**LA SELEZIONE: IL COSTRUTTO IF**

# 

[INTRODUZIONE](#_4r3c29uxi1t3)

[COSA FA IL CODICE? [5 minuti]](#_ljxlilwf1p3p)

[DOMANDE [5 minuti]](#_vpog5ejff5e)

[COSA FA IL CODICE? [5 minuti]](#_hzbm7t2eszry)

[DOMANDE [5 minuti]](#_k46lfpai5fqm)

[COSA FA IL CODICE? [5 minuti]](#_va38dtroul8g)

[DOMANDE [5 minuti]](#_7rirgqz1f1w)

[MODIFICA IL CODICE [10 minuti]](#_nurp4hrzp16r)

[TROVA GLI ERRORI [5 minuti]](#_qarslsy342q)

[METTI IN ORDINE IL CODICE [5 minuti]](#_1vi2gi14r53y)

[TERMINI CHIAVE](#_dwx8dylvwd77)

[DA RICORDARE IN PYTHON](#_9mzdwubdvr5f)

[ESERCIZI](#_9l3oq24a5etb)

# 

# 

# INTRODUZIONE

In Python il costrutto ifè uno degli elementi fondamentali della programmazione che consente di prendere decisioni nel codice. Esattamente come il nome suggerisce, con l’istruzione if, è possibile creare rami condizionali all'interno del programma, consentendo al programma di eseguire determinate azioni solo se una condizione specificata è soddisfatta. Questo costrutto è essenziale per la logica di controllo di flusso e per scrivere programmi che rispondano in modo dinamico alle variazioni delle situazioni.

Nel blocco di codice all'interno dell'istruzione *if* è possibile inserire una serie di istruzioni che verranno eseguite solo se la condizione specificata è valutata come *True (vera)*. Se la condizione è *False (Falsa)*, il blocco di codice all'interno dell'istruzione *if* viene ignorato, e il programma passa all'istruzione successiva.

L’istruzione *if* può essere combinata con le istruzioni *elif* e *else* per gestire più casi e situazioni complesse.

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

print("Benvenuto nel mio programma di conversazione")

print("Ti piace andare in bicicletta?")

risposta=input("Rispondi sì o no: ")

if risposta=="sì":

print("Molto bene! Ti terrai in forma.")

print("Ciao ciao.")

|  |
| --- |
| Secondo voi, che cosa appare a video eseguendo il codice qui sopra? Descrivete a parole **SENZA eseguire il codice.** |
| Chiede se ti piace andare in bicicletta, se la riposta è si stampa la risposta “molto bene! Ti terrai in forma” se la risposta è no stampa solo “Ciao ciao.” |
| Ora copiate e incollate il codice in VSCode (salvate con nome *22\_Selezione\_1via.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione del perché il programma si comporta in questo modo. |
| Il codice ha eseguito come mi aspettavo |

# 

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande:

1. A cosa serve l’istruzione *if ?*

*Serve ad eseguire un blocco di codice solo se la condizione è vera*

1. Hai notato che la riga sotto all’istruzione *if* è indentata? Perché secondo te?

(NB: indentata significa che è spostata a destra di una tabulazione)

Serve a inserire le condizioni dell’if

1. Cosa succede se si digita "Sì" invece di "sì" quando si esegue il programma? (provate e verificate)

Il codice avendo solo una condizione vera stamperà “ciao ciao.”

1. Qual è la differenza tra = e == ? In questo programma vengono utilizzati entrambi?

Viene usati tutti e due nel codice, “=” serve per assegnare un valore ad una variabile come “Risposta = input” o “x = 10”, mentre i due uguali: “==” servono a verificare se i valori sono uguali, per esempio “risposta == “si” serve per verificare se la risposta è si

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

print("Benvenuto nel mio programma di conversazione")

print("Ti piace andare in bicicletta?")

risposta=input("Rispondi sì o no: ")

if risposta=="sì":

print("Molto bene! Ti terrai in forma.")

else:

print("Forse ti piace qualche altro sport.")

print("Ciao ciao.")

