CORSO PHYTON

In [50]:
```
x = 20   # dicharo variabile x
print(x) # stampa
```

```
20
```

In [51]:
```
y = [1, 2, 3, 4] # tipo di lato letteral
print(y)
```

```
[1, 2, 3, 4]
```

In [52]:
```
x = 2000;
y = 7
```

In [53]:
```
x * y
```

Out[53]:
```
14000
```

In [54]:
```
x + y
```

Out[54]:
```
2007
```

In [55]:
```
x - y
```

Out[55]:
```
1993
```

In [56]:
```
x / y
```

Out[56]:
```
285.7142857142857
```

In [57]:
```
y = 1.0
type (y)
```

Out[57]:
```
float
```

In [58]:
```
x // y
```

Out[58]:
```
2000.0
```

In [59]:
```
x ** y
```

Out[59]:
```
2000.0
```

In [61]:
```
x = 8.0
y = 6
x+y
```

Out[61]:
```
14.0
```

TIPI NUMERICI

i tipi di dati in PHYTON sono:

interi floating point boolean

In [62]:
```
x = 20000000000
y = 2_00_000_0000_000
```

In [63]:
```
x = 2e12 # notazione scentifica numero float
print (x)
```

```
2000000000000.0
```

In [64]:
```
x = 0xABCD # esadecimale x chiave
y = 0o765   # ottale o chiave
z = 0b010101010

print (x)
print (y)
print (z)
```

```
43981
501
170
```

In [65]:
```
x = 0b10101010000101010  # numero binario in decimale b chiave binario
print (x)
```

```
87082
```

In [66]:
```
stringa = "buongiorno"
```

In [67]:
```
stringa[6]  #lettera posizione 6 stringa
```

Out[67]:
```
'o'
```

In [68]:
```
x = 5.6
print (x)
```

```
5.6
```

In [69]:
```
stringa [:6] # da carattere 6 in poi
```

Out[69]:
```
'buongi'
```

In [70]:
```
stringa [6:] # da carattere 6 in poi
```

Out[70]:
```
'orno'
```

In [71]:
```
stringa [1:4:2] # parte primo carattere arriva fino a quarto escluso con passaggi
```

Out[71]:
```
'un'
```

In [72]:
```
stringa.upper() #lettere stringa tutte in maiuscolo
```

Out[72]:
```
'BUONGIORNO'
```

In [73]:
```
stringa.lower() #lettere tutte in minuscolo
```

Out[73]:
```
'buongiorno'
```

slicing

stringa[start:stop:step]

tipo lista

In [74]:
```
mylist = []     # normale
mylist = list() # costruttore oggetto liste
```

In [75]:
```
mylist = [1 ,2 , 3, 4] # mia lista
```

In [76]:
```
mylist[2] # posto 2 stringa
```

Out[76]:
```
3
```

In [77]:
```
mylist [-1] # parte dal dal ultimo posto
```

Out[77]:
```
4
```

In [78]:
```
mylist = [[1,2] , [3,4], [5,6]] # più liste
```

In [79]:
```
mylist [2][0]
```

Out[79]:
```
5
```

In [80]:
```
mylist.insert(2, "ciao") # aggiungo in posizione 2 ciao
```

In [82]:
```
mylist # vedo mia lista
```

Out[82]:
```
[[1, 2], [3, 4], 'ciao', [5, 6]]
```

In [83]:
```
del mylist [2] # rirorno lista prima elimino parte 2
```

In [84]:
```
mylist
```

Out[84]:
```
[[1, 2], [3, 4], [5, 6]]
```

In [86]:
```
mylist.insert(2, ["ciao", 5,6])
```

In [87]:
```
mylist
```

Out[87]:
```
[[1, 2], [3, 4], ['ciao', 5, 6], ['ciao', 5, 6], [5, 6]]
```

In [89]:
```
del mylist [2]
```

In [90]:
```
mylist
```

Out[90]:
```
[[1, 2], [3, 4], ['ciao', 5, 6], [5, 6]]
```

In [91]:
```
mylist
```

Out[91]:
```
[[1, 2], [3, 4], ['ciao', 5, 6], [5, 6]]
```

In [92]:
```
mylist.insert(0, "ciao")
```

In [93]:
```
mylist
```

Out[93]:
```
['ciao', [1, 2], [3, 4], ['ciao', 5, 6], [5, 6]]
```

In [102…
```
mylist1 = [10,20,30]
mylist2 = mylist1
mylist2[2] = 50
mylist1 # uguale deve dare 10,20,50
```

Out[102]:
```
[10, 20, 50]
```

In [103…
```
mylist1 = [10,20,30]
mylist2 = mylist1.copy()
mylist[2] = 50
mylist1 # lascia il valore di mylast1
```

