웹프로그래밍의 기초

Week6

while; intput; File IO

while

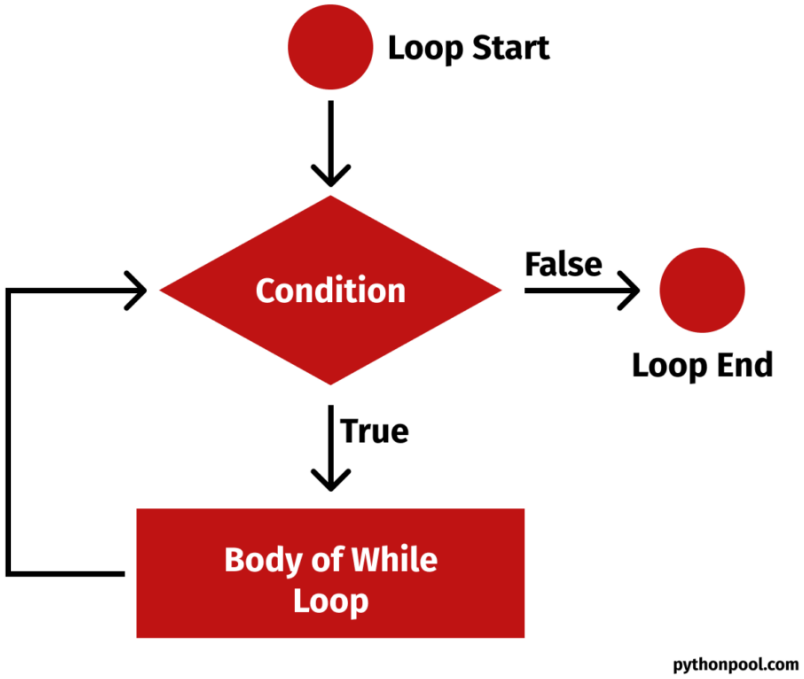
while

- With the `while` loop we can execute a set of statements as long as a condition is true.

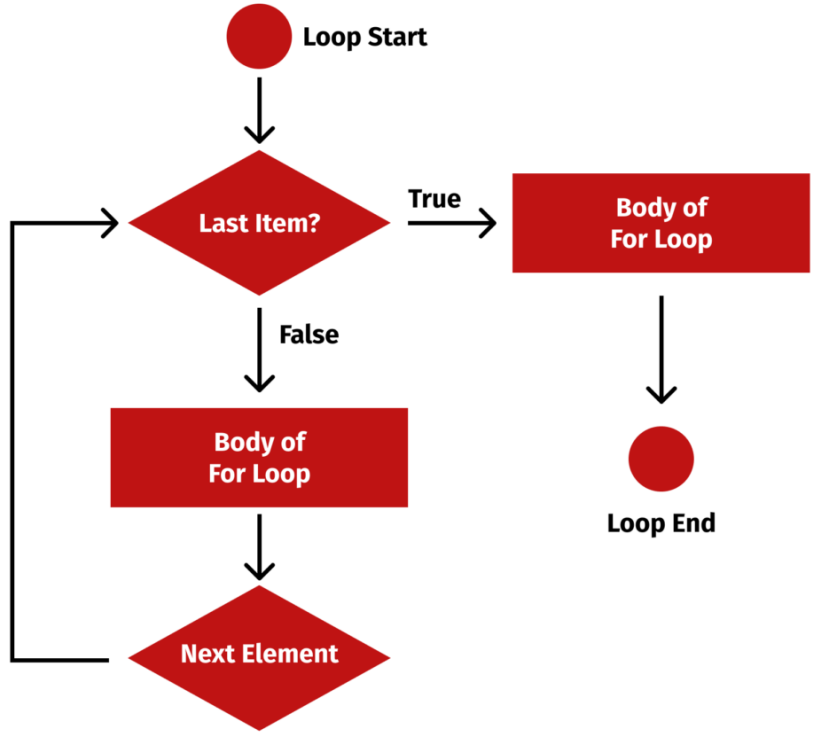
```
i = 1
while i < 6:
  print(i)
  i += 1
--------------------
1
2
3
4
5
```

