

1. IF, ELIF ed ELSE

Come si Scrive

IF semplice - Si scrive così:

```
if condizione:  
    codice_da_eseguire
```

Esempio:

```
num = 8  
if num % 2 == 0:  
    print("Questo numero è pari.")
```

IF-ELSE - Si scrive così:

```
if condizione:  
    codice_se_True  
else:  
    codice_se_False
```

Esempio:

```
eta = 20  
if eta < 18:  
    print("Sei minorenni")  
else:  
    print("Sei maggiorenne")
```

IF-ELIF-ELSE - Si scrive così:

```
if condizione1:  
    codice_condizione1  
elif condizione2:  
    codice_condizione2  
else:  
    codice_altrimenti
```

Esempio:

```
voto = 7  
if voto < 6:  
    print("Insufficiente")  
elif voto < 8:  
    print("Sufficiente")  
else:  
    print("Ottimo!")
```

Altri Esempi Pratici

Esempio - Controllo età per patente:

```
eta = int(input("Quanti anni hai? "))  
if eta < 18:  
    print("Non puoi ancora prendere la patente.")
```

```
elif eta == 18:
print("Hai 18 anni! Finalmente puoi prendere la patente")
else:
print("Puoi prendere la patente!")
```

Esempio - Operatori logici AND:

```
patente = input("Hai la patente? (si/no) ")
auto = input("Hai la macchina? (si/no) ")
if patente == "si" and auto == "si":
print("Allora puoi guidare.")
else:
print("Non puoi guidare.")
```

Operatori Logici

- **AND:** entrambe le condizioni devono essere True
- **OR:** almeno una condizione deve essere True
- **NOT:** inverte il valore booleano

IF Annidati - Si scrive così:

```
if condizione_esterna:
codice
if condizione_interna:
codice_annidato
else:
altro_codice
else:
codice_alternativo
```

Esempio:

```
num = int(input("Dimmi un numero"))
if num > 0:
print("Il numero è positivo")
if num > 100:
print("ed è anche maggiore di 100")
else:
print("e non è maggiore di 100")
else:
print("Il numero è negativo")
```

2. LISTE

Come si Crea una Lista - Si scrive così:

```
nome_lista = [elemento1, elemento2, elemento3]
```

Esempi:

```
nomi = ["Marco", "Giulia", "Antonio", "Sofia"]
```

```
voti = [10, 4, 8, 5, 9]
lista_mista = ["Ciao", 24, 5.5, "ABCDEF"]
lista_vuota = []
```

Come si Accede agli Elementi - Si scrive così:

```
nome_lista[indice] # L'indice parte da 0!
```

Esempio:

```
nomi = ["Marco", "Giulia", "Antonio", "Sofia"]
print(nomi[0]) # Marco (primo elemento)
print(nomi[1]) # Giulia (secondo elemento)
print(nomi[3]) # Sofia (quarto elemento)
```

Aggiungere Elementi - Si scrive così:

```
nome_lista.append(elemento)
```

Esempio:

```
artisti_preferiti = []
artista = input("Dimmi un tuo artista preferito: ")
artisti_preferiti.append(artista)
print(artisti_preferiti) # ['Nome artista inserito']
```

Rimuovere Elementi - Si scrive così:

```
del nome_lista[indice]
```

Esempio:

```
lista_cibi = ["Pizza", "Cioccolata", "Carbonara", "Minestrone"]
del lista_cibi[2] # Rimuove "Carbonara" (posizione 2)
print(lista_cibi) # ['Pizza', 'Cioccolata', 'Minestrone']
```

Lunghezza Lista - Si scrive così:

```
len(nome_lista)
```

Esempio:

```
voti = [10, 4, 8, 5, 9]
print(len(voti)) # 5
numero_voti = len(voti)
print(f"Hai {numero_voti} voti") # Hai 5 voti
```

Scorrere Liste con While - Si scrive così:

```
posizione = 0
while posizione < len(nome_lista):
    print(nome_lista[posizione])
    posizione = posizione + 1
```

Esempio:

```
lista_citta = ["Verona", "Parigi", "Amsterdam", "Tokyo"]
count_pos = 0
```

```
while count_pos < len(lista_citta):
print(f"Posizione {count_pos}: {lista_citta[count_pos]}")
count_pos = count_pos + 1
```

Output:

Posizione 0: Verona

Posizione 1: Parigi

Posizione 2: Amsterdam

Posizione 3: Tokyo

Calcolo Media con Liste - Si scrive così:

```
somma = 0
posizione = 0
while posizione < len(lista):
somma = somma + lista[posizione]
posizione = posizione + 1
media = somma / len(lista)
```

Esempio:

```
lista_voti = [10, 4, 8, 5, 9]
count_pos = 0
somma_voti = 0
```

```
while count_pos < len(lista_voti):
somma_voti = somma_voti + lista_voti[count_pos]
count_pos = count_pos + 1
```

```
media_voti = somma_voti / len(lista_voti)
print(f"La media è {media_voti}")
```

3. CICLO WHILE

Come si Scrive - Si scrive così:

```
while condizione:
codice_da_ripetere
```

IMPORTANTE: La condizione viene verificata PRIMA di ogni iterazione.