|  |
| --- |
| Secondo voi, che cosa appare a video eseguendo il codice qui sopra? Descrivete a parole **SENZA eseguire il codice.** |
| Esegue lo stesso codice precedente aggiugendo un “else” quindi se la risposta non è “Si” verrà stampato il blocco nell’else + “Ciao ciao” |
| Ora copiate e incollate il codice in VSCode (salvate con nome *23\_Selezione\_2vie.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione del perché il programma si comporta in questo modo. |
| Il codice stampa una domanda, verifica se la risposta è si o no, se è si stampa “Molto bene! Ti terrai in forma.” se la risposta è tutt’altro che si, il codice stamperà “Forse ti piace qualche altro sport… Ciao ciao” |

# 

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande:

1. A cosa serve l’istruzione *else*?

Serve a eseguire un blocco di codice se la condizione non è vera

1. Hai notato che la riga sotto all’istruzione *else* è indentata? Perché secondo te?

Stesso concetto dell’if per essere eseguito il blocco di codice deve essere indentato sotto all’else

# 

# 

# COSA FA IL CODICE? [5 minuti]

**Lavoro in coppia**

print("Benvenuto nel mio programma di conversazione")

print("Ti piace andare in bicicletta?")

risposta=input("Rispondi sì o no: ")

if risposta=="sì":

print("Molto bene! Ti terrai in forma.")

elif risposta=="no":

print("Forse ti piace qualche altro sport.")

else:

print("Risposta errata, dovevi rispondere sì o no!")

print("Ciao ciao.")

|  |
| --- |
| Secondo voi, che cosa appare a video eseguendo il codice qui sopra? Descrivete a parole **SENZA eseguire il codice.** |
|  |
| Ora copiate e incollate il codice in VSCode (salvate con nome *24\_Selezione\_3vie.py*) e controllate cosa fa. Fa quello che vi aspettavate? Provate a dare una spiegazione del perché il programma si comporta in questo modo. |
|  |

# 

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande:

1. A cosa serve l’istruzione *elif?* Di quali parole inglesi è abbreviazione?
2. Quante condizioni puoi controllare con una *if* semplice (senza *else*)? E quanti percorsi diversi puoi impostare?
3. Quante condizioni puoi controllare con una if-else? E quanti percorsi diversi puoi impostare?
4. Se vuoi controllare 4 casistiche (esempio: se a corrisponde ad uno dei valori 1,2,3,4), di quante condizioni (e quindi quanti *if* e quanti *elif* o *else*) avrai bisogno?

# 

# 

# MODIFICA IL CODICE [10 minuti]

**Lavoro in autonomia**

Crea un nuovo file *25\_Voti.py*, copia il seguente codice e modificalo in modo tale che stampi *“Purtroppo il tuo voto è insufficiente”* nel caso in cui il voto sia minore di 6 e stampi *“Voto sufficiente”* nel caso in cui il voto inserito sia uguale a 6.

print("Come è andata la verifica?")

voto = float(input("Inserisci un voto da 0 a 10: "))

if voto > 6:

print("Bene. Hai preso più della sufficienza!")

# TROVA GLI ERRORI [5 minuti]

**Lavoro in coppia**

|  |
| --- |
| Questo programma contiene **4 errori**. Segnateli con un colore diverso. |
| print("Ciao! Controlliamo come va in informatica")  voto1 = float(input("inserisci il 1° voto che hai preso"))  voto2 = input("inserisci il 2° voto che hai preso")  if voto1=voto2:  print("Il tuo rendimento è stabile.")  elif voto2>voto1  print("Stai migliorando")  else:  print("Purtroppo stai peggiorando.") |
|  |

# 

# 

# METTI IN ORDINE IL CODICE [5 minuti]

**Lavoro in autonomia**

1. Clicca sul seguente [link](https://mariaserafini.github.io/daVinci_ParsonsPuzzles/parsons/06bis_selezione3vie_semplice.html)
2. Metti in ordine i pezzetti di codice che trovi nella pagina e clicca sul pulsante “Get Feedback” per verificare se hai capito quanto fatto finora.
3. Copia e incolla il codice nell’ordine corretto nel box qui sotto