while and for

- while



- for



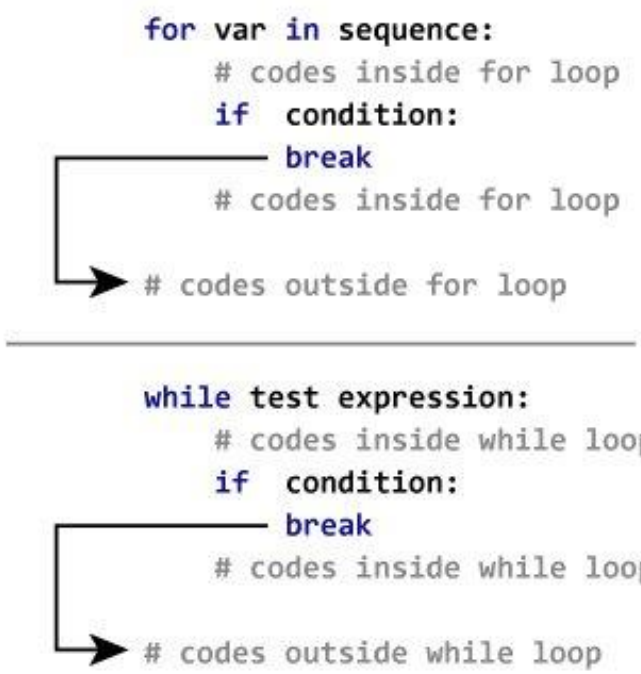
For vs While Loop

Comparison Chart

For Loop	While Loop
The for loop is used for definite loops when the number of iterations is known.	The while loop is used when the number of iterations is not known.
For loops can have their counter variables declared in the declaration itself.	There is no built-in loop control variable with a while loop.
This is preferable when we know exactly how many times the loop will be repeated.	The while loop will continue to run infinite number of times until the condition is met.
The loop iterates infinite number of times if the condition is not specified.	If the condition is not specified, it shows a compilation error.

break in loop statement

- The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.




```
for i in range(6):
    if i ==4:
        break
    print(i)
```

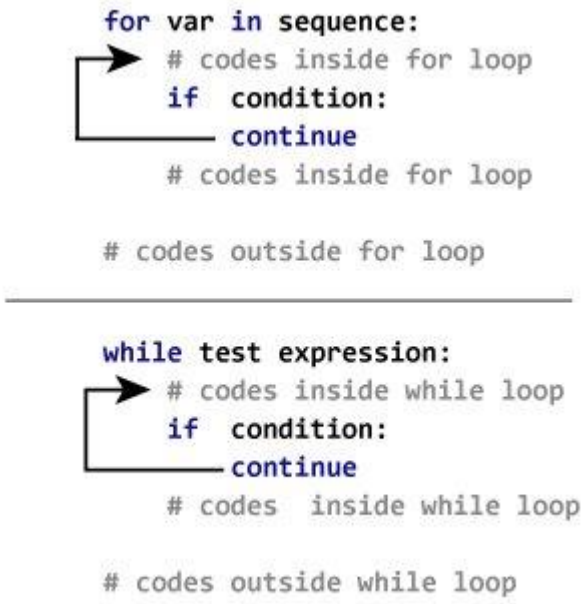
```
0
1
2
3
```

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        break
    print(i)
```

```
1
2
```

continue in loop statement

- The continue statement is used to skip the rest of the code inside a loop for the current iteration only.




