



# Week 4 materials

## topics covered

- boolean expressions
- comparison operators
- boolean operators: and, or, not
- conditional statement - if, if-else

## some recap

- in Python, the two boolean values are `True` and `False` (the capitalization must be exactly as shown), and the Python type is `bool`

```
# Checking the types of booleans
a = True
b = 'True'

print('The type of a is:', type(a))
print('The type of b is:', type(b))

# Returns
# The type of a is: <class 'bool'>
# The type of b is: <class 'str'>
```

- A boolean expression is an expression that returns either True or False

For example, the operator `==` tests if two values are equal

```
5 == (3 + 2) # Is 5 equal to the result of 3 + 2?
```

returns: `True`

```
5 == 6
```

returns: `False`

A non-zero integer is always `True`

An integer with a value of `0` is `False`

```
a = 1
print(bool(a))

#returns True
```

six common **comparison operators** which all produce a bool result

Operation	Description	Example	Returns
<code>x == y</code>	x is equal to y	<code>3 == 2</code>	False
<code>x != y</code>	x is not equal to y	<code>3 != 2</code>	True
<code>x &lt; y</code>	x is less than y	<code>3 &lt; 2</code>	False
<code>x &gt; y</code>	x is greater than y	<code>3 &gt; 2</code>	True
<code>x &lt;= y</code>	x is less than or equal to y	<code>3 &lt;= 2</code>	False
<code>x &gt;= y</code>	x is greater than or equal to y	<code>3 &gt;= 2</code>	True

- A common **error** is to use a single equal sign (`=`) instead of a double equal sign (`==`)  
Remember that `=` is an assignment operator and `==` is a comparison operator
- There is no such thing as `=<` or `=>`

some examples of `==` operator:

```
a = 4
b = 4
print(a == b)

#returns True
```

```
a = 4
print(a == 5)

#returns False
```

Strings must be identical in case and length to be considered equal in Python

```
a = "Code"
b = "code"
print(a == b)

#returns False
```

```
b = "code"
c = "code"
print(b == c)

#returns True
```

some examples of `>`, `<`

```
a = 5
b = 3
print(a > b)

#returns True
```

```
a = 5
b = 3
print(a < b)

#returns False
```

## boolean operators

- `and`: This is the Python “logical and” operator. It returns `True` if both expressions are `True` and `False` otherwise
- `or`: The Python “logical or” operator returns `True` if either expression is `True` and `False` otherwise
- `not`: Python’s “logical not” operator returns `True` only if the expression it is evaluating is `False`

```
a = 3
b = 3
c = 4
d = 4
print((a == b) and (c == d))

#returns True
```

```
a = 3
b = 3
c = 4
d = 5
print((a == b) or (c == d))

#returns True
```

```
a = 3
print(not(a))

#returns False
#when a is 3, not a is False
```

- The most important role for Boolean operators is for their use in *conditional statements*
- In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly

- When a Boolean expression evaluates to `True` or `False`, it can be used to control the flow of a program
- Boolean expressions are used in `if` and `else` statements
- the `if` statement consists of a header line and a body. The header line begins with the keyword `if` followed by a *Boolean expression* and ends with a colon `:`
- The indented statements that follow are called a **block**. The first unindented statement marks the end of the block.

## if

when the condition evaluates to True, the statements are executed, otherwise the flow of execution continues to the statement after the if

```
if BOOLEAN_EXPRESSION:
    SOME_STATEMENTS           # Executed if condition evaluates to True
```

example:

```
humidity = 85
if humidity > 80:
    print("It is a humid day")

# returns: It is a humid day
```

## if else

if condition is true, then all the indented statements get executed. If not, then all the statements indented under the `else` clause get executed

```
if BOOLEAN_EXPRESSION:
    STATEMENTS_1             # Executed if condition evaluates to True
else:
    STATEMENTS_2             # Executed if condition evaluates to False
```

## if elif else

Sometimes there are more than two possibilities and we need more than two branches. One way to express a computation like that is a **chained conditional**, we use `elif` statement between `if` and `else` statements

```
if hour <= 9:
    print("It's breakfast time!")
elif hour <= 13:
    print("It's lunch time!")
else:
    print("It's dinner time!")
```

## exercises

Save each python program which you wrote any code for in this format:

p4\_exerciseNum.py e.g. p4\_20.py

For each exercise which does not require you to code, please firstly try to answer the question without writing it as a program. It's a good practice to think through problems and exercises without relying on code right away. If you find yourself stuck or unsure, you can use Python to help you figure out the answer.

