**ALGEBRA BOOLEANA**

[INTRODUZIONE](#_heading=h.2et92p0)

[RIPASSO [5 min]](#_heading=h.3dy6vkm)

[COSA FA IL CODICE? [10 minuti]](#_heading=h.2s8eyo1)

[DOMANDE [5 minuti]](#_heading=h.lnxbz9)

[COSA FA IL CODICE? [10 minuti]](#_heading=h.1ksv4uv)

[DOMANDE [10 minuti]](#_heading=h.z337ya)

[TABELLE DI VERITA’](#_heading=h.3j2qqm3)

[MODIFICA IL CODICE? [5 minuti]](#_heading=h.1ci93xb)

[TERMINI CHIAVE](#_heading=h.qsh70q)

[ESERCIZI](#_heading=h.1pxezwc)

# 

# 

# INTRODUZIONE

In questa lezione affrontiamo un argomento matematico che si connette all’informatica chiamato **algebra booleana**. In particolare, esso è un ramo dell’algebra che si definisce sui valori di verità **VERO** o **FALSO** (spesso indicati con 1 e 0 rispettivamente) e sulle operazioni logiche principali **AND**, **OR** e **NOT**.

Si definisce anche un nuovo tipo di dato, chiamato *bool*.

Per la prima parte della lezione utilizziamo dei comandi nella *shell*, invece di eseguire il codice scritto in un file.

Shell:



# 

# RIPASSO [5 min]

**Lavoro in autonomia**

Completa la seguente tabella:

| **TIPO OPERATORE** | **ELENCO DEGLI OPERATORI** | **TIPO DATO INPUT** | **TIPO DATO OUTPUT** |
| --- | --- | --- | --- |
| Aritmetici | +, -, \*, \*\*, /, //, % | int, float, str | int, float, str |
| Relazionali | >, >=, <, <=, =, == | str, int, float | bool |

Una **condizione** deve sempre contenere due \_\_\_\_\_\_\_\_ e un operatore \_\_\_\_\_\_\_\_\_

# 

# COSA FA IL CODICE? [10 minuti]

**Lavoro in coppia**

Immaginate le seguenti istruzioni scritte sul terminale di VSCode







| Secondo voi, cosa apparirà come risultato?? |
| --- |
| Vero, falso, vero, falso, falso, vero |
| Scrivete le istruzioni sul prompt e controllate. Fa quello che vi aspettavate? Se non lo fa, descrivete cosa accade in modo diverso. |
| Si ha fatto quello che mi aspettavo |

# 

# 

# DOMANDE [5 minuti]

**Lavoro in coppia**

Rispondete alle seguenti domande.

1. Che cosa fanno nello specifico i comandi a = 3 < 7 e b = 5 == 7 ?

Assegna alle variabili una condizione che può essere vera o falsa

1. Per rispondere alla domanda precedente scrivete **una alla volta in ordine** le seguenti istruzioni e prendete nota dei risultati nella tabella sottostante:

| **ISTRUZIONE** | **RISULTATO** |
| --- | --- |
| a = 3 < 7 | True |
| b = 5 != 7 | True |
| a | True |
| b | True |
| type(a) | Class Bool |
| c = 7 | None |
| d = "ciao" | None |
| type(c) | Class int |
| type(d) | Class str |
| not a | False |

1. Che cosa fa la funzione *type*?

Stampa la classe della variabile (str, int, bool…)

1. Che cosa fa l’operatore *not*?

Inverte il valore bool, quindi se è vero diventa falso, se è falso diventa vero

# COSA FA IL CODICE? [10 minuti]

**Lavoro in coppia**

Eseguite le seguenti istruzioni python su VSCode e completate la seguente tabella con le risposte che ottieni. Prima di rispondere, provate ad immaginare quale può essere il risultato ;-)

| **ISTRUZIONE** | **RISPOSTA** |
| --- | --- |
| True and True | True |
| False and True | False |
| True and False | False |
| False and False | False |
| True or True | True |
| True or False | True |
| False or True | True |
| False or False | False |
| True or False and False | True |
| not True and False | False |

Quindi, potete completare le seguenti affermazioni:

* l’operatore booleano **and** restituisce **True** soltanto nel caso in cui \_\_\_\_\_\_\_\_\_\_\_\_  
  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* l’operatore booleano **or** restituisce **False** soltanto nel caso in cui \_\_\_\_\_\_\_\_\_\_\_\_\_  
  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 

# DOMANDE [10 minuti]

Possiamo aggiungere alla tabella degli operatori le seguenti informazioni (completa la tabella):

| **TIPO OPERATORE** | **ELENCO DEGLI OPERATORI** | **TIPO DATO INPUT** | **TIPO DATO OUTPUT** |
| --- | --- | --- | --- |
| Booleani (o logici) | and, or, not | bool | bool |

* Avete notato che l’operatore *not* è un operatore diverso dagli altri due (*and* e *or*)? In cosa si differenzia?
* Se non inseriamo alcuna parentesi, in che ordine vengono eseguite le operazioni booleane? Metti in ordine di precedenza i tre operatori.
* Secondo voi, in quale punto del codice saranno utili gli operatori booleani? In quale contesto?