Esempio Base - Contatore:

```
contatore = 0
while contatore < 10:
```

```
print(contatore)
contatore = contatore + 1
```

Output: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Esempio - Ripetere Stringa N Volte:

```
stringa_da_scrivere = input("Dimmi una stringa: ")
quante_volte = int(input("Quante volte vuoi che scriva questa stringa? "))
contatore = 1
```

```
while contatore <= quante_volte:
    print(stringa_da_scrivere)
    contatore = contatore + 1
```

Esempio - Indovina il Numero:

```
numero_da_indovinare = 47
numero_inserito = int(input("Prova ad indovinare il numero"))
```

```
while numero_da_indovinare != numero_inserito:
    print("Hai sbagliato!")
    if numero_inserito < numero_da_indovinare:
        print("Il numero da indovinare è più grande!")
    else:
        print("Il numero da indovinare è più piccolo!")
    numero_inserito = int(input("Prova ad indovinare il numero"))

print("Complimenti! Hai indovinato il numero!!!")
```

Print senza andare a capo

Normalmente print va a capo

```
print("Ciao")
print("sono")
print("Iacopo")
```

Per evitare di andare a capo

```
print("Ciao", end=" ")
print("sono", end=" ")
print("Iacopo")
```

4. CICLO FOR

Come si Scrive - Si scrive così:

```
for elemento in sequenza:  
    codice_da_ripetere
```

Spiegazione: Il ciclo FOR prende automaticamente ogni elemento dalla sequenza (lista, stringa, range) e lo mette nella variabile elemento.

Esempio Base:

```
nomi = ["Marco", "Giulia", "Sofia"]  
for nome in nomi:  
    print(nome)
```

Output:

Marco

Giulia

Sofia

Scorrere Liste - Confronto While vs For

Confronto While vs For:

Con WHILE

```
lista_citta = ["Verona", "Tokyo", "New York", "Parigi"]  
pos = 0  
while pos < len(lista_citta):  
    print(lista_citta[pos])  
    pos = pos + 1
```

Con FOR (molto più semplice!)

```
for citta in lista_citta:  
    print(citta)
```

Calcolare media con FOR:

```
lista_numeri = [3, 22, 18, 4, 38, 89]  
somma = 0
```

```
for numero in lista_numeri:  
    somma = somma + numero
```

```
media = somma / len(lista_numeri)  
print(f"La media è {media}")
```

Funzione range() - Come si Scrive

La funzione range() genera una sequenza di numeri.

Con 1 argomento (da 0 a N-1) - Si scrive così:

```
for variabile in range(numero):  
    codice
```

Esempio:

```
for num in range(10):  
    print(num)
```

Output: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Con 2 argomenti (da START a STOP-1) - Si scrive così:

```
for variabile in range(inizio, fine):  
    codice
```

Esempio:

```
for num in range(5, 10):  
    print(num)
```

Output: 5, 6, 7, 8, 9

Con 3 argomenti (START, STOP, STEP) - Si scrive così:

```
for variabile in range(inizio, fine, passo):  
    codice
```

Esempio:

```
for num in range(5, 30, 3):  
    print(num)
```

Output: 5, 8, 11, 14, 17, 20, 23, 26, 29

Equivalenza range() e while

Ciclo FOR con range	Ciclo WHILE equivalente
<pre>for num in range(10): ~~~~print(num)</pre>	<pre>count = 0 while count < 10: ~~~~print(count) ~~~~count += 1</pre>
<pre>for num in range(5, 10): ~~~~print(num)</pre>	<pre>count = 5 while count < 10: ~~~~print(count) ~~~~count += 1</pre>
<pre>for num in range(5, 30, 3): ~~~~print(num)</pre>	<pre>count = 5 while count < 30:</pre>

~~~~~print(count)
~~~~~count += 3

Table 1: Equivalenza tra FOR con range e WHILE

5. DIZIONARI

Come si Crea un Dizionario - Si scrive così:

```
nome_dizionario = {
"chiave1": valore1,
"chiave2": valore2,
"chiave3": valore3
}
```