1. What are the types of these variables?

- variable1 = 'True'
- variable2 = True
- variable3 = False
- variable4 = 'False'
- variable5 = false
- variable6 = true

2. What will be printed on the screen?

- `print(7 == (5 + 2))`
- `print((3 + 1) == (5 + 2))`

c. `print(10 == (2 * 5))`

d. `print((8 % 3) == 2)`

e. `print((2 ** 3) == 8)`

f. `print((6 // 4) == 1)`

g. `a = 3`

`print(bool(a))`

h. `a = 0`

`print(bool(a))`

3. What will be the result of the following comparisons?

a. `print(5 == 5)`

b. `print(10 != 5)`

c. `print(3 < 7)`

d. `print(8 > 12)`

e. `print(6 <= 6)`

f. `print(9 >= 10)`

4. Are the following strings considered equal?

a. `"hello" == "hello"`

b. `"Hello" == "hello"`

c. `"plant" == "plan"`

d. `"Python" == "python"`

e. `"puppy" == "puppy"`

f. `"computer" == 'computer'`

5. What will be the result of the following comparisons?

a. `print(3 + 4 == 7)`

b. `print(5 * 2 != 10)`

- c. `print(10 - 3 < 8)`
- d. `print(12 / 4 > 2)`
- e. `print(6 % 3 <= 0)`
- f. `print(9 ** 2 >= 81)`

6. Evaluate the following boolean expressions:

- a. `print(3 > 2 and 4 > 5)`
- b. `print(3 > 2 or 4 > 5)`
- c. `print(not (3 > 2))`
- d. `print(not (4 > 5))`
- e. `print(5 > 2 and 4 > 3)`
- f. `print(10 < 5 or 7 < 2)`
- g. `print(not (8 == 8))`
- h. `print(not (5 != 5))`
- i. `print(3 >= 3 and 4 < 2)`
- j. `print(6 <= 6 or 7 > 10)`

7. Write a program that asks the user to enter a number. If the number is greater than 10, print "The number is greater than 10."

8. Write a program that asks the user to enter a character. If the character is a vowel (a, e, i, o, u), print "It is a vowel." You might use one of the boolean operators: and, or, not. You also can convert input to lowercase or uppercase and do your checks: functions: `.lower()` and `.upper()`

9. Write a program that asks the user to enter their age. If the age is greater than or equal to 18, print "You are eligible to vote." Otherwise, print "You are not eligible to vote."



10. Write a program that asks the user to enter a password. If the password is "password123", print "Access granted." Otherwise, print "Access denied."
11. Write a program that asks the user to enter their exam score. If the score is greater than or equal to 60, print "Congratulations! You passed the exam." Otherwise, print "Sorry, you did not pass the exam."
12. Write a program that prompts the user to enter their username. If the username is "admin", print "Welcome, administrator!" Otherwise, print "Welcome, guest."
13. Write a program to read three numbers and display the largest and smallest of the numbers entered. You may compare the numbers yourself, or use functions `max()` and `min()`, search on the Internet for more info
14. Write a program that prompts the user to enter a number. If the number is divisible by 2, print "The number is even." Otherwise, print "The number is odd." Think how you can check that the number is divisible by 2!
15. Write a program that asks the user to enter a year. If the year is a leap year (divisible by 4), print "It is a leap year." Otherwise, print "It is not a leap year." Think how you can check that the number is divisible by 4!
16. Write a program that prompts the user to enter their age and whether they have a driver's license ("yes" or "no"). If the person is 18 or older and has a driver's license, print "You can legally drive." If the person is 18 or older but does not have a driver's license, print "You can apply for a driver's license." If the person is under 18, print "You are not old enough to drive." You might also do some checks if a person didn't enter correct data.
17. Write a program that prompts the user to enter their exam score (out of 100). If the score is 90 or above, print "You got an A!" If the score is between 80 and 89, print "You got a B." If the score is between 70 and 79, print "You got a C." If the

score is between 60 and 69, print "You got a D." If the score is below 60, print "You got an F."

