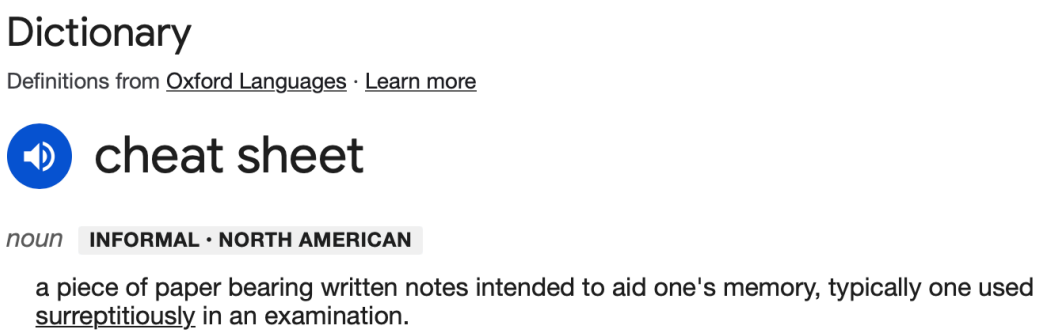
Corso ITS: *ARTIFICIAL INTELLIGENCE SPECIALIST*

Modulo: **Programmazione ad oggetti in Python e librerie esterne**

Docente: *Andrea Ribuoli*

Giovedì **13 Marzo** 2025

08:30 - 14:30



Python cheat sheet

In [1]:
```
0b101 == 5
```

Out[1]: True

In [2]:
```
0b000101 == 5
```

Out[2]: True

In [3]:
```
0b10 == 2    ##  0, 1
```

Out[3]: True

In [4]:
```
0o10 == 8    ##  0, 1, 2, 3, 4, 5, 6, 7
```

Out[4]: True

In [7]:
```
10 == 10    ##  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

Out[7]: True

In [6]:
```
0x10 == 16   ##  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
```

Out[6]: True

In [8]:
```
0x123 == 1 * 256 + 2 * 16 + 3 * 1
```

Out[8]: True

In [9]:
```
31 == 0x1F  ## 0xA == 10, 0xB == 11, 0xC == 12, 0xD == 13, 0xE == 1
```

Out[9]: True

In [10]:
```
True = 5
```

```
  Input In [10]
    True = 5
    ^
SyntaxError: cannot assign to True
```

In [12]:
```
True == 5
```

Out[12]: False

In [11]:
```
z5 = True
```

In [17]:
```
a = """
  Questo testo
lo sto assegnando
alla variabile a
"""
print(a)
```

```
  Questo testo
lo sto assegnando
alla variabile a
```

In [18]:
```
a = """
a = 0x1F
for x in range(0, a) :
    print(a)
"""
print(a)
```

```
a = 0x1F
for x in range(0, a) :
    print(a)
```

In [35]:
```
png = open("cheatsheet.png", "rb")
a = png.read(26)
print(a)
```

```
b'\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x05V\x00\x00\x01\xc2\x
08\x06'
```

In [37]:
```
euro = "€"
b = b'\xE2\x82\xAC'
euro == b.decode()
```

Out[37]: True

In [39]:
```
città = "Pesaro"
print(città)
```

```
Pesaro
```

In [40]:
```
print(0x123)
```

```
291
```

In [41]:
```
print(int("123"))
```

```
123
```

In [42]:
```
print(int("123", 16)) ## 256 + 32 + 3
```

```
291
```

In [43]:
```
print(int("123", 8))  ## 64 + 16 + 3
```

```
83
```

In [44]:
```
print(int("123", 4))  ## 16 + 8 + 3
```

```
27
```

In [45]:
```
bool(2)
```

Out[45]: True

In [46]:
```
bool(-2)
```

Out[46]: True

In [47]:
```
bool(0)
```

Out[47]: False

In [48]:
```
bool(["a"])
```

Out[48]: True

In [49]:
```
bool([])
```

Out[49]: False

In [50]:
```
bool(False)
```

Out[50]: False

In [51]:
```
bool(set())
```

Out[51]: False

In [52]:
```
bool(set('a'))
```

Out[52]: True

---

conversione da codepoint Unicode a notazione UTF-8

In [75]:
```
codepoint = ord("€")
cifre = list(f"{codepoint:b}")
cifre.insert(0, '0')
cifre.insert(0, '0')
print(cifre)
len(cifre)
cifre.insert(0, '0')
cifre.insert(0, '1')
cifre.insert(0, '1')
cifre.insert(0, '1')
cifre.insert(8, '0')
cifre.insert(8, '1')
cifre.insert(16, '0')
cifre.insert(16, '1')
print(cifre)
utf8_chars = "".join(cifre)
utf8_int = int(utf8_chars, 2)
print(utf8_int, int("E282AC", 16))
```

```
['0', '0', '1', '0', '0', '0', '0', '0', '1', '0', '1', '0', '1', '
1', '0', '0']
['1', '1', '1', '0', '0', '0', '1', '0', '1', '0', '0', '0', '0', '
0', '1', '0', '1', '0', '1', '0', '1', '1', '0', '0']
14844588 14844588
```

In [77]:
```
dict([(3,"three"),(1,"one")])
```

Out[77]: {3: 'three', 1: 'one'}

In [78]:
```
dict([("Silvia","13/3"),("Juan","13/3")])
```

Out[78]:
```
{'Silvia': '13/3', 'Juan': '13/3'}
```

In [79]:
```
set(["one","two"])
```

Out[79]:
```
{'one', 'two'}
```

In [80]:
```
set(["one","two","one"])
```

Out[80]:
```
{'one', 'two'}
```

In [81]:
```
print("one","two")
```

