Flow Control

WHILE Loops

WHILE loops

Creates a condition and while it is True the block of code below is repeatedly executed. Once it is not true, the execution stops. (Checked in between loops.)

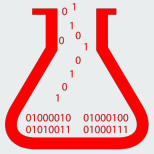
Format:

```
while boolean_condition:
#code to execute
```

The block of code is indented with a tab (or 4 spaces)

The while block ends when the indentation ends

The format is the same a simple IF statement. It uses the same kinds of boolean conditions. But, a WHILE loop repeats until the boolean condition evaluates as False when it's next evaluated, while IF statements just execute once (if the boolean condition is true).

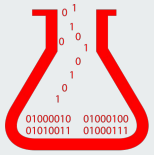


WHILE loops

Example:

```
>>  Index = 0
      while Index <10:
          #code to do
          print(Index)
          #increment index
          Index += 1
```

Output: (0,1,2,3,4,5,6,7,8,9)



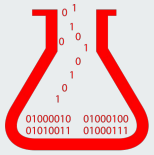
WHILE loop teaching example

What if we want to check if every element in "x" is even?

```
x = [4,5,6,6.5,7]
i = 0
# i is commonly used for the index of an object
# while loops require explicitly instantiating the counter

while i < len(x):
    if x[i] % 2 == 0:
                        # % is called the modulus operator
                        # % yields the remainder from the division of the first argument by the second
        print(x[i], "is even")
    else:
        print(x[i], "is not even")

    i += 1
```



WHILE loop teaching example

```
x = [4,5,6,6.5,7]
i = 0

while i < len(x):
    if x[i] % 2 == 0:
        print(x[i], "is even")
    else:
        print(x[i], "is not even")

    i += 1
```

Output:

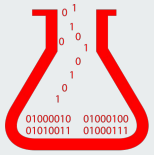
4 is even

5 is not even

6 is even

6.5 is not even

7 is not even



WHILE loops - Break

Without a stopping condition, the loop throws an error. Use **break** to gracefully stop a loop.

```
1  x = [0,1,2,3,4,5]
2  while True:
3      x.pop()
4      print(x)
```

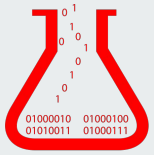
```
[0, 1, 2, 3, 4]
[0, 1, 2, 3]
[0, 1, 2]
[0, 1]
[0]
[]

---------------------------------------------------------------------------
IndexError                                Trace
<ipython-input-70-bbc18fe3391b> in <module>()
      1 x = [0,1,2,3,4,5]
      2 while True:
----> 3     x.pop()
      4     print(x)

IndexError: pop from empty list
```

```
1  x = [0,1,2,3,4,5]
2  while True:
3      x.pop()
4      print(x)
5      if len(x) < 2:
6          break
```

```
[0, 1, 2, 3, 4]
[0, 1, 2, 3]
[0, 1, 2]
[0, 1]
[0]
```



WHILE loops - Continue

Using “continue” passes the control back to the top of the loop without exiting the loop, allowing certain conditions to not engage the rest of the loop.

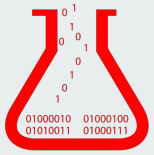
```
1  x = [0,1,2,3,4,5]
2  while True:
3      x.pop()
4      if x[-1] == 2 or x[-1] == 4:
5          continue
6      print(x)
7      if len(x) < 2:
8          break
```

```
[0, 1, 2, 3]
[0, 1]
[0]
```

Nested WHILE loops:

Can insert while loops inside each other!

As more loops are ‘nested’ inside of each other, need to be more cognisant of accidentally making an infinite loop (a loop that never reaches the conditions to terminate).

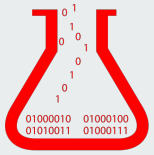


Nested WHILE example (find the prime numbers):

```
i = 2
while i < 100:
    j = 2
    while j <= i/j:
        if i%j == 0:
            break
        j += 1
    if j > i/j:
        print(i, "is prime")
    i += 1

print("Complete!")
```

Read this carefully, and try to figure out what's going on here. We didn't go through this one in detail in class, but going through this one is good practice to better understand while loops (even if you get a little lost in the algorithm for identifying primes).

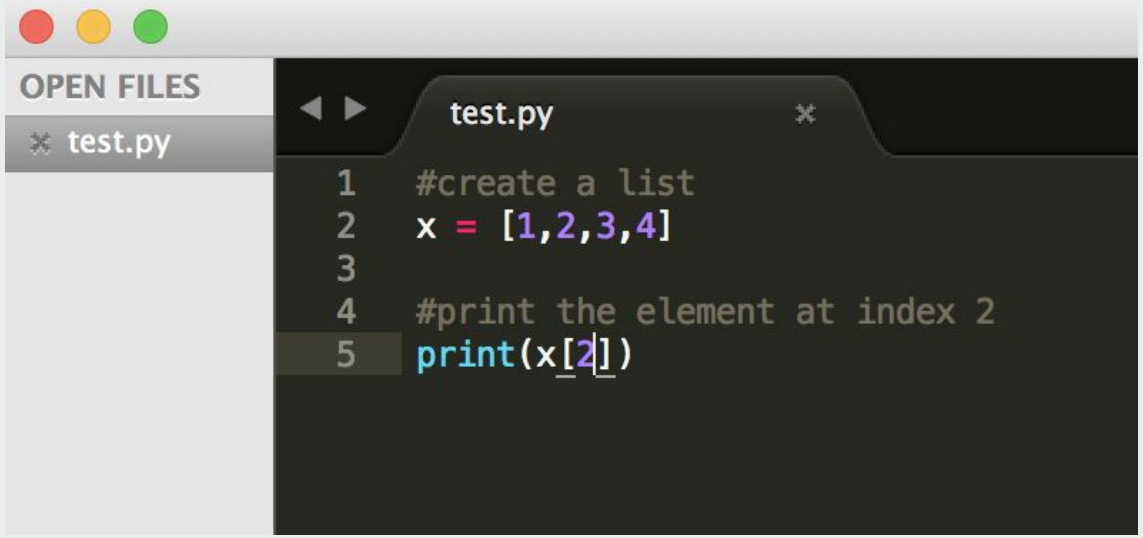


Run Python from Command Line

Create a .py file of the code that want to run

Syntax to run it:

>> python /path/to/file.py




```
Last login: Mon Oct  1 10:43:25 on ttys003
You have new mail.
Bens-MacBook-Pro:~ benthompson$ python ~/Desktop/test.py
3
Bens-MacBook-Pro:~ benthompson$
```

Run Python from Command Line

Create .py of the code that want to run

Syntax to run it:

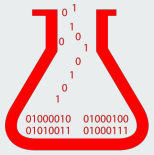
>> python /path/to/file.py

```
Bens-MacBook-Pro:~ benthompson$ python ~/Desktop/test_2.py
What is your name: Ben
Hello,  Ben
Bens-MacBook-Pro:~ benthompson$
```

Practice Problem

Full instructions:

Create a standalone python script (.py file, not a .ipynb notebook) that asks a user for an input, does something (eg, prints user's input) repeatedly in an infinite loop, but quits if the first letter of the user input is a 'q' or a 'Q'



Questions?

Contact:

Denis Vrdoljak
denis@bds.group