18. Write a program that prompts the user to enter their grade. If the grade is "A", print "Excellent job!" If the grade is "B" or "C", print "Good job!" If the grade is "D" or "F", print "You need to study harder." If user didn't enter any of those grades, print appropriate error message.
  
19. Write a program that asks the user to enter the lengths of three sides of a triangle. Determine if the triangle is equilateral (all sides equal length), isosceles (two sides equal length), or scalene (no sides equal length). Print the corresponding triangle type.
  
20. Write a program that asks the user to enter their age and determines their ticket price for a theme park. If the age is under 5, print "Children under 5 get in for free!". If the age is between 5 and 12 (inclusive), print "Child ticket: \$10". If the age is between 13 and 17 (inclusive), print "Teen ticket: \$15". If the age is 18 or older, print "Adult ticket: \$20". Ask the user to confirm their age and display the corresponding ticket price.
  
21. Write a program for a library that asks the user to enter their age and the number of books they want to borrow.. If the user is under 11 years old, allow them to borrow up to 3 books. If the user is between 11 and 17 years old (inclusive), allow them to borrow up to 5 books. If the user is 18 years or older, allow them to borrow up to 10 books. Display a message confirming the number of books they borrowed, if user entered more than they are allowed to borrow display appropriate warning.
  
22. Write a program for a shopping app that asks the user to enter the total amount of their purchase. If the total amount is less than \$50, apply a 10% discount and display the discounted price. If the total amount is between \$50 and \$100 (inclusive), apply a 20% discount. If the total amount is over \$100, apply a 30% discount. Display the discounted price along with a message indicating the amount saved.

23. Extend your exercise from previous week #23 about BMI, here is how it looked like:

You want to track your fitness progress by calculating your body mass index (BMI). Ask the user to enter their weight in kilograms and their height in meters. Convert the user input to the appropriate data type using the `float()` function. Calculate the BMI by dividing the weight by the square of the height (BMI = weight / (height \* height)) and assign it to a variable called `bmi`.

Delete all the print statements from the program that you wrote.

Based on the calculated BMI, print the corresponding category:

- BMI less than 18.5: "Underweight"
- BMI between 18.5 and 24.9: "Normal weight"
- BMI between 25 and 29.9: "Overweight"
- BMI 30 or greater: "Obese"

24. Write a program that asks the user to enter their height in centimeters. If the entered height is between 150 and 180 centimeters (inclusive), print a message indicating that it is an average height. If the entered height is less than 150 centimeters, print a message indicating that it is a short height. If the entered height is greater than 180 centimeters, print a message indicating that it is a tall height. If the entered value is not a valid height, print an error message and terminate the program.

25. Write a program that asks the user to enter their birth date: day, month and year. Based on the input, print their astrological sign. Use the following table:

- Aries (March 21 – April 19)
- Taurus (April 20 – May 20)
- Gemini (May 21 – June 20)
- Cancer (June 21 – July 22)
- Leo (July 23 – August 22)
- Virgo (August 23 – September 22)

- Libra (September 23 – October 22)
- Scorpio (October 23 – November 21)
- Sagittarius (November 22 – December 21)
- Capricorn (December 22 – January 19)
- Aquarius (January 20 – February 18)
- Pisces (February 19 – March 20)

note: you take user's year, but don't use it in this program to evaluate the astrological sign.

26. Write a program for a shipping company that asks the user to enter the weight and destination of a package. Based on the weight and destination, calculate and display the shipping cost using the following rules:

- For packages weighing up to 1 kilogram, the shipping cost is \$10 for local destinations and \$20 for international destinations.
- For packages weighing between 1 and 5 kilograms (inclusive), the shipping cost is \$15 for local destinations and \$25 for international destinations.
- For packages weighing over 5 kilograms, the shipping cost is \$20 for local destinations and \$30 for international destinations.

note: you might use nested if statements: inside if statement you can use another if statement, you can read more about nested if statements online. But there are many ways to solve this problem!

