

## ITS ICT - Ripasso Python - Teoria con esempi

### 1) Liste

- Mutabili. Elementi di tipi diversi.

Esempio:

```
lista = [1, "ciao", True]
```

```
lista.append(10)
```

```
lista.remove("ciao")
```

Slicing:

```
a = [10, 20, 30, 40, 50, 60, 70, 80]
```

```
print(a[2:5]) # [30, 40, 50]
```

```
print(a[-4:-1]) # [50, 60, 70]
```

Reverse:

```
a.reverse()
```

Clear:

```
a.clear()
```

pop:

```
a.pop()
```

### 2) Tuple

- Immutabili. Supportano slicing.

Esempio:

```
t = (100, 200, 300, 400, 500)
```

```
print(t[-4:-1]) # (200, 300, 400)
```

Concatenazione:

```
t1 = (1, 2)
```

```
t2 = (3, 4)
t3 = t1 + t2
```

### 3) Dizionari

- Coppie chiave: valore. Chiavi uniche.

Esempio:

```
diz = {"nome": "pippo", "classe": 5}
diz_vuoto = {}
```

Accesso:

```
print(diz["nome"])
print(diz.get("classe"))
```

Modifica/rimozione:

```
diz["classe"] = 6
del diz["classe"]
diz.pop("nome")
```

Metodi:

```
diz.clear()
```

### 4) Cicli for / while

```
for x in [1,2,3]:
    print(x)
```

```
for i in range(3):
    print(i)
else:
    print("Fine ciclo")
```

```
while x < 5:
```

```
x += 1  
print(x)
```

#### 5) \*args

```
def fun(*data):  
    for item in data:  
        print(item)  
  
fun(10, 20)  
fun(25, 75, 55)
```

#### 6) Comprehensions

```
a = [1, 2, 3, 4, 5]  
a2 = [2 * x for x in a]  
print(a2)
```

#### 7) Funzioni

```
def add(a, b):  
    return a + 5, b + 5  
  
result = add(3, 2)  
print(result) # (8, 7)
```

#### 8) Metodi utili di lista

append() - aggiunge un elemento  
remove() - rimuove il primo elemento uguale  
pop() - rimuove ultimo elemento  
clear() - svuota la lista  
reverse() - inverte gli elementi  
sort() - ordina crescente

## 9) Immutabilita'

Stringhe e tuple - immutabili

Liste e dizionari - mutabili

Errore comune:

```
var1 = 1
```

```
var2 = 2
```

```
var3 = "3"
```

```
print(var1 + var2 + var3) # Errore!
```

Altri slicing:

```
lista = [10, 20, 30, 40, 50, 60, 70, 80]
```

```
print(lista[:4])    # [10, 20, 30, 40]
```

```
print(lista[::-2])  # [80, 60, 40, 20]
```

Accesso annidato:

```
diz = {"class": {"student": {"name": "Mike", "marks": {"physics": 70, "history": 80}}}}
```

```
print(diz["class"]["student"]["marks"]["history"])
```

Attenzione ai tipi:

```
t = (1120, 'a')
```

```
# max(t) -> TypeError
```

Output ciclo nidificato:

```
x = 0
```

```
for i in range(10):
```

```
for j in range(-1, -10, -1):  
    x += 1  
print(x) # 90
```