```
for i in range(6):
    if i ==4:
        continue
    print(i)
--------------------------------
0
1
2
3
5
```

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
------------------------
1
2
4
5
6
```

input()

input

- Python allows for user input, which means we are able to ask the user for input.
  - Python 3.6 uses the input() method.
  - Python 2.7 uses the raw_input() method.
- Python stops executing when it comes to the input() function, and continues when the user has given some input.

```
username = input("Enter username:")
print("Username is: " + username)
```

```
--------------------------
Enter username:kim
Username is: kim
```

input() converts input into string

- Input() takes user's input values only as string. In case you need to handle that in any other forms of datatype, you should cast the datatype you want to use on the input.

```
byear = int(input("Enter the year your were born: "))
if byear <= 2002:
    print("You are an adult.")
print(type(byear))
```

```
--------------------------
Enter the year your were born: 1945
You are an adult.
<class 'int'>>
```

input() with while

- With combining input() and break or continue statements in while, you can control the flow of the program.

```
prompt  = "\nTell me something, then I will repeat it back to you."
prompt += "\nEnter 'quit', if you want to end this program."
prompt += "\n: "

message = ""
while message != 'quit':
    message = input(prompt)
    if message == 'quit':
        print("Thank you, bye.")
    else:
        print("You said \"" + message + "\".")
```

```
--------------------------------------------------
Tell me something, then I will repeat it back to you.
Enter 'quit', if you want to end this program.
: hi i am a human
You said "hi i am a human".

Tell me something, then I will repeat it back to you.
Enter 'quit', if you want to end this program.
: quit
Thank you, bye.
>
```

File IO

File io method in Python

- The key function for working with files in Python is the open() function.
  - The open() function takes two parameters; filename, and mode.
  - There are four different methods (modes) for opening a file:
    - "r" - Read - Default value. Opens a file for reading, error if the file does not exist
    - "a" - Append - Opens a file for appending, creates the file if it does not exist
    - "w" - Write - Opens a file for writing, creates the file if it does not exist
    - "x" - Create - Creates the specified file, returns an error if the file exists

Open and read the entire file

```
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$ cat io_text.txt
About ALA

The American Library Association (ALA) is the oldest and largest library associa
tion in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the
 mission of ALA is “to provide leadership for the development, promotion and imp
rovement of library and information services and the profession of librarianship
 in order to enhance learning and ensure access to information for all.”
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$
```

```
with open ('io_text.txt') as file_object:
    contents = file_object.read()
    print(contents)
```

```
/home/scott/anaconda3/envs/wpProject/bin/python /home/scott/PycharmProjects/wpProject/main.py
About ALA

The American Library Association (ALA) is the oldest and largest library association in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the mission of ALA is
 “to provide leadership for the development, promotion and improvement of library and information
 services and the profession of librarianship in order to enhance learning and ensure access to
 information for all.”


Process finished with exit code 0
```

Open and read the entire file, then returns each line with loop

```
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$ cat io_text.txt
About ALA

The American Library Association (ALA) is the oldest and largest library associa
tion in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the
 mission of ALA is “to provide leadership for the development, promotion and imp
rovement of library and information services and the profession of librarianship
 in order to enhance learning and ensure access to information for all.”
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$
```

```
with open ('io_text.txt') as file_objects:
    for line in file_objects:
        print(line.rstrip())
```

```
/home/scott/anaconda3/envs/wpProject/bin/python /home/scott/PycharmProjects/wpProject/main.py
About ALA

The American Library Association (ALA) is the oldest and largest library association in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the mission of ALA is
 “to provide leadership for the development, promotion and improvement of library and information
 services and the profession of librarianship in order to enhance learning and ensure access to
 information for all.”


Process finished with exit code 0
```

