Corso ITS:

PROGETTISTA E SVILUPPATORE SOFTWARE:

FULL STACK DEVELOPER E CLOUD SPECIALIST

Modulo: **Programmazione in Python**

Docente: *Andrea Ribuoli*

Mercoledì **30 Aprile** 2025

09:00 - 13:00

13:30 - 16:30

In [2]:
```python
a = False
b = False
print(f"a = {a}, b = {b}, a and b = {a and b}")
a = True
b = False
print(f"a = {a}, b = {b}, a and b = {a and b}")
a = False
b = True
print(f"a = {a}, b = {b}, a and b = {a and b}")
a = True
b = True
print(f"a = {a}, b = {b}, a and b = {a and b}")
```

```
a = False, b = False, a and b = False
a = True, b = False, a and b = False
a = False, b = True, a and b = False
a = True, b = True, a and b = True
```

a	b	a and b
0	0	0
0	1	0
1	0	0

a	b	a and b
1	1	1

In [7]:
```
a = False
b = False
print(f"a = {a}, b = {b}, not (a or b) = {not (a or b)}")
a = True
b = False
print(f"a = {a}, b = {b}, not (a or b) = {not (a or b)}")
a = False
b = True
print(f"a = {a}, b = {b}, not (a or b) = {not (a or b)}")
a = True
b = True
print(f"a = {a}, b = {b}, not (a or b) = {not (a or b)}")
```

```
a = False, b = False, not (a or b) = True
a = True, b = False, not (a or b) = False
a = False, b = True, not (a or b) = False
a = True, b = True, not (a or b) = False
```

a	b	a or b
0	0	0
0	1	1
1	0	1
1	1	1

In [8]:
```
a = False
b = False
print(f"a = {a}, b = {b}, (not a) and (not b) = {(not a) and (not b)}")
a = True
b = False
print(f"a = {a}, b = {b}, (not a) and (not b) = {(not a) and (not b)}")
a = False
b = True
print(f"a = {a}, b = {b}, (not a) and (not b) = {(not a) and (not b)}")
a = True
b = True
print(f"a = {a}, b = {b}, (not a) and (not b) = {(not a) and (not b)}")
```

```
a = False, b = False, (not a) and (not b) = True
a = True, b = False, (not a) and (not b) = False
a = False, b = True, (not a) and (not b) = False
a = True, b = True, (not a) and (not b) = False
```

In [5]:
```
a = False
b = False
print(f"a = {a}, b = {b}, not(a or b) = {not (a or b)}")
```

```
a = True
b = False
print(f"a = {a}, b = {b}, not(a or b) = {not (a or b)}")
a = False
b = True
print(f"a = {a}, b = {b}, not(a or b) = {not (a or b)}")
a = True
b = True
print(f"a = {a}, b = {b}, not(a or b) = {not (a or b)}")
```

```
a = False, b = False, not(a or b) = True
a = True, b = False, not(a or b) = False
a = False, b = True, not(a or b) = False
a = True, b = True, not(a or b) = False
```

In [10]:
```
print("not(a or b) <== identiche ==> (not a) and (not b)")
print("not(a and b) <== identiche ==> (not a) or (not b)")
```

```
not(a or b) <== identiche ==> (not a) and (not b)
not(a and b) <== identiche ==> (not a) or (not b)
```

leggi di MORGAN

In [ ]:
```
import urllib.request
url = "https://www.andrearibuoli.it"
risultato = urllib.request.urlopen(url)
theBytes = risultato.read()
text = theBytes.decode()
import bs4
doc = bs4.BeautifulSoup(text)
print(doc.prettify())
```

spigolatura.py

pip install bs4

In [13]:
```
!python3 resoconto.py spigolatura
```

```
 1  Mirco  Azzolini
 2  Wallace  Bezerra Beretta
 3  Alexandru Razvan  Brasovianu
 4  Edoardo  Caprini
 5  Maryuri  Catozzi
 6  Federico  De Grandis
 7  Maikol  Freddari
 8  Sofia  Gaona
 9  Alessia  Gasparini
10  Enrico  Giorgi
11  Andrea  Kanakciu
12  Francesco  Marinelli
13  Filippo  Martino
14  Eleonora  Moroni
15  Norman  Muzi
16  Mattia  Roberti
17  Alessandro  Rovinelli
18  Davide  Sambughi
19  Maximiliano  Serafini
20  Giovanni  Sperandini
21  Alessio  Stomeo
22  Lesly Pierina  Vera Castillejo
```

In [ ]:
```
lista = dir(bs4.element.Tag)
for met in lista:
    if not met.startswith('__'):
        print(met)
```