27. Write a program for a library that asks the user to enter their age and the number of books they want to borrow. If number of books is greater than the limit print an appropriate message. Based on their age and the number of books, calculate the borrowing limit and due date using the following rules:

- For members under 12 years old, they can borrow up to 5 books for 7 days.
- For members between 12 and 18 years old (inclusive), they can borrow up to 10 books for 14 days.
- For members above 18 years old, they can borrow up to 15 books for 21 days. Additionally, if they borrow more than 10 books, they need to pay a deposit of \$20 which will be refunded upon returning all the books.

Print the borrowing limit and the due date as an output.

28. Write a program for a university that asks the user to enter their course, year of study, and current GPA. Based on their inputs, determine their eligibility for scholarships using the following criteria:
- For courses in the Science, Technology, Engineering, and Mathematics (STEM) field and students with a GPA above 3.5, display a message indicating that they are eligible for a STEM scholarship.
  - For courses outside the STEM field and students with a GPA above 3.0, display a message indicating that they are eligible for a general scholarship.
  - For students with a GPA below the eligibility criteria, display a message indicating that they are not eligible for any scholarship.
29. Modify the calculator program `calc3.py` from the section on *Conditional Statements*, in the *Handbook Introduction to Programming* to handle multiplication and division as well as addition and subtraction.
30. Write a program to simulate a cash register for a single purchase. The program should read the unit cost (real number) of an item and the numbers of items purchased. The program should display the total cost for the items. If the unit cost is greater than 10000, the program should display an error message, "Invalid unit cost – too large". If the unit cost is 0 it should display an error message, "Unit cost cannot be 0".

```
Enter unit cost: 5.5
Enter number of units: 10
Total cost: 55.0

Enter unit cost: 0
Unit cost cannot be 0
```

31. Write a program to show a menu of areas to be calculated and to calculate the area chosen by the user. The menu you are to display, is shown in italics below

```
Choose the area you wish to calculate from the menu below
a for the area of a square
b for the area of a circle
c for the area of a rectangle
```

```
Enter your choice:
```

The program should prompt for the dimensions of the area:

length of a side in the case of a (area = length \*\* 2);

radius in the case of b (area = 3.14 \* r\*\*2)

length and breadth in the case of c (area = length \* breadth)

32. Write a program to read a string from the user and display an appropriate message depending on the first character of the string indicating if the character was an uppercase letter, a lowercase letter or a digit ('0' to '9'). You might use some built in functions, search on the Internet for them. Sample outputs are shown below:

```
Enter a string: A
You entered an uppercase letter

Enter a string: b
You entered a lowercase letter

Enter a string: 5
You entered a digit

Enter a string: &
You did not enter a letter or a digit
```

#### EXTRA TASK:

Create a cinema ticket booking program, all data here is given as an example, we encourage you to use your own imagination! You can do slight alterations to this program as long as it has the ticket booking flow. You must use variables, conditional statements, input from the user, arithmetic operations, print statements.

1. Display a menu to the user with different types of tickets and their prices:
  - Standard ticket: \$10.00

- Premium ticket: \$15.00
- VIP ticket: \$20.00

Prompt the user to enter the type of ticket they want to buy.

2. Prompt the user to enter the number of tickets they want to purchase.
3. Display a menu of available movies:
  - Movie 1: "The Avengers"
  - Movie 2: "Jurassic Park"
  - Movie 3: "Harry Potter and the Sorcerer's Stone"

Prompt the user to select a movie by entering the corresponding movie number.

4. Display a menu of available dates and times:
  - Date 1: July 1, 2023, 5:00 PM
  - Date 2: July 2, 2023, 7:30 PM
  - Date 3: July 3, 2023, 2:00 PM

Prompt the user to select a date and time by entering the corresponding date number.

5. Calculate the total cost of the tickets based on the ticket type and quantity selected by the user.
6. Print a summary to the user, including:
  - The selected movie
  - The selected date and time
  - The total cost of the tickets

Finally, print the address of the cinema, store it with other variables.