Open and read the entire file, then returns each line as a list item

```
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$ cat io_text.txt
About ALA

The American Library Association (ALA) is the oldest and largest library associa
tion in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the
 mission of ALA is “to provide leadership for the development, promotion and imp
rovement of library and information services and the profession of librarianship
 in order to enhance learning and ensure access to information for all.”
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$
```

```
with open ('io_text.txt') as file_objects:
    lines = file_objects.readlines()

for line in lines:
    print(line.rstrip())
```

```
/home/scott/anaconda3/envs/wpProject/bin/python /home/scott/PycharmProjects/wpProject/main.py
About ALA

The American Library Association (ALA) is the oldest and largest library association in the world.

Founded on October 6, 1876 during the Centennial Exposition in Philadelphia, the mission of ALA is
 “to provide leadership for the development, promotion and improvement of library and information
 services and the profession of librarianship in order to enhance learning and ensure access to
 information for all.”


Process finished with exit code 0
```

Open and save the new file

- opening the file object with ‘w’ flag in open() allows you to create the new file.

```
filename    = 'wp_text.txt'
message     = "Thank you.\n"
message     += "Thank you, again\n"

with open(filename, 'w') as file_object:
    file_object.write(message)
```

```
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$ cat wp_text.txt
Thank you.
Thank you, again
 (base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$
```

Open the file, and append to it

- opening the file object with ‘a’ flag in open() allows you to add something into the end of the existing file.

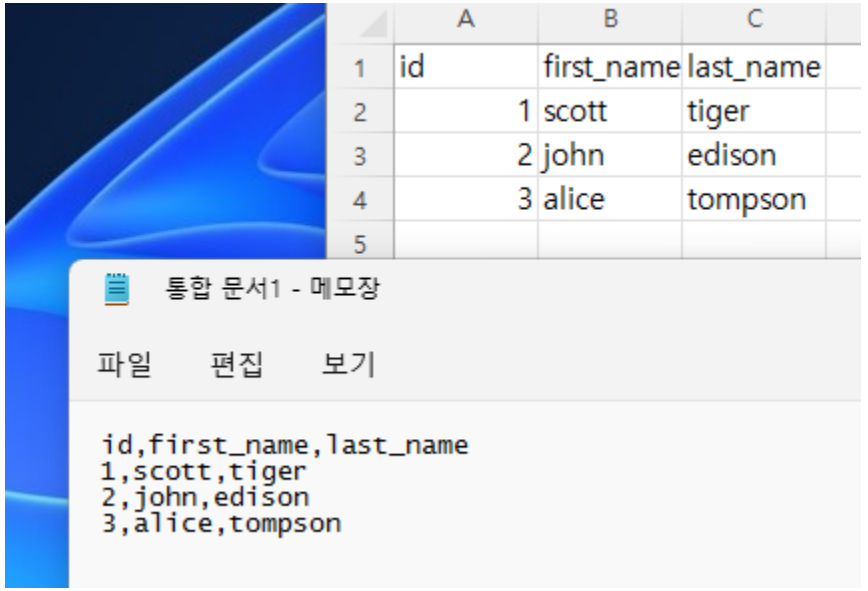
```
filename    = 'wp_text.txt'
message     = "I forgot saying thank you.\n"

with open(filename, 'a') as file_object:
    file_object.write(message)
```

```
(base) scott@scott-virtual-machine:~/PycharmProjects/wpProject$ cat wp_text.txt
Thank you.
Thank you, again
I forgot saying thank you.
```

csv file

- A comma-separated values file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