In [16]:
```
html = """<html>
 <head>
  <title>
    CORSO PYTHON
  </title>
 </head>
 <body>
  <p>
    Facciamo un primo esempio
  </p>
  <p>
    di pagina HTML remota
    <a href="https://www.andrearibuoli.it">
      link
    </a>
  </p>
  <p>
    composta di paragrafi.
  </p>
 </body>
</html>
"""
```

In [17]:
```
print(html)
```

```
<html>
 <head>
  <title>
    CORSO PYTHON
  </title>
 </head>
 <body>
  <p>
    Facciamo un primo esempio
  </p>
  <p>
    di pagina HTML remota
    <a href="https://www.andrearibuoli.it">
      link
    </a>
  </p>
  <p>
    composta di paragrafi.
  </p>
 </body>
</html>
```

In [ ]:
```
html_compatto = "".join([x.strip() for x in html.split("\n")])
```

In [19]:
```
print(html_compatto)
```

```
<html><head><title>CORSO PYTHON</title></head><body><p>Facciamo un primo ese
mpio</p><p>di pagina HTML remota<a href="https://www.andrearibuoli.it">link
</a></p><p>composta di paragrafi.</p></body></html>
```

In [20]:
```
import bs4
doc = bs4.BeautifulSoup(html_compatto)
```

In [29]:
```
def naviga2(tag, indent) :
    print(indent + tag.name.upper())
    if tag.name.upper() == "A":
        print(tag.get("href"))
    for stag in tag.contents:
        if type(stag) == bs4.element.Tag :
            naviga2(stag, indent + " ")
```

In [31]:
```
def naviga3(tag) :
    if tag.name.upper() == "A":
        print(tag.get("href"))
    for stag in tag.contents:
        if type(stag) == bs4.element.Tag :
            naviga3(stag)
```

In [33]:
```
def naviga4(tag) :
    if tag.name.upper() == "A":
        print(tag["href"])
    for stag in tag.contents:
        if type(stag) == bs4.element.Tag :
            naviga4(stag)
```

In [34]:
```
naviga4(doc.contents[0])
```

```
https://www.andrearibuoli.it
```

dipendenze.py

In [3]:
```
import urllib.request
import bs4

def main():
    url = input("URL della pagina di partenza: ")
    visited = []
    depth = 4
    crawl(url, depth, visited)

def crawl(url, depth, visited) :
    if depth == 0:
        return
    response = urllib.request.urlopen(url)
    doc = bs4.BeautifulSoup(response)
    print(f"Sto visitando il percorso: '{url}'")
    try:
        for link in doc.find_all("a"):
            href = link["href"]
            if href[0:4] == "http" and href not in visited :
                visited.append(href)
                crawl(href, depth - 1, visited)
    except:
        return

main()
```

```
URL della pagina di partenza:  https://www.andrearibuoli.it
Sto visitando il percorso: 'https://www.andrearibuoli.it'
Sto visitando il percorso: 'https://www.andrearibuoli.it/wp'
Sto visitando il percorso: 'https://www.andrearibuoli.it/wp/'
Sto visitando il percorso: 'https://www.andrearibuoli.it/wp/sample-page/'
Sto visitando il percorso: 'https://www.andrearibuoli.it/wp/author/admin/'
```

In [ ]:
```
import urllib.request
url = "https://www.comuni-italiani.it/province.html"
response = urllib.request.urlopen(url)
theBytes = response.read()
text = theBytes.decode(encoding="iso-8859-1")

import bs4
doc = bs4.BeautifulSoup(text)
elems = doc.find_all("table")
table = elems[3]

def naviga2(tag, indent) :
    print(indent + tag.name.upper())
    for stag in tag.contents:
        if type(stag) == bs4.element.Tag :
            naviga2(stag, indent + " ")
```

```
naviga2(table, "")
```

In [ ]:
```
import urllib.request
url = "https://www.comuni-italiani.it/province.html"
response = urllib.request.urlopen(url)
theBytes = response.read()
text = theBytes.decode(encoding="iso-8859-1")

import bs4
doc = bs4.BeautifulSoup(text)
elems = doc.find_all("table")
table = elems[3]
for tr in table.contents[2:-1]:
    if type(tr) == bs4.element.Tag :
        tds = tr.contents
        print(tds[0])
```

