



# Week 6 materials + answers

## topics covered

- increment & decrement values
- loops
- the while loop

## some recap

- to increment a value, we can use `+=`:

```
a = 2
a += 1

# a will be 3
# a += 1 is the same as a = a + 1 (we are updating value of variable a, on the right hand side we have 2 + 1)
```

- to decrement a value, we can use `-=`:

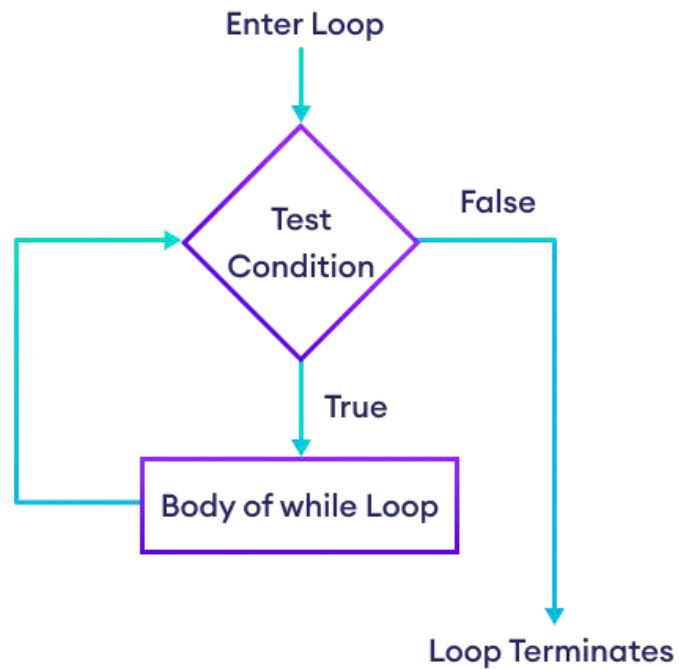
```
a = 3
a -= 1

# a will be 2
# a -= 1 is the same as a = a - 1 (we are updating value of variable a, on the right hand side we have 3 - 1)
```

- we can increment or decrement by any value: `a += 10` `b -= 5`
- loops are used to repeat a block of code: e.g. if we want to show a message 100 times, then we can use a loop
- Python `while` loop is used to run a block code until a certain condition is met

- syntax:

```
while condition:  
    # body of while loop
```



- example 1 involving numbers:

```
# program to display numbers from 1 to 5  
  
# initialize the variable  
i = 1  
n = 5  
  
# while loop from i = 1 to 5  
while i <= n:  
    print(i)  
    i = i + 1
```

output:

```
1  
2  
3  
4  
5
```

- example 2 involving lists:

```
a = ['foo', 'bar', 'baz']  
while a:  
    print(a.pop(-1))
```

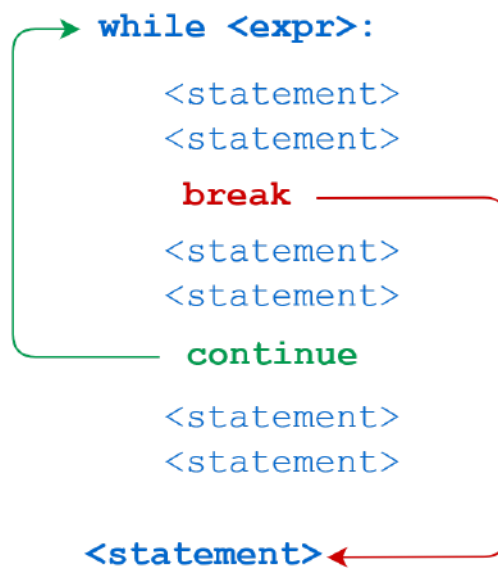
output:

baz

bar

foo

- Python `break` statement immediately terminates a loop entirely
- Python `continue` statement immediately terminates the current loop iteration. Execution jumps to the top of the loop



