

The background of the slide is a vibrant blue with a digital theme. It features floating binary code (0s and 1s) in a lighter blue shade. On the left side, there is a partial view of a laptop, and in the upper center, two server racks are visible. The main content is framed by a thick orange border.

# ***Python***

A teal-colored rectangular frame with a thin border, centered on the slide, enclosing the text for the second section.

## **Cicli FOR e WHILE**

# Introduzione

---

In Python esistono due tipi di cicli (anche detti *loop*):

- il ciclo **for**: esegue un'iterazione per ogni elemento di un iterabile;
- il ciclo **while**: itera fintanto che una condizione è vera.

# Ciclo FOR

---

- Il ciclo for ci permette di iterare su tutti gli elementi di un *iterabile* ed eseguire un determinato blocco di codice.
- Gli oggetti *iterabili* che abbiamo visto sono i seguenti:
  - liste,
  - tuple,
  - dizionari (restituiscono le chiavi)

# Ciclo FOR

---

```
sequenza = [3, 2, 5, 4, 7]
for i in sequenza:
    ...quadrato = i**2
    ...print(f"Il quadrato di {i} è {quadrato}")
```

```
Il quadrato di 3 è 9
Il quadrato di 2 è 4
Il quadrato di 5 è 25
Il quadrato di 4 è 16
Il quadrato di 7 è 49
```

# Ciclo FOR

---

```
for x in range(7):  
    print(x, end = " ")
```

```
0 1 2 3 4 5 6
```

# Ciclo FOR

---

```
for i in range(3, 6):  
    quadrato = i**2  
    print(f"Il quadrato di {i} è {quadrato}")
```

```
Il quadrato di 3 è 9  
Il quadrato di 4 è 16  
Il quadrato di 5 è 25
```

# Ciclo FOR

---

```
for x in range(2, 30, 3):  
    print(x, end = " ")
```

```
2 5 8 11 14 17 20 23 26 29
```

# Ciclo FOR

---

```
for x in range(20, 3, -3):  
    print(x, end = " ")
```

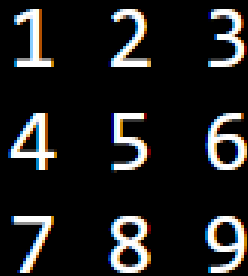
```
20 17 14 11 8 5
```



# Ciclo FOR

---

```
list_of_lists = [ [1, 2, 3], [4, 5, 6], [7, 8, 9]]  
for list in list_of_lists:  
    for x in list:  
        print(x, end = " ")  
    print("")
```



1	2	3
4	5	6
7	8	9

---

## **Ciclo WHILE**

# Ciclo WHILE

---

- Il ciclo while itera fintanto che una condizione è vera

```
numero = 4
while numero < 8:
    print(f"Numero: {numero}")
    numero += 1
```

```
Numero: 4
Numero: 5
Numero: 6
Numero: 7
```

# Ciclo WHILE

---

Il ciclo while itera fintanto che una condizione è vera

```
numero = 4
while numero < 8:
    print(f"Numero: {numero}")
    numero += 1
```

```
Numero: 4
Numero: 5
Numero: 6
Numero: 7
```

# Ciclo WHILE

---

```
# chiedi all'utente di inserire numeri finchè indovina
numero_inserito = 0
numero_da_indovinare = 8
while numero_inserito != numero_da_indovinare:
    ... numero_inserito = int(input('Inserire un numero: '))
print('Trovato!')
```

```
Inserire un numero: 3
Inserire un numero: 5
Inserire un numero: 8
Trovato!
```

# Break e continue

---

***break***: interrompe il ciclo

```
sequenza = [3, 2, 5, 4, 7]
for i in sequenza:
    if i == 5:
        break
    quadrato = i**2
    print(f"Il quadrato di {i} è {quadrato}")
```

```
Il quadrato di 3 è 9
Il quadrato di 2 è 4
```

# Break e continue

---

***break***: interrompe il ciclo

```
# chiedi all'utente di inserire numeri finchè indovina
numero_da_indovinare = 8
while True:
    guess = int(input('Inserisci un numero da 1 a 10: '))
    if guess == numero_da_indovinare:
        print('Hai indovinato!')
        break # numero indovinato, interrompi il ciclo
    else:
        print('Ritenta sarai più fortunato')
```

# Cicli FOR annidati

---

```
combinazioni = []  
for i in 'ABC':  
    for j in '123':  
        combinazioni.append(i + j)  
print(combinazioni)
```

```
['A1', 'A2', 'A3', 'B1', 'B2', 'B3', 'C1', 'C2', 'C3']
```



# Cicli FOR annidati

---

```
for i in range(1,5):  
    for j in range(1,5):  
        # end=' '  
        # permette di rimanere sulla stessa riga  
        print(i + j, end=' ')  
    # Passa alla riga successiva  
    print()
```

2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8