In [35]:
```
import urllib.request
url = "https://www.comuni-italiani.it/province.html"
response = urllib.request.urlopen(url)
theBytes = response.read()
text = theBytes.decode(encoding="iso-8859-1")

import bs4
doc = bs4.BeautifulSoup(text)
elems = doc.find_all("table")
table = elems[3]
for tr in table.contents[2:-2]:
    if type(tr) == bs4.element.Tag :
        tds = tr.contents
        sequ = int(tds[0].get_text())
        prov =     tds[1].get_text()
        resi = int(tds[2].get_text().replace(".",""))
        sigl =     tds[7].get_text()
        print(f"{sequ:3d} {prov:25s} {resi:9d} {sigl}")
```

```
 1 Agrigento                    442049 AG
 2 Alessandria                  426658 AL
 3 Ancona                       474124 AN
 4 Aosta                        126883 AO
 5 Arezzo                       344374 AR
 6 Ascoli Piceno                209450 AP
 7 Asti                         216677 AT
 8 Avellino                     423506 AV
 9 Bari                        1260142 BA
10 Barletta-Andria-Trani        392546 BT
11 Belluno                      205781 BL
12 Benevento                    279675 BN
13 Bergamo                     1109933 BG
14 Biella                       178551 BI
15 Bologna                     1009210 BO
16 Bolzano                      524256 BZ
17 Brescia                     1262318 BS
18 Brindisi                     397083 BR
19 Cagliari                     560373 CA
20 Caltanissetta                269710 CL
21 Campobasso                   224644 CB
22 Carbonia-Iglesias            126324 CI
23 Caserta                      924166 CE
24 Catania                     1113303 CT
25 Catanzaro                    362343 CZ
26 Chieti                       389169 CH
27 Como                         600190 CO
28 Cosenza                      711739 CS
29 Cremona                      359388 CR
30 Crotone                      175566 KR
31 Cuneo                        589108 CN
32 Enna                         168052 EN
33 Fermo                        174849 FM
34 Ferrara                      348362 FE
35 Firenze                     1014423 FI
36 Foggia                       628556 FG
37 Forlì-Cesena                 394067 FC
38 Frosinone                    493067 FR
39 Genova                       850071 GE
40 Gorizia                      139673 GO
41 Grosseto                     223045 GR
42 Imperia                      215130 IM
43 Isernia                       85805 IS
44 La Spezia                    220698 SP
45 L'Aquila                     301910 AQ
46 Latina                       574891 LT
47 Lecce                        802082 LE
48 Lecco                        339238 LC
49 Livorno                      337334 LI
50 Lodi                         229338 LO
51 Lucca                        390042 LU
52 Macerata                     318921 MC
53 Mantova                      412610 MN
54 Massa-Carrara                196580 MS
55 Matera                       199685 MT
56 Messina                      636653 ME
```

```
 57 Milano                    3218201 MI
 58 Modena                     700862 MO
 59 Monza e della Brianza      868859 MB
 60 Napoli                    3107006 NA
 61 Novara                     370143 NO
 62 Nuoro                      156096 NU
 63 Olbia-Tempio               160672 OT
 64 Oristano                   160746 OR
 65 Padova                     936274 PD
 66 Palermo                   1268217 PA
 67 Parma                      448899 PR
 68 Pavia                      547251 PV
 69 Perugia                    660690 PG
 70 Pesaro e Urbino            360711 PU
 71 Pescara                    321309 PE
 72 Piacenza                   286758 PC
 73 Pisa                       421851 PI
 74 Pistoia                    291839 PT
 75 Pordenone                  312051 PN
 76 Potenza                    370680 PZ
 77 Prato                      254608 PO
 78 Ragusa                     321359 RG
 79 Ravenna                    391414 RA
 80 Reggio Calabria            553861 RC
 81 Reggio Emilia              532483 RE
 82 Rieti                      157420 RI
 83 Rimini                     336786 RN
 84 Roma                      4353738 RM
 85 Rovigo                     238588 RO
 86 Salerno                   1104731 SA
 87 Medio Campidano             98623 VS
 88 Sassari                    333116 SS
 89 Savona                     279408 SV
 90 Siena                      268341 SI
 91 Siracusa                   402822 SR
 92 Sondrio                    181437 SO
 93 Taranto                    583479 TA
 94 Teramo                     309859 TE
 95 Terni                      228218 TR
 96 Torino                    2277857 TO
 97 Ogliastra                   57185 OG
 98 Trapani                    434476 TP
 99 Trento                     538604 TN
100 Treviso                    885972 TV
101 Trieste                    234682 TS
102 Udine                      531466 UD
103 Varese                     890043 VA
104 Venezia                    854275 VE
105 Verbano-Cusio-Ossola       159664 VB
106 Vercelli                   173868 VC
107 Verona                     921557 VR
108 Vibo Valentia              161619 VV
109 Vicenza                    865082 VI
110 Viterbo                    319008 VT
```