- a `while` loop that theoretically never ends is called an infinite loop  
`while True:` will be executing statements infinitely

here's some some examples with while loop:

```
# Prints all letters except whitespace ' ', 'e', and 's'  
i = 0  
a = 'hello ladies and gentlemen'  
  
while i < len(a):  
    if a[i] == ' ' or a[i] == 'e' or a[i] == 's' :  
        i += 1  
        continue  
  
    print('Current Letter :', a[i])  
    i += 1
```

```
# Python while loop with user input  
a = int(input('Enter a number (-1 to quit): '))
```

```
while a != -1:
    a = int(input('Enter a number (-1 to quit): '))
```

```
# Initialize a counter
count = 0

# Loop infinitely
while True:
    # Increment the counter
    count += 1
    print(f"Count is {count}")

    # Check if the counter has reached a certain value
    if count == 10:
        # If so, exit the loop
        break

# This will be executed after the loop exits
print("The loop has ended.")
```

## exercises

Save each python program which you wrote any code for in this format: p5\_exerciseNum.py e.g. p5\_20.py

Note: your answers might be slightly different

1. Print 'Hello World' 5 times using While loop.

answer:

```
count = 0
while count < 5:
    print("Hello World")
    count += 1
```

2. Imagine you have a list `my_list = ['apple', 'banana', 'lemon']`. Iterate over the items in the list `my_list` using while loop and print each element.

answer:

```
my_list = ['apple', 'banana', 'lemon']
index = 0
while index < len(my_list):
    print(my_list[index])
    index += 1
```

3. Iterate from `i = 1` to `i = 10`, but break the loop if `i` equals 4. Use while loop, break statement, and print out `i` every time you iterate.

answer:

```
i = 1
while i <= 10:
    if i == 4:
        break
    print(i)
    i += 1
```

4. Iterate from `i = 1` to `i = 10`, but if `i` equals 4, skip the next statements in the loop and continue with the next iteration. Use while loop, break statement, and print out `i` every time you iterate.

answer:

```
i = 1
while i <= 10:
    if i == 4:
        i += 1
        continue
    print(i)
    i += 1
```

5. Write a program that uses a while loop to print numbers from 1 to 10.

answer:

```
num = 1
while num <= 10:
    print(num)
    num += 1
```

6. Write a program that prompts the user to enter a password. Keep asking for the password until the user enters the correct password, which is "password123". Print a success message when the correct password is entered.

answer:

```
password = "password123"
while True:
    user_input = input("Enter the password: ")
    if user_input == password:
        print("Success! Correct password entered.")
        break
    else:
        print("Incorrect password. Try again.")
```

7. Write a program that calculates the sum of all even numbers between 0 and 50 using a while loop.

answer:

```
sum_even = 0
num = 0
while num <= 50:
```

```
sum_even += num
num += 2

print("Sum of even numbers:", sum_even)
```

8. Write a program that prompts the user to enter a positive integer and prints its multiplication table from 1 to 10 using a while loop.

answer:

```
num = int(input("Enter a positive integer: "))
i = 1

while i <= 10:
    result = num * i
    print(num, "*", i, "=", result)
    i += 1
```

9. Write a program that asks the user for a series of numbers until the user enters a negative number. Then, calculate and print the sum of all the entered numbers, do not use lists in this program.

answer:

```
total_sum = 0
number = 0

while number >= 0:
    number = int(input("Enter a number: "))
    if number >= 0:
        total_sum += number

print("Sum of the entered numbers:", total_sum)
```

10. Write a program that allows the user to add items to a shopping list. The program should continue to ask for items until the user enters "done" to stop adding items. Finally, print the complete shopping list

answer:

```
shopping_list = []

while True:
    item = input("Enter an item to add to the shopping list (or 'done' to finish): ")
    if item.lower() == "done":
        break
    shopping_list.append(item)

print(f"Shopping List:\n{shopping_list}")
```