Esempio:

```
dati_personali = {
"nome": "Marco",
"eta": 28,
"lingue_conosciute": ["Italiano", "Inglese", "Spagnolo"]
}
```

Come si Accede agli Elementi - Si scrive così:

```
nome_dizionario["chiave"]
```

Esempio:

```
print(dati_personali["nome"]) # Marco
print(dati_personali["eta"]) # 28
print(dati_personali["lingue_conosciute"]) # ['Italiano', 'Inglese', 'Spagnolo']
```

Aggiungere Elementi - Si scrive così:

```
nome_dizionario["nuova_chiave"] = valore
```

Esempio:

```
dizionario_preferiti = {}
dizionario_preferiti["colore"] = "rosso"
print(dizionario_preferiti) # {'colore': 'rosso'}
```

Modificare Elementi - Si scrive così:

```
nome_dizionario["chiave_esistente"] = nuovo_valore
```

Esempio:

```
dizionario_preferiti["colore"] = "arancione"
print(dizionario_preferiti) # {'colore': 'arancione'}
```

Scorrere Dizionari - Si scrive così:

```
for chiave in nome_dizionario:  
    print(chiave)  
    print(nome_dizionario[chiave])
```

Esempio:

```
dati_personali = {  
    "nome": "Marco",  
    "eta": 28,  
    "citta": "Verona"  
}
```

```
for chiave in dati_personali:  
    print(f"{chiave}: {dati_personali[chiave]}")
```

Output:

nome: Marco

eta: 28

citta: Verona

Operatore IN - Si scrive così:

```
if elemento in sequenza:  
    codice
```

Esempio:

```
colori_preferiti = ["rosso", "arancione", "azzurro", "verde"]  
colore_tentativo = input("Prova ad indovinare uno dei miei colori preferiti")
```

```
if colore_tentativo in colori_preferiti:  
    print(f"Hai indovinato! {colore_tentativo} è uno dei miei colori preferiti!")  
else:  
    print(f"Mi dispiace, {colore_tentativo} non è uno dei miei colori preferiti.")
```

Dizionari Annidati

```
musica_preferita = {  
    "rap": {  
        "prima_scoperta": "Eminem",  
        "preferito": "Salmo"  
    },  
    "rock": {  
        "prima_scoperta": "Jimi Hendrix",  
        "preferito": "Arctic Monkeys"  
    }  
}
```

Accesso

```
print(musica_preferita["rock"]["preferito"]) # Arctic Monkeys
```

Scorrere

```
for genere in musica_preferita:  
    print(f'GENERE: {genere}')  
for tipo_artista in musica_preferita[genere]:  
    print(f'{tipo_artista}: {musica_preferita[genere][tipo_artista]}')
```

Liste vs Dizionari

Aspetto	Liste	Dizionari
Ordine	Hanno ordine preciso (indice numerico)	Non c'è ordine significativo
Accesso	Tramite indice numerico (0, 1, 2...)	Tramite chiave personalizzata
Uso tipico	Collezioni omogenee, sequenze	Strutture complesse con attributi
Esempio	[10, 20, 30, 40]	{"nome": "Marco", "eta": 28}

Table 2: Differenze principali tra liste e dizionari

6. FUNZIONI

Come si Scrive una Funzione - Si scrive così:

```
def nome_funzione(parametri):  
    codice_della_funzione  
    return valore # opzionale
```

Esempio - Funzione Semplice Senza Return:

```
def saluta(nome):  
    print(f'Ciao {nome}')
```

Come si chiama la funzione:

```
saluta("Iacopo") # Output: Ciao Iacopo  
saluta("Giulia") # Output: Ciao Giulia  
saluta("Sofia") # Output: Ciao Sofia
```

IMPORTANTE: La variabile `nome` esiste SOLO all'interno della funzione.

Funzioni con Return - Si scrive così:

```
def nome_funzione(parametri):  
    # calcoli  
    return risultato
```

Esempio:

```
def calcola_media(neri):  
    somma = 0  
    for numero in numeri:  
        somma += numero  
    media = somma / len(numeri)  
    return media
```

Come si usa:

```
lista_numeri = [6, 4, 9, 10, 6, 2]  
media_voti = calcola_media(lista_numeri)  
print(f"La media è {media_voti}") # La media è 6.166666666666667
```

Funzioni con Più Parametri - Si scrive così:

```
def nome_funzione(param1, param2, param3):  
    codice  
    return risultato
```

