

# Logical Operators: and, or and not

---

## And, Or, Not

- Have the user enter three integers between 1 and 10. Display one message if all three are even, and another message if they are not.
- Have the user enter three integers between 1 and 10. Display one message if one or more of them is even, and another message if this is not the case.
- Have the user enter three integers between 1 and 10. Display one message if *exactly one* of them is even, and another message if not.
- Rewrite the following piece of code, so that it uses logical operators instead of nested ifs.
 

```
num = int(input("Please enter a positive integer: "))
if num > 9:
    if num < 100:
        print("Your number is two digits long.")
    else:
        print("Your number is not two-digits long.")
else:
    print("Your number is not two-digits long.")
```
- Display a message indicating if two user-generated integers are both positive, both negative, or one positive and one negative.
- Using only one if statement, and one else statement, display a message is the user enters a vowel ('A', 'E', 'I', 'O' or 'U'), and another message if they do not.
- Some programming languages provide an *exclusive or* operator, often abbreviated *xor*. Unlike the *or* operator, *xor* is only True when *exactly one* of two boolean values is True. Write a program that asks the user to enter 'T' or 'F' twice, then outputs True or False as determined by *xor*.
- De Morgan's Laws*: Using two boolean values a and b, complete the tables below.

a	b	not (a and b)
True	True	
True	False	
False	True	
False	False	

a	b	not (a) or not(b)
True	True	
True	False	
False	True	
False	False	

What can you conclude about the expressions **not (a and b)** and **not (a) or not (b)**?

- De Morgan's Laws, Part 2*: Make a hypothesis about the expression **not (a or b)**. Test your prediction using tables similar to those in the previous question.