11. Write a program that allows the user to add tasks to a to-do list. The program should prompt the user to add a new task or enter "done" to finish. Once done, print the final to-do list. Use list concept here, you need to append each task.

answer:

```

to_do_list = []

while True:
    task = input("Enter a task to add to the to-do list (or 'done' to finish): ")
    if task.lower() == "done":
        break
    to_do_list.append(task)

print("\nThis is your to-do list:")
index = 0
while index < len(to_do_list):
    print(f"{index + 1}. {to_do_list[index]}")
    index += 1

```

12. Write a program that asks the user to enter their grades for a course. The program should calculate the average grade and provide a corresponding letter grade based on the following scale: 90-100: A, 80-89: B, 70-79: C, below 70: F. To stop entering grades user should enter `-1`

answer:

```

grades = []
while True:
    grade = float(input("Enter a grade (or -1 to finish): "))
    if grade == -1:
        break
    grades.append(grade)

average_grade = sum(grades) / len(grades)

if average_grade >= 90:
    letter_grade = "A"
elif average_grade >= 80:
    letter_grade = "B"
elif average_grade >= 70:
    letter_grade = "C"
else:
    letter_grade = "F"

print("Average Grade:", average_grade)
print("Letter Grade:", letter_grade)

```

13. Write a program to read a string and display it, replacing any occurrence of A in the string by o:

```

Enter a string with an A in it: ABBA
Replacing A with o gives: oBBo

```

answer:

```

# using replace()

string = input("Enter a string with an A in it: ")
modified_string = string.replace('A', 'o')
print("Replacing A with o gives:", modified_string)

```

```

# using while

```

```

string = input("Enter a string with an A in it: ")
modified_string = ""
index = 0

while index < len(string):
    if string[index] == 'A':
        modified_string += 'o'
    else:
        modified_string += string[index]
    index += 1

print("Replacing A with o gives:", modified_string)

```

14. Write a program to read a string and count the number of digits (0,1,2,..9) in it

```

Enter a string: 12abc34def56
There are 6 digits in 12abc34def56

```

Hint: if `s[i] >='0'` **and** `s[i] <= '9'` then `s[i]` contains a digit.

answer:

```

string = input("Enter a string: ")
digit_count = 0

for char in string:
    if char >= '0' and char <= '9':
        digit_count += 1

print("There are", digit_count, "digits in", string)

```

15. Write a program to read a string and count the number of uppercase letters (A..Z) and lowercase letters (a..z) in the string

```

Enter a string : ABC defg
3 uppercase letters and 4 lowercase letters in ABC defg

```

Hint: if `s[i] >='A'` **and** `s[i] <= 'Z'` then `s[i]` contains an uppercase letter.

answer:

```

string = input("Enter a string: ")
uppercase_count = 0
lowercase_count = 0
index = 0

while index < len(string):
    if string[index] >= 'A' and string[index] <= 'Z':
        uppercase_count += 1
    elif string[index] >= 'a' and string[index] <= 'z':
        lowercase_count += 1
    index += 1

print(uppercase_count, "uppercase letters and", lowercase_count, "lowercase letters in", string)

```



16. Write a program that allows the user to create a checklist for party supplies. The program should prompt the user to enter an item for the checklist or enter "done" to finish adding items. Once the user is done, the program should display the final party supplies checklist.

answer:

```
checklist = []

while True:
    item = input("Enter an item for the party supplies checklist (or 'done' to finish): ")
    if item.lower() == "done":
        break
    checklist.append(item)

print("\nParty Supplies Checklist:")
index = 0
while index < len(checklist):
    print(f"{index + 1}. {checklist[index]}")
    index += 1
```