Out[103]:
```
[10, 20, 30]
```

In [105…
```
mylist1.append(70) #mette in ultimo posto disponibile
```

In [106…
```
mylist1
```

Out[106]:
```
[10, 20, 30, 70]
```

TUPLE

In [109…
```
medaglie = ()       # normale
medaglie = tuple()  # costruttore
```

In [110…
```
medaglie = ("oro" , "argento", "bronzo") #lista non modificabile visualizzabile
```

In [113…
```
o, a, b = medaglie # etichette comode per dati che devo tirare fuori unpacching
```

In [116…
```
o
```

Out[116]:
```
'oro'
```

DIZIONARI

In [159…
```
myDict = {}     # normale
myDict = dict() # costruttore
mydict = {"a": 10 , "b" : 20 , "c" :30} # mio dizionario con chiave e numero
myDict["a"]
myDict ["d"] = 70
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13484\1460399286.py in <module>
      2 myDict = dict() # costruttore
      3 mydict = {"a": 10 , "b" : 20 , "c" :30} # mio dizionario con chiave e nume
ro
----> 4 myDict["a"]
      5 myDict ["d"] = 70

KeyError: 'a'
```

In [124…
```
"b " in myDict #vedi se c'è chiave nel dizionario
50 in myDict.values()
30 in myDict.values()
```

```
  File "C:\Users\Utente\AppData\Local\Temp\ipykernel_13484\3085335829.py", line 1
    myDict = {"d" . 40}
                    ^
SyntaxError: invalid syntax
```

In [132…
```
myDict
```

Out[132]:
```
{'a'}
```

SET

In [133…
```
myset = set ()
```

In [134…
```
myset = {10,20,30,40,10}
```

In [135…
```
myset
```

Out[135]:
```
{10, 20, 30, 40}
```

In [136…
```
myset.add("ciao") # aggiungi dati
```

In [137…
```
myset
```

Out[137]:
```
{10, 20, 30, 40, 'ciao'}
```

In [138…
```
myset2 = {10 ,30 ,50}
```

In [139…
```
myset | myset2 # unione
```

Out[139]:
```
{10, 20, 30, 40, 50, 'ciao'}
```

In [140…
```
myset & myset2 # intersezione
```

Out[140]:
```
{10, 30}
```

In [141…
```
myset - myset2 # differenza
```

Out[141]:
```
{20, 40, 'ciao'}
```

In [142…
```
myset ^ myset2 # differenza simmetrica
```

Out[142]:
```
{20, 40, 50, 'ciao'}
```

STATEMENT IF

if condizione:

    condizione1

elif

```
    condizione2
```

else:

```
    condizione3
```

In [149…
```
x = input("inserisci un mumero: ") #inserisci numero
x = int(x)

if x % 2 == 0:                          # if
    print(f"il numero {x} è pari")     # f intende valore variabile x
else:
    print(f"il numero {x} è dispari")
```

```
inserisci un mumero: 59
il numero 59 è dispari
```

In [157…
```
y = input("inserisci un mumero: ") #inserisci numero
y = int(y)

if y < 10:          # if
    print("<10")    # f intende valore variabile x

elif y < 20:
    print("<10 e >=10")

elif y < 30:               # elif è come lo switch di c#
    print("<30 e >=20")
else:
    print(">=30")
```

```
inserisci un mumero: 25
<30 e >=20
```

STATEMENT WHILE

while condizione:

```
    condizione1
```

else:

```
    condizione2
```

In [ ]:
```
x = 0
while x < 3:
    print(x)
    x += 1
else:
    print("false")
```

In [ ]:
```
while true:
    x = input("inserisci una stringa")
    if x == "STOP":
        break;
```

```
    elif x < "b":
        continue
    print(x)
else:
    print("fine")
```

STATEMENT FOR

for indice in iterabile:

    condizione 1

else:

    condizione2

In [ ]:
```
mylist = {1, 2, 3, 4}

for i in mylist
    prrint(i)
```

In [ ]:
```
string = "buongiorno"
for _ in stringa:
    print(_ , end = " ") # aggiungi spazio con end tra due variabili
```

In [ ]:
```
myDict
```

In [ ]:
```
for _ in myDict:
    print(_)
```

In [ ]:
```
for _ in myDict.values():  # stampa chiavi
    print (_)
```

In [ ]:
```
for _ in myDict.temi().   # stampa numeri dizionario
      print(_)
```

In [ ]:
```
for _ int range (11):
    print(_)         # stampa fino a 11
```

In [ ]:
```
for _ int range (6,15,2):
    print(_)
```

In [ ]:
```
numero1 = input ("inserisci il primo numero :")
numero1 = int (numero1)
numero2 = input ("inserisci il secondo numero :")
numero2 = int (numero2)

if numero1 %2 == 0
```