCITA = QUANTITA DI SUCCESSORI DI UN NODO

ALBERO BINARIO = UN ALBERO IN CUI OGNI NODO HA AL MASSIMO 2 SUCCESSORI

DI RICERCA = QUANDO IL VALORE DI OGNI NODO E MAGGIOR O UGUALE RISPETTO A TUTTI I NODI DEL SOTTO ALBERO SINISTRO E MINORE DI QUELLO DEL SOTTO ALBERO DESTRO



CLASSE NODO:

CLASS NODE:

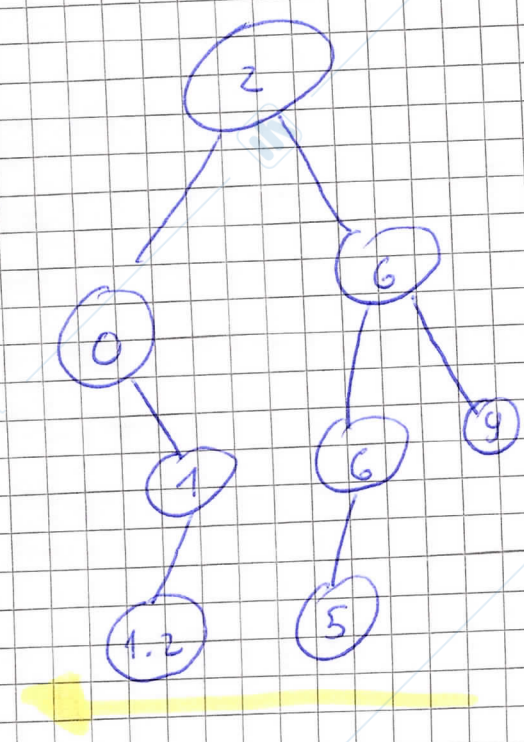
```

def __init__(self, data):
    self.right = None
    self.left = None
    self.data = data
    if data > n.data:
        return search(n.right, data)
  
```

VISITA ITERATIVA

DA DESTRA VERSO SINISTRA

2, 6, 9, 6, 5, 0, 1, 1.2



VISITA IN AMPIEZZA

DAL ALTO VERSO IL BASSO

2, 0, 6, 1, 6, 9, 1.2, 5

FUNZIONI

COME CREARE UNA FUNZIONE

CHIAMO FUNZIONE

NOME PARAMETRI

DEF SOMMA (N1, N2):

S = N1 + N2

PRINT ("LA SOMMA DEI NUMERI E'", S)

FUNZIONE SUM

CHIAMO FUNZIONE

NUMERO_A = INPUT("INSERISCI IL PRIMO NUMERO")

NUMERO_B = INPUT("INSERISCI IL SECONDO NUMERO")

N_A = FLOAT(NUMERO_A)

N_B = FLOAT(NUMERO_B)

SOMMA(N_A, N_B)

→ RICHIAMO FUNZIONE SOMMA
O PASSO I 2 NUMERI

IMPORTARE LE FUNZIONI

IMPORTIAMO LA LIBRERIA MATEMATICA PER USARE LE VARIE FUNZIONI

IMPORT MATH PRIMA DEL CODICE

↓ OTTENIAMO

MATH.PI → PI GRECO

MATH.COS(x) → COSENO X

MATH.SQRT(x) → RADICE QUADRATA X

- FUNZIONI PRODUTTIVE

FUNZIONE CHE RESTITUISCE UN VALORE

UTILIZZA IN FUNZIONE

RETURN VARIABLE

MATRICI (LISTE DI LISTE)

```

M = [ ] LISTA
FOR I IN RANGE(2): → RIGHE
    M.append(LEN(M))

```

```

N = [ ] SOTTOLISTA
FOR J IN RANGE(2): → COLONNE
    N.append(LEN(N))

```

```

x = INPUT + (F "INSERISCI ELEMENTO { I, J } : ")
N.append(x)

```

```

M.append(N)

```

```

PRINT ("LA MATRICE E' ", {M})

```