17. Write a program that allows the user to track their daily expenses. The program should prompt the user to enter the amount spent for each expense and categorize them into "food", "transportation", or "other". At the end, calculate and display the total amount spent in each category. To stop entering expenses user should enter -1

answer:

```
food_expenses = 0
transportation_expenses = 0
other_expenses = 0

while True:
    expense = float(input("Enter the amount spent (or -1 to finish): "))
    if expense == -1:
        break
    category = input("Enter the category (food/transportation/other): ")
    if category.lower() == "food":
        food_expenses += expense
    elif category.lower() == "transportation":
        transportation_expenses += expense
    else:
        other_expenses += expense

print("Total food expenses:", food_expenses)
print("Total transportation expenses:", transportation_expenses)
print("Total other expenses:", other_expenses)
```

18. Write a program that asks the user to enter a series of numbers. The program should store each number in a list and stop accepting numbers when the user enters 0. At the end, calculate and display the sum of all the entered numbers, you can use `sum()`

answer:

```
numbers = []

while True:
    number = float(input("Enter a number (or 0 to finish): "))
    if number == 0:
        break
```

```

numbers.append(number)

total_sum = sum(numbers)
print("Sum of the entered numbers:", total_sum)

```

19. Write a program that allows the user to enter their daily steps and displays the total steps and average steps. Store user's input in a list. Use `sum()` and `len()`

answer:

```

steps = []

while True:
    daily_steps = int(input("Enter your daily steps (or -1 to finish): "))
    if daily_steps == -1:
        break
    steps.append(daily_steps)

total_steps = sum(steps)
average_steps = total_steps / len(steps)

print("Total Steps:", total_steps)
print("Average Steps:", average_steps)

```

20. Write a program that allows the user to track daily temperatures. The program prompts the user to enter the temperature for each day and stores them in a list. It then calculates the highest temperature, lowest temperature, and average temperature. Display the results at the end. Stop receiving input from the user when they enter -999.

answer:

```

temperatures = []

while True:
    temperature = float(input("Enter the temperature for the day (or -999 to finish): "))
    if temperature == -999:
        break
    temperatures.append(temperature)

highest_temperature = max(temperatures)
lowest_temperature = min(temperatures)
average_temperature = sum(temperatures) / len(temperatures)

print("Highest Temperature:", highest_temperature)
print("Lowest Temperature:", lowest_temperature)
print("Average Temperature:", average_temperature)

```

21. Write a program that allows a student to track their daily study time. The program prompts the user to enter their study time for each day and stores them in a list. Calculate the total study time and average study time per day. Display the results at the end. Stop receiving input from the user when they enter -1.

answer:

```

study_time_list = []

while True:

```

```

study_time = float(input("Enter your study time for the day (in hours), or -1 to finish: "))
if study_time < 0:
    break
study_time_list.append(study_time)

total_study_time = sum(study_time_list)
average_study_time = total_study_time / len(study_time_list)

print("Total Study Time:", total_study_time, "hours")
print("Average Study Time per Day:", average_study_time, "hours")

```

22. Write a program that allows the user to create a music playlist. The program prompts the user to enter song titles and stores them in a list. The user can continue adding songs until they choose to stop, and then the program displays the final playlist.

answer:

```

playlist = []

while True:
    song_title = input("Enter a song title to add to the playlist (or 'done' to finish): ")
    if song_title.lower() == "done":
        break
    playlist.append(song_title)

print("\nMusic Playlist:")
index = 0
while index < len(playlist):
    print(f"{index + 1}. {playlist[index]}")
    index += 1

```

23. Using a `while` loop and an `if` statement append all the elements in a list to a new list unless the element is an empty string `""`.

answer:

```

original_list = [1, "", 3, "", 5, 6, "", 8]
new_list = []
index = 0

while index < len(original_list):
    if original_list[index] != "":
        new_list.append(original_list[index])
    index += 1

print("New List:", new_list)

```

24. Write a python program to read a number and print a right triangle using `*` E.g. :

Input : 5

-----Output-----

```

*
* *
* * *

```

\*\*\*\*

\*\*\*\*\*

answer:

```
n = int(input('Enter number of rows : '))

i = 1
while i <= n :
    print("* " * i)
    i += 1
```