Esempio:

```
def media_tra_due_numeri(num1, num2):  
    somma = num1 + num2  
    media = somma / 2  
    return media  
  
media_calcolata = media_tra_due_numeri(17, 11)  
print(media_calcolata) # 14.0
```

Funzioni che Chiamano Altre Funzioni

```
def calcola_somma(neri):  
    somma = 0  
    for numero in numeri:  
        somma += numero  
    return somma  
  
def calcola_media(neri):  
    somma = calcola_somma(neri) # Riutilizzo funzione precedente  
    media = somma / len(numeri)  
    return media  
  
lista_numeri = [6, 4, 9, 10, 6, 2, 8, 5]  
somma_voti = calcola_somma(lista_numeri)
```

```
media_voti = calcola_media(lista_numeri)
print(somma_voti) # 50
print(media_voti) # 6.25
```

Parametri con Valori Default - Si scrive così:

```
def nome_funzione(parametro_obbligatorio, parametro_opzionale="valore_default"):
    codice
```

Esempio:

```
def saluta(nome, cognome="Rossi"):
    print(f"Ciao {nome} {cognome}")
```

Come si usa:

```
saluta("Francesco") # Ciao Francesco Rossi
saluta("Pieralberto", "Antoniazzi") # Ciao Pieralberto Antoniazzi
```

REGOLA: I parametri con default devono essere DOPO quelli senza default.

Esempio Avanzato: Replica di range()

```
def custom_range(param1=None, param2=None, param3=None):
    if param2 is None: # 1 parametro
        num_max = param1
        num_start = 0
        operatore = 1
    elif param3 is None: # 2 parametri
        num_max = param2
        num_start = param1
        operatore = 1
    else: # 3 parametri
        num_max = param2
        num_start = param1
        operatore = param3
```

```
lista_numeri = []
count = num_start

if operatore > 0:
    while count < num_max:
        lista_numeri.append(count)
        count += operatore
else:
    while count > num_max:
        lista_numeri.append(count)
        count += operatore

return lista_numeri
```

Utilizzo

```
for elemento in custom_range(10):  
    print(elemento) # 0 a 9
```

```
for elemento in custom_range(3, 10):  
    print(elemento) # 3 a 9
```

```
for elemento in custom_range(4, 20, 4):  
    print(elemento) # 4, 8, 12, 16
```

7. CLASSI (OOP)

Cos'è una Classe

Una classe è un "modello" per creare oggetti con caratteristiche (attributi) e comportamenti (metodi) comuni.

Come si Scrive una Classe - Si scrive così:

```
class NomeClasse:  
    def init(self, parametri):  
        self.attributo = valore
```

```
def metodo(self):  
    codice
```

Spiegazione parti:

- `class NomeClasse:` - Definisce la classe
- `__init__` - Costruttore, eseguito quando crei un oggetto
- `self` - Riferimento all'oggetto stesso
- `self.attributo` - Variabile dell'oggetto
- `def metodo(self):` - Funzione della classe

Esempio Base: Classe Persona

```
class Persona:  
    def init(self, nome):  
        self.nome = nome
```

Come si crea un oggetto dalla classe - Si scrive così:

```
oggetto = NomeClasse(parametri)
```

Esempio:

```
persona1 = Persona("Iacopo")  
persona2 = Persona("Giulia")
```

Come si accede agli attributi - Si scrive così:

```
oggetto.attributo
```

Esempio:

```
print(persona1.nome) # Iacopo  
print(persona2.nome) # Giulia
```

Classe con Più Attributi - Si scrive così:

```
class NomeClasse:  
    def init(self, param1, param2, param3):  
        self.attributo1 = param1  
        self.attributo2 = param2  
        self.attributo3 = param3
```

Esempio:

```
class Persona:  
    def init(self, nome, eta, artisti_preferiti):  
        self.nome = nome  
        self.eta = eta  
        self.artisti_preferiti = artisti_preferiti  
  
persona1 = Persona("Iacopo", 25, ["Miles Davis", "Jimi Hendrix"])  
persona2 = Persona("Giulia", 24, ["The Lumineers", "Bon Iver"])  
  
print(persona1.nome) # Iacopo  
print(persona1.eta) # 25  
print(persona1.artisti_preferiti) # ['Miles Davis', 'Jimi Hendrix']
```

Metodi - Come si Scrivono - Si scrive così:

```
class NomeClasse:  
    def init(self, parametri):  
        self.attributo = valore  
  
    def nome_metodo(self):  
        codice  
        # può usare self.attributo
```

Esempio:

```
class Persona:  
    def init(self, nome, eta):  
        self.nome = nome  
        self.eta = eta
```

```
def presentazione(self):  
    print(f"Ciao, mi chiamo {self.nome}")
```

Come si chiama un metodo - Si scrive così:

```
oggetto.metodo()
```

Esempio:

```
persona1 = Persona("Iacopo", 25)  
persona1.presentazione() # Output: Ciao, mi chiamo Iacopo
```

IMPORTANTE: self rappresenta l'istanza stessa della classe.