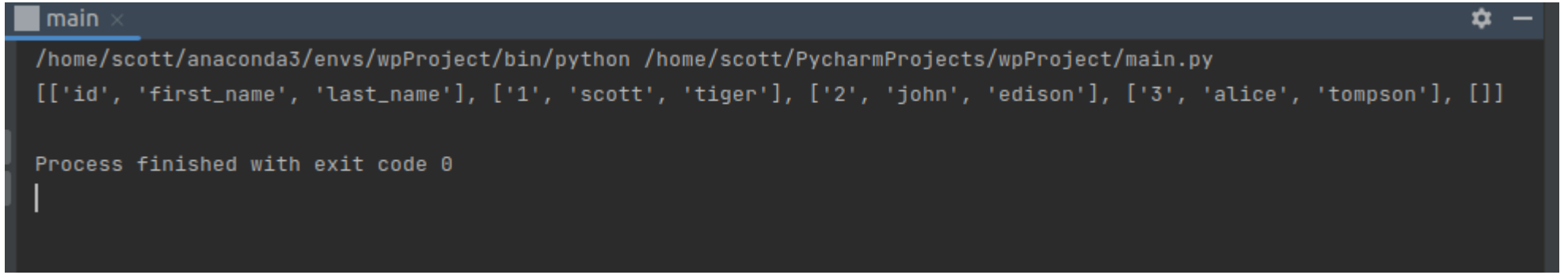


Read csv file into list

```
id,first_name,last_name
1,scott,tiger
2,john,edison
3,alice,tompson
```

```
import csv

with open("wp_csv.csv", "r") as csv_object:
          data = list(csv.reader(csv_object, delimiter=","))
          print(data)
```



Read csv file into dictionary

```
id,first_name,last_name
1,scott,tiger
2,john,edison
3,alice,tompson
```

```
import csv

with open("wp_csv.csv", "r") as csv_object:
          data = list(csv.DictReader(csv_object, delimiter=","))
          print(data)
```

```
/home/scott/anaconda3/envs/wpProject/bin/python /home/scott/PycharmProjects/wpProject/main.py
[{'id': '1', 'first_name': 'scott', 'last_name': 'tiger'}, {'id': '2', 'first_name': 'john', 'last_name':
 'edison'}, {'id': '3', 'first_name': 'alice', 'last_name': 'tompson'}]

Process finished with exit code 0
```

All things considered, so far.

```
id,first_name,last_name
1,scott,tiger
2,john,edison
3,alice,tompson
```

```
import csv

your_first_name       = input("Enter your FIRST name to check if you are on the list:\n")
your_last_name        = input("Enter your LAST name to check if you are on the list:\n")
onlist                = False

with open("wp_csv.csv", "r") as csv_object:
          data = list(csv.DictReader(csv_object, delimiter=","))

          for datum in data:
                    if (your_first_name == datum['first_name']) and (your_last_name == datum['last_name']):
                              onlist = True
                              break
                    else:
                              pass

          if onlist:
                    print("Your name is on the list.")
          else:
                    print("Your name is NOT on the list.")
```

Exception

- Python has built-in exceptions which can output an error. If an error occurs while running the program, it's called an exception.
- If an exception occurs, the type of exception is shown. Exceptions needs to be dealt with or the program will crash. To handle exceptions, the try-catch block is used.
- Python Built-in Exceptions¶
  - https://docs.python.org/3/library/exceptions.html

```
print(5/0)
-------------------------------------------
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ZeroDivisionError: division by zero
```

```
filename = 'data'
with open(filename, 'r') as file:
    print(file.read())
-------------------------------------------
Traceback (most recent call last):
  File "main.py", line 5, in
    with open(filename, 'r') as file:
FileNotFoundError: [Errno 2] No such file or directory: 'data'
```

Exception handling

- When an error occurs, or exception as we call it, Python will normally stop and generate an error message.
  - These exceptions can be handled using the try statement.
  - Since the try block raises an error, the except block will be executed.

```
print(5/0)
```

```
--------------------------------------------
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ZeroDivisionError: division by zero
```

```
try:
    print(5/0)
except ZeroDivisionError:
    print('Divided by zero')
```

```
--------------------------------------------
Divided by zero
>
```

- Without the try block, the program will crash and raise an error. You can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error:

```
try:
  print(x/0)
except NameError:
  print("Variable x is not defined")
except:
  print("Something else went wrong")
```

```
--------------------------------------------
Variable x is not defined
```

```
x=0
try:
  print(x/0)
except NameError:
  print("Variable x is not defined")
except:
  print("Something else went wrong")
```

```
--------------------------------------------
Something else went wrong
```