|  |
| --- |
| Inserisci qui il codice nell’ordine corretto |
|  |

# TERMINI CHIAVE

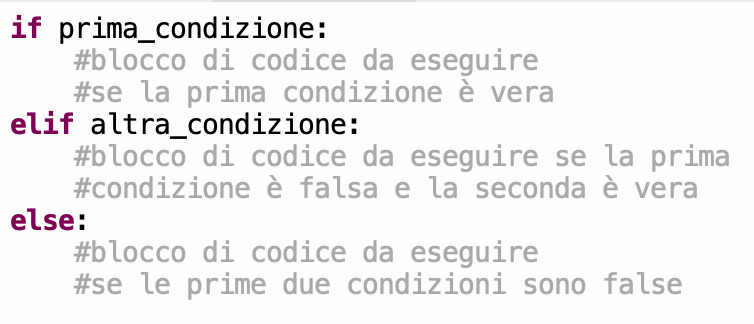
|  |  |  |
| --- | --- | --- |
| **operatore relazionale** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Un simbolo che serve per fare un confronto tra due dati. |  | >, >=, <, <=, ==, != sono gli operatori relazionali di Python |
| **condizione** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Il test che inseriamo nel costrutto *if*. Si costruisce inserendo un *operatore relazionale* tra due dati.  Il risultato della condizione deve essere vero (**True**) o falso (**False** |  | a > 5  b == a  risposta == “sì” |
| **selezione** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Quando nel codice c’è un punto in cui viene effettuata una scelta e viene utilizzato il costrutto *if* per creare percorsi alternativi |  |  |
| **selezione a due vie** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Il codice prevede due percorsi alternativi, uno quando la condizione è vera (ramo *if*) , l’altro quando è falsa (ramo *else*) |  |  |
| **selezione a tre vie** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| ll test prevede 3 percorsi alternativi, uno quando la prima condizione è vera (ramo *if*), uno quando la seconda condizione è vera (ramo elif) e l’ultimo quando le prime due condizioni sono false (ramo *else*). | In scratch in ramo *elif* non è previsto. |  |
| **indentare** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Scrivere una o più istruzioni “rientrate” (di una tabulazione o 4 spazi) rispetto alle istruzioni precedenti/successive |  | L’istruzione print è **indentata** rispetto a if |

# 

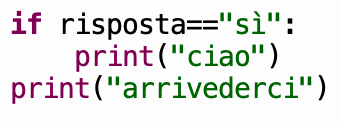
Puoi trovare una spiegazione sull’uso della selezione anche alle pagine 171 - 173 del libro di testo.

# DA RICORDARE IN PYTHON

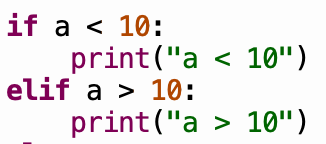
* La sintassi del costrutto di selezione è la seguente:



* L’istruzione **if** esegue un blocco di istruzioni (ramo **if**) se la condizione è vera, **altrimenti** **se** è vera la seconda condizione (ramo **elif**) esegue il blocco di istruzioni al suo interno, **altrimenti** esegue il blocco di istruzioni dell’ultimo ramo (ramo **else**).
* I **due punti :** sono usati per annunciare che sta arrivando un nuovo blocco di codice e tutte le righe del blocco **DEVONO** essere indentate della stessa quantità di spazi. Alla fine del blocco puoi interrompere il rientro.

Esempio:

In questo caso la scritta “ciao” viene visualizzata soltanto nel caso in cui la risposta è "sì", mentre la scritta “arrivederci” viene visualizzata indipendentemente dal valore contenuto nella variabile risposta.

* L**’indentazione** in Python è fondamentale per organizzare i blocchi di istruzioni
* L**'allineamento** dei rami *elif* e *else* deve essere lo stesso del **loro** *if***.** Infatti bisogna ricordare che i rami *elif* e *else* possono essere utilizzati solo se preceduti da un ramo *if*.
* È possibile utilizzare il costrutto if-elif senza il caso else, ma non è possibile utilizzare elif senza if. Il codice qui sotto (ad esempio) è corretto da un punto di vista sintattico.

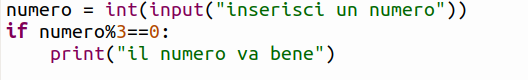
# ESERCIZI

**Lavoro in autonomia**

Risolvi i seguenti esercizi. Quando è richiesto di scrivere del codice, il nome del file in cui deve essere salvato è indicato in parentesi.