24. Write a program using a loop to
- read number of hours worked per week (maximum is 100)
  - read rate per hour (max is 50)
  - compute gross pay (pay before tax)
  - compute tax at 10% of pay if pay > 100
  - tax is 0 if pay <100
  - display result as:

```
Gross Pay: 200
Rate: 10
Hours worked: 20
Tax: 20
Net Pay: 180

Press Y to continue for another employee
```

answer:

```
while True:
    hours_worked = float(input("Enter the number of hours worked per week (maximum 100): "))
    rate_per_hour = float(input("Enter the rate per hour (maximum 50): "))

    gross_pay = hours_worked * rate_per_hour

    if gross_pay > 100:
        tax = gross_pay * 0.1
    else:
        tax = 0

    net_pay = gross_pay - tax

    print("Gross Pay:", gross_pay)
    print("Rate:", rate_per_hour)
    print("Hours Worked:", hours_worked)
    print("Tax:", tax)
    print("Net Pay:", net_pay)

    choice = input("Press Y to continue for another employee (or any other key to exit): ")
    if choice.lower() != "y":
        break
```

25. Write a program that allows the user to calculate their monthly expenses based on their budget and expenses for each category. The program should prompt the user to enter the budget amount and then ask for expenses in different categories. It should calculate the total expenses, determine if the user exceeded the budget, and display the results. The program should offer the option to calculate expenses for another month.

answer:

```
while True:
    budget = float(input("Enter your monthly budget: "))
    total_expenses = 0

    while True:
        expense_category = input("Enter an expense category (or 'done' to finish): ")
        if expense_category.lower() == "done":
            break
        expense = float(input("Enter the expense amount for the category: "))
        total_expenses += expense

    if total_expenses > budget:
        exceeded_amount = total_expenses - budget
        print("You exceeded your budget by", exceeded_amount)
    else:
        print("You stayed within your budget.")

    choice = input("Press Y to calculate expenses for another month (or any other key to exit): ")
    if choice.lower() != "y":
        break
```

26. Write a program that allows the user to track their daily tasks. The program prompts the user to enter task names and their estimated durations. It then calculates the total time estimated for all tasks and determines if it exceeds a predefined limit of 8 hours. The program displays the result and offers the option to track tasks for another day.

answer:

```
while True:
    total_time_estimated = 0
    task_count = int(input("Enter the number of tasks for the day: "))
    counter = 0
    while counter < task_count:
        task_name = input("Enter the task name: ")
        task_duration = float(input("Enter the estimated duration for the task: "))
        total_time_estimated += task_duration
        counter += 1

    time_limit = 8 # Predefined limit of 8 hours

    if total_time_estimated > time_limit:
        exceeded_time = total_time_estimated - time_limit
        print("Total time estimated exceeds the limit by", exceeded_time)
    else:
        print("Total time estimated is within the limit.")

    choice = input("Press Y to track tasks for another day (or any other key to exit): ")
    if choice.lower() != "y":
        break
```

27. Write a program that simulates a library catalog. Allow the user to add books by entering the title and author. Provide options to search for a book by title, display all books, or exit the catalog. Create different lists for different data.

answer:

```
library_books = []
library_authors = []
choice = ""

while choice != 4:
    print("\nLibrary Catalog Menu:")
    print("1. Add a book")
    print("2. Search for a book by title")
    print("3. Display all books")
    print("4. Exit")

    choice = int(input("Enter your choice (1-4): "))

    if choice == 1:
        title = input("\nEnter the book title: ")
        author = input("Enter the book author: ")
        library_books.append(title)
        library_authors.append(author)
        print("Book added successfully.")
    elif choice == 2:
        title = input("\nEnter the book title to search: ")
        if title in library_books:
            index = library_books.index(title)
            print(f"The book '{title}' by {library_authors[index]} is at index {index}.")
        else:
            print("No book found with the given title.")
    elif choice == 3:
        print("\nLibrary Catalog:")
        index = 0
        while index < len(library_books):
            print("Title:", library_books[index])
            print("Author:", library_authors[index])
            index += 1
    elif choice == 4:
        break
    else:
        print("Invalid choice. Please try again.")
```