# TABELLE DI VERITA’

Una tabella di verità è una tabella che mostra tutti i possibili risultati di tutte le possibili combinazioni di input. Vediamo di seguito le tabelle di verità degli operatori **and**, **or** e **not.** Dati due valori A e B le prime due tabelle rappresentano rispettivamente il risultato delle operazioni logiche **A and B** e **A or B**. La terza tabella rappresenta invece, dato A, il risultato dell’operazione logica **not A**.

| A | B | **A and B** |
| --- | --- | --- |
| T | T | **T** |
| T | F | **F** |
| F | T | **F** |
| F | F | **F** |

| A | B | **A or B** |
| --- | --- | --- |
| T | T | **T** |
| T | F | **T** |
| F | T | **T** |
| F | F | **F** |

| A | **not A** |
| --- | --- |
| T | **F** |
| F | **T** |

# 

# 

# 

# MODIFICA IL CODICE? [5 minuti]

**Lavoro in autonomia**

1. È dato il seguente codice, inserite al posto della condizione inserita nell’if due condizioni legate da un operatore booleano. Potete copiare e incollare il codice da qui sotto, salvate il file con nome *48-OpBool.py*

voto = float(input("inserire voto "))

if 3 <= voto <= 10:

print("il voto va bene")

else:

print("il voto non va bene")

1. Modificate il seguente codice sostituendo le if annidate con condizioni multiple (legate da and e or). Potete copiare e incollare il codice da qui sotto, salvate il file come *49-TriangoloEquilatero.py*

lato1 = float(input("inserire primo lato "))

lato2 = float(input("inserire secondo lato "))

lato3 = float(input("inserire terzo lato "))

if lato1 == lato2:

if lato2 == lato3:

print("il triangolo è equilatero")

else:

print("il triangolo non è equilatero")

else:

print("il triangolo non è equilatero")

Modificate il codice dell’esercizio precedente inserendo i controlli opportuni anche per verificare che se il triangolo è isoscele (o scaleno). Salvate il file come *50-VerificaTriangolo.py*

1. Dato il codice seguente, v erificate in VSCode per quali valori di a e b appare la scritta “ok”. Poi, riscrivete il codice con condizioni opportune in modo da eliminare il ramo *else,* ovvero utilizzando solo il ramo *if* per visualizzare la scritta. Salvate il file con nome *51-Morgan.py*

a = int(input("inserisci un numero "))

b = int(input("inserisci un altro numero "))

if a > 10 and b < 5:

print()

else:

print("ok")

# TERMINI CHIAVE

| **Bool** | | |
| --- | --- | --- |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Tipo di dato di python che rappresenta i valori True (vero) e False (falso). |  | a = True |
| **Operatore unario** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Un operatore che prende un solo valore come input. (es. operatore *not*, l’operatore - in alcuni casi) |  | not a  -5 |
| **Operatore binario booleano (o logico)** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Un operatore che posto tra due valori booleani (tipicamente due condizioni) restituisce un valore booleano. Gli operatori binari booleani sono *and* e *or* |  | a > 7 and b < 4  a == 2 or b < 10 |
| **and** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| operatore booleano che restituisce **vero** solo nel caso in cui **entrambe** le condizioni siano **vere**, **falso** nel caso in cui **almeno una** condizione è **falsa** |  | a > 7 and b < 4 |
| **or** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| operatore booleano che restituisce **vero** solo nel caso in cui **entrambe** le condizioni siano **vere**, **falso** nel caso in cui **almeno una** condizione è **falsa** |  | a == 2 or b < 10 |
| **type()** | | |
| **SIGNIFICATO** | **SCRATCH** | **PYTHON** |
| Funzione di python che restituisce il tipo di dato di una variabile |  | a = 5  type(a) → restituisce Int |

Puoi trovare una spiegazione sull’uso della selezione anche alle pagine 172 - 173 del libro di testo.

DA RICORDARE IN PYTHON

* Quando abbiamo bisogno di “invertire” una condizione multipla, possiamo ricordare le **leggi di de Morgan**:
* not (a **and** b) = not a **or** not b
* not (a **or** b) = not a **and** not b

Quindi, ad esempio se vogliamo negare la condizione:

x > 3 and risposta ==”sì”

possiamo scrivere

x <= 3 or risposta != “sì”

* Gli operatori booleani vengono svolti in un ordine preciso, ovvero prima l’operatore not, poi l’operatore and e infine l’operatore or. Per evitare che accadano fatti imprevisti, è sempre meglio mettere le parentesi per indicare l’ordine in cui devono essere svolte le operazioni.

# 

# 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. [52\_MisureTriangolo.py]

Stabilire se tre misure inserite in input possono costituire i lati di un triangolo. *Ricorda che in un triangolo ogni lato deve essere minore della somma degli altri due.*

1. [*53\_Mesi.py*]  
   Chiedere all’utente il numero corrispondente al mese dell’anno e rispondere con l’opportuno numero di giorni che ha il mese richiesto. Ad esempio, inserendo 3 (marzo) si otterrà come risposta 31 (numero dei giorni del mese di marzo).
2. [*54\_Bisestile.py*]  
   Stabilire se un anno (inserito in input) è bisestile. Ricorda che un anno è bisestile se è divisibile per 4 ma non per 100, tranne gli anni divisibili per 400 che lo sono. Per esempio, l’anno 2000 è stato bisestile, mentre il 2100 non lo sarà.
3. [*55\_Vocali.py*]

Dato un carattere inserito da tastiera (in minuscolo), stabilire se esso è una vocale oppure no.

1. [*56\_Quadrante.py*]

Date le coordinate x e y di un punto nel piano, stabilire se esso appartiene al primo o al terzo quadrante (non serve individuare quale dei due), oppure ad uno degli altri due (sempre senza individuare quale). Inoltre, stabilire se esso si trova sulla bisettrice dei quadranti considerati oppure no. *Nota: per semplicità considerare l’origine degli assi come parte del primo/terzo quadrante.*