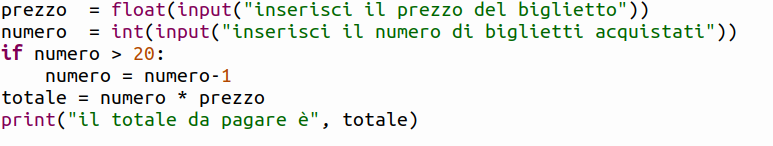
**Dove non specificato il valore di un dato, questo va chiesto in input all’utente.**

1. Cosa stampa il seguente codice?



[*25\_Divisibile.py*] Ricopia il codice per accertarti di aver risposto correttamente e modificalo in modo tale che venga controllato se un numero è divisibile per 5.

1. [*26\_Multiplo.py*] Modificail codice dell’esercizio precedente per verificare se dati due numeri (interi) inseriti dall’utente, il primo è multiplo del secondo.
2. Cosa stampa il seguente codice?



*[27\_Biglietteria.py]* Ricopia il codice per accertarti di aver risposto correttamente e modificalo in modo tale che venga regalato un biglietto omaggio **ogni** 20 biglietti (se si acquistano 30 biglietti se ne pagano 29, acquistandone 50 se ne pagano 48, ecc).

1. *[28\_Fruttivendolo.py]* Calcolare il totale speso per acquistare delle pesche sapendo che se si superano i 10 euro si ha il 20% di sconto.
2. *[29\_Bolletta.py]* Calcolare il costo della bolletta di consumo del gas (espresso in metri cubi) ottenuto dalla somma delle seguenti voci:
   1. quota fissa pari a 20€
   2. 0,575€ al metro cubo per i primi 500 metri cubi
   3. 0,783€ al metro cubo per ogni metro cubo eccedente i primi 500
3. *[30\_PariDispari.py]* Verifica se un numero è pari o dispari. Stampa “pari” in caso affermativo, “dispari” altrimenti
4. *[31\_Sottrazione.py]* Scrivi un programma che esegua la sottrazione di due numeri, sottraendo il numero minore al maggiore.
5. *[32\_DifferenzaSommaRandom.py]*

Scrivi un programma che generi due numeri casuali interi fra 0 e 10 e calcoli la differenza fra i due numeri se il loro prodotto è maggiore di 20 oppure la loro somma se il prodotto è minore o uguale a 20.

1. *[33\_Rifornimento.py]*

Un automobilista fa rifornimento di benzina in una stazione di servizio che applica uno sconto del 5% se la spesa è superiore ai 60€ . Ipotizzando che un litro di benzina costi 1.76€, calcolare il totale che l’automobilista deve pagare in base ai litri inseriti.

1. [*34\_NomeGenerazione.py*] Chiedere all’utente l’anno di nascita e mostrare a video il nome della sua generazione: se è nato prima del 1946, mostrare “Prima dei Boomer”; se è nato tra il 1946 e il 1964, mostrare “Boomer”; se è nato tra il 1965 e il 1980, mostrare “Generazione X”; se è nato tra il 1981 e il 2000, mostrare “Generazione Y (Millennials)". In ultimo, se è nato dopo il 2000, indicare la “Generazione Z”.
2. [*35\_Multa.py*] Dato l’articolo 142 del Codice della strada, scrivere un programma che in automatico calcoli la multa da pagare.

Articolo 142 del Codice della strada:  
Chiunque non osserva i limiti minimi di velocità ovvero supera i limiti massimi di velocità di non oltre 10 km/h, è soggetto alla sanzione amministrativa del pagamento di una somma da euro 36.   
Chiunque supera di oltre 10 km/h e di non oltre 40 km/h i limiti massimi di velocità è soggetto alla sanzione amministrativa del pagamento di una somma di euro 148.  
Chiunque supera di oltre 40 km/h ma non di oltre 60 km/h i limiti massimi di velocità è soggetto alla sanzione amministrativa del pagamento di una somma di euro 370.  
Chiunque supera di oltre 60 km/h i limiti massimi di velocità è soggetto alla sanzione amministrativa del pagamento di una somma da euro 500.