28. Write a program that simulates a phonebook. Allow the user to add contacts by entering a name and phone number. Provide options to search for a contact by name, display all contacts, or exit the phonebook.

answer:

```
contact_name = []
contact_number = []
choice = ""

while choice != 4:
    print("\nPhonebook Menu:")
    print("1. Add a contact")
    print("2. Search for a contact by name")
    print("3. Display all contacts")
    print("4. Exit the program")

    choice = int(input("Enter your choice (1-4): "))

    if choice == 1:
        name = input("\nEnter the contact name: ")
```

```

phone_number = input("Enter the contact phone number: ")
contact_name.append(name)
contact_number.append(phone_number)
print("Contact added successfully.")
elif choice == 2:
    name = input("\nEnter the contact name to search: ")
    if name in contact_name:
        index = contact_name.index(name)
        print(f"{name}'s number is {contact_number[index]}.")
    else:
        print("No contacts found with the given name.")
elif choice == 3:
    print("\nPhonebook Contacts:")
    index = 0
    while index < len(contact_name):
        print("Name:", contact_name[index], "\tPhone number:", contact_number[index])
        index += 1
elif choice == 4:
    break
else:
    print("Invalid choice. Please try again.")

```

## FINAL PROJECT:

Extend exercise number 19 from week 2:

Create a program that simulates a hospital registration system. Prompt the user to enter the following information:

- Name
- Surname
- Age
- Height (in cm)
- Health complaints

Store each piece of information in a separate variable with an appropriate name.

Finally, print the information in the following format:

Name: [Name]

Surname: [Surname]

Age: [Age]

Height: [Height]

Health complaints: [Health Complaints]

Replace [Name], [Surname], [Age], [Height], and [Health Complaints] with the actual values entered by the user.

Here is what you need to add:

1. you need to ask how many patients a user wants to enter data for
2. create separate lists for each piece of information e.g. name, surname etc
3. then ask all information for those patients
4. store data in appropriate lists with appropriate index

5. after user completes entering all information about the patients ask if they want to see information about particular patient, they should use surname to search for that patient, find the index of that patient and print all info about them
6. user might want to see information about several patients, so consider using while loop and each time ask user if they want to see the info about the patient
7. when user enters 'no' stop executing the program

Use user friendly outputs, and all of the concepts that you studied previous weeks. You might also check for erroneous input (optional).

answer:

```

num_patients = int(input("How many patients would you like to enter data for? "))

names = []
surnames = []
ages = []
heights = []
complaints = []

counter = 0
print("\nPatient Registration:")

while counter < num_patients:
    print(f"\nPatient {counter + 1}:")
    name = input("Enter the patient's name: ")
    surname = input("Enter the patient's surname: ")
    age = input("Enter the patient's age: ")
    height = input("Enter the patient's height (in cm): ")
    complaint = input("Enter the patient's health complaints: ")

    names.append(name)
    surnames.append(surname)
    ages.append(age)
    heights.append(height)
    complaints.append(complaint)

    counter += 1

while True:
    print("\nPatient Information Menu:")
    print("1. Search for a patient")
    print("2. Exit")

    choice = int(input("Enter your choice (1-2): "))

    if choice == 1:
        surname = input("Enter the patient's surname to search: ")
        if surname in surnames:
            index = surnames.index(surname)
            print("\nPatient Information:")
            print("Name:", names[index])
            print("Surname:", surnames[index])
            print("Age:", ages[index])
            print("Height:", heights[index], "cm")
            print("Health Complaints:", complaints[index])
        else:
            print("No patient found with the given surname.")
    elif choice == 2:
        print("Exiting the program. Goodbye!")
        break
    else:
        print("Invalid choice. Please try again.")

```