```
one two
```

In [82]:
```
print("one","two", sep="")
```

```
onetwo
```

In [83]:
```
print("one","two", sep="")
print("one","two", sep="")
```

```
onetwo
onetwo
```

In [84]:
```
print("one","two", sep="", end="")
print("one","two", sep="", end="")
```

```
onetwoonetwo
```

In [88]:
```
outfile = open("esito.txt", "w")
print("uno","due", file=outfile)
print("tre","quattro", file=outfile)
print("cinque","sei", file=outfile)
outfile.close()
```

In [92]:
```
outfile = open("esito.txt", "w")
print("uno","due", file=outfile)
outfile.close()
outfile = open("esito.txt", "a")
print("tre","quattro", file=outfile)
outfile.close()
outfile = open("esito.txt", "a")
print("cinque","sei", file=outfile)
outfile.close()
```

In [94]:
```
with open("esito.txt", "r") as n :
    for riga in n :
        print(riga, end="")
```

```
uno due
tre quattro
cinque sei
```

In [95]:
```
with open("esito.txt", "r") as n :
```

```
    for riga in n :
        for w in riga.split() :
            print(w)
```

```
uno
due
tre
quattro
cinque
sei
```

In [96]:
```
"1,4,8,2".split(",")
```

Out[96]:
```
['1', '4', '8', '2']
```

In [97]:
```
[int(x) for x in ('1','29','-3')]
```

Out[97]:
```
[1, 29, -3]
```

In [98]:
```
[x*2 for x in ('1','29','-3')]
```

Out[98]:
```
['11', '2929', '-3-3']
```

In [99]:
```
[int(x)*2 for x in ('1','29','-3')]
```

Out[99]:
```
[2, 58, -6]
```

In [101…
```
a=b=c=-1
print(a, b, c)
```

```
-1 -1 -1
```

In [106…
```
a=b=c="ciao"
print(a, b, c)
```

```
ciao ciao ciao
```

In [102…
```
a=-1
b=-1
c=-1
print(a, b, c)
```

```
-1 -1 -1
```

In [107…
```
y,z,r = 9.2 , "ciao" , 0
print(y, z, r)
```

```
9.2 ciao 0
```

In [108…
```
(y,z,r) = (9.2 , "ciao" , 0)
print(y, z, r)
```

```
9.2 ciao 0
```

In [109…
```
a = 2
b = 3
print(a, b)
```

```
c = a
a = b
b = c
print(a, b)
```

```
2 3
3 2
```

In [110…

```
a = 2
b = 3
print(a, b)
a, b = b, a
print(a, b)
```

```
2 3
3 2
```

In [120…

```
codepoint = ord("€")
cifre = list(f"{codepoint:b}")
cifre.insert(0, '0')
cifre.insert(0, '0')
cifre.insert(0, '0')
cifre.insert(0, '1')
cifre.insert(0, '1')
cifre.insert(0, '1')
cifre.insert(8, '0')
cifre.insert(8, '1')
cifre.insert(16, '0')
cifre.insert(16, '1')
print(cifre)
print(cifre[-3:])
print(cifre[16:])
print(cifre[-8:])
print(cifre[16:20])
print(cifre[-8:-4])
print(cifre[16:-4])
print(cifre[-8:20])
print(cifre[16:20:2])
print(cifre[::-1])
```

```
['1', '1', '1', '0', '0', '0', '1', '0', '1', '0', '0', '0', '0', '
0', '1', '0', '1', '0', '1', '0', '1', '1', '0', '0']
['1', '0', '0']
['1', '0', '1', '0', '1', '1', '0', '0']
['1', '0', '1', '0', '1', '1', '0', '0']
['1', '0', '1', '0']
['1', '0', '1', '0']
['1', '0', '1', '0']
['1', '0', '1', '0']
['1', '1']
['0', '0', '1', '1', '0', '1', '0', '1', '0', '1', '0', '0', '0', '
0', '0', '1', '0', '1', '0', '0', '0', '1', '1', '1']
```

In [121…

```
note = ["do", "re", "mi", "fa", "sol", "la", "si"]
len(note)
print(sorted(note))
print(note)
```

In [125…
```
note = ("do", "re", "mi", "fa", "sol", "la", "si")
len(note)
print(sorted(note))
print(note)
```

```
['do', 'fa', 'la', 'mi', 're', 'si', 'sol']
('do', 're', 'mi', 'fa', 'sol', 'la', 'si')
```

In [129…
```
multiplo = [ True, False, True, False ]
print(all(multiplo))
print(any(multiplo))
```

```
False
True
```

In [132…
```
note = ["do", "re", "mi", "fa", "sol", "la", "si"]
for n in reversed(note) :
    print(n)
```

```
si
la
sol
fa
mi
re
do
```

In [135…
```
note.index("fa")
```

Out[135… 3

In [136…
```
note.index("fa", 1)
```

Out[136… 3

In [137…
```
note.index("fa", 5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [137], in <cell line: 1>()
----> 1 note.index("fa", 5)

ValueError: 'fa' is not in list
```

appuntamento.py

Implementate la classe **Appuntamento** e le tre sottoclassi:

- **Occasionale**
- **Annuale**
- **Mensile**

Un appuntamento ha una descrizione e una data.

Scrivete il metodo **occursOn(anno, mese, giorno)** che verifica se un appuntamento avviene in una certa data. Deve funzionare anche per appuntamento con ricorrenza.