Esempio Completo: Gioco di Ruolo

```
class Personaggio:  
    def init(self, nome, classe, vita, attacco, num_pozioni=0):  
        self.nome = nome  
        self.classe = classe  
        self.vita = vita  
        self.vita_max = vita  
        self.attacco = attacco  
        self.num_pozioni = num_pozioni
```

```
def attacca(self, nemico):  
    nemico.vita -= self.attacco  
    print(f"{self.nome} ha colpito {nemico.nome}!")  
    print(f"{nemico.nome} ora ha {nemico.vita} di vita.")
```

```
def stampa_scheda(self):  
    print("\n=== SCHEDA PERSONAGGIO ===")  
    print(f"Nome: {self.nome}")  
    print(f"Classe: {self.classe}")  
    print(f"Attacco: {self.attacco}")  
    print(f"Vita: {self.vita}/{self.vita_max}")  
    print(f"Pozioni: {self.num_pozioni}")  
    print("=====\n")
```

Creare personaggi

```
pg1 = Personaggio("Ercole", "Guerriero", 100, 20, 2)
pg2 = Personaggio("Merlino", "Mago", 80, 30, 3)
```

Usare i metodi

```
pg1.stampa_scheda()
pg2.stampa_scheda()
```

```
pg1.attacca(pg2)
pg1.attacca(pg2)
```

```
pg1.stampa_scheda()
pg2.stampa_scheda()
```

Concetti Chiave

- **Classe:** Modello/template per creare oggetti
 - **Istanza:** Oggetto specifico creato da una classe
 - **Attributo:** Variabile che appartiene all'istanza (`self.nome`)
 - **Metodo:** Funzione che appartiene alla classe
 - **`__init__`:** Costruttore, eseguito quando si crea un'istanza
 - **`self`:** Riferimento all'istanza corrente
-

8. CHEAT SHEET RAPIDO

Operatori di Confronto

Operatore	Significato
<code>==</code>	Uguale
<code>!=</code>	Diverso
<code>></code>	Maggiore
<code><</code>	Minore
<code>>=</code>	Maggiore o uguale
<code><=</code>	Minore o uguale

Operatori Logici

Operatore	Descrizione
and	Entrambe le condizioni True
or	Almeno una condizione True
not	Inverte valore booleano

Conversioni Tipo

```
int("42") # Stringa → Intero
str(42) # Intero → Stringa
float("3.14") # Stringa → Float
```

Input e Output

```
nome = input("Come ti chiami? ") # Input stringa
eta = int(input("Età? ")) # Input convertito
print(f"Ciao {nome}, hai {eta} anni") # f-string
```

Incremento Abbreviato

```
x = x + 1 # equivale a → x += 1
x = x - 1 # equivale a → x -= 1
x = x * 2 # equivale a → x *= 2
x = x / 2 # equivale a → x /= 2
```

9. SCHEMA DECISIONALE

Quando Usare Liste vs Dizionari?

USA LISTE quando:

- Hai collezioni omogenee di dati
- L'ordine è importante
- Accedi tramite posizione numerica
- Esempio: `voti = [7, 8, 6, 9]`

USA DIZIONARI quando:

- Hai dati con attributi specifici
- Accedi tramite nome/chiave
- Rappresenti entità complesse
- Esempio: `studente = {"nome": "Marco", "eta": 20}`

Quando Usare While vs For?

USA WHILE quando:

- Non sai quante iterazioni servono
- Dipende da una condizione dinamica
- Esempio: indovina il numero, menu interattivo

USA FOR quando:

- Devi scorrere una sequenza (lista, stringa, range)
- Sai esattamente quante iterazioni servono
- Esempio: calcolare media, stampare elementi

Quando Creare una Funzione?

CREA FUNZIONE quando:

- Ripeti lo stesso codice più volte
- Vuoi organizzare meglio il codice
- Devi testare una porzione isolata
- Vuoi riutilizzare logica in progetti futuri

Quando Creare una Classe?

CREA CLASSE quando:

- Rappresenti entità con attributi E comportamenti
 - Devi creare più oggetti simili
 - Vuoi raggruppare dati e funzioni correlate
 - Esempio: Personaggio, Auto, Studente
-