Loop (1A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

While loop

```
>>> spam = 0
>>> while spam < 5:
...         print('Hello, world.')
...         spam = spam + 1
...
# Hello, world.
# Hello, world.
# Hello, world.
# Hello, world.
# Hello, world.
```

http://sixthresearcher.com/wp-content/uploads/2016/12/Python3_reference_cheat_sheet.pdf

Loops

```
while <condition>:
    <code>

for <variable> in <list>:
    <code>

for <variable> in
range(start,stop,step):
    <code>

for key, value in
dict.items():
    <code>
```

http://sixthresearcher.com/wp-content/uploads/2016/12/Python3_reference_cheat_sheet.pdf

Loop control statements

break	finishes loop execution
continue	jumps to next iteration
pass	does nothing

http://sixthresearcher.com/wp-content/uploads/2016/12/Python3_reference_cheat_sheet.pdf

Loop control statements

statements block executed as long as condition is true

while logical condition:
    statements block

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

While loop

```
# initializations before the loop
s = 0
i = 1

# condition with a least one variable value (here i)

while i <= 100:
    s = s + i**2
    i = i + 1   # make condition variable change !
    print("sum:",s)
```

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

Break

If the execution reaches a break statement,
it immediately exits the while loop's clause:

```
>>> while True:
...         name = input('Please type your name: ')
...         if name == 'your name':
...             break
...
>>> print('Thank you!')
# Please type your name: your name
# Thank you!
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

Continue

When the program execution reaches a **continue** statement,
the program execution immediately jumps back
to the start of the loop.

```
>>> while True:
...         name = input('Who are you? ')
...         if name != 'Joe':
...             continue
...         password = input('Password? (It is a fish.): ')
...         if password == 'swordfish':
...             break
...
>>> print('Access granted.')
# Who are you? Charles
# Who are you? Debora
# Who are you? Joe
# Password? (It is a fish.): swordfish
# Access granted.
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

For

The for loop iterates over a list, tuple, dictionary, set or string:

```
>>> pets = ['Bella', 'Milo', 'Loki']
>>> for pet in pets:
...     print(pet)
...
# Bella
# Milo
# Loki
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

For

statements block executed for each
item of a container or iterator

for var **in** sequence**:**
    statements block

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

For

```
# Go over sequence's values
# initializations before the loop
s = "Some text"
cnt = 0

# loop variable, assignment managed by for statement
for c in s:
        if c == "e":
                cnt = cnt + 1
print("found",cnt,'"e"')
```

loop on dict/set ⇔ loop on keys sequences
use slices to loop on a subset of a sequence

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

For

```
# Go over sequence's index
# modify item at index
# access items around index (before / after)
lst = [11,18,9,12,23,4,17]
lost = []
for idx in range(len(lst)):
        val = lst[idx]
        if val > 15:
                lost.append(val)
                lst[idx] = 15
        print("modif:",lst,"-lost:",lost)
```

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

For

Go simultaneously over sequence's index and values:

```
for idx,val in enumerate(lst):
```

https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

The range() function

The range() function returns a sequence of numbers. It starts from 0, increments by 1, and stops before a specified number:

```
>>> for i in range(5):
...     print(f'Will stop at 5! or 4? ({i})')
...
# Will stop at 5! or 4? (0)
# Will stop at 5! or 4? (1)
# Will stop at 5! or 4? (2)
# Will stop at 5! or 4? (3)
# Will stop at 5! or 4? (4)
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

For else statement

This allows to specify a statement to execute in case of the full loop has been executed. Only useful when a break condition can occur in the loop:

```
>>> for i in [1, 2, 3, 4, 5]:
...     if i == 3:
...         break
...     else:
...         print("only executed when no item is equal to 3")
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

Ending a Program with sys.exit()

exit() function allows exiting Python.

```
>>> import sys

>>> while True:
...         feedback = input('Type exit to exit: ')
...         if feedback == 'exit':
...                 print(f'You typed {feedback}.')
...                 sys.exit()
...
# Type exit to exit: open
# Type exit to exit: close
# Type exit to exit: exit
# You typed exit
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

The range() function

The range() function can also modify it's 3 defaults arguments. The first two will be the start and stop values, and the third will be the step argument. The step is the amount that the variable is increased by after each iteration.

```
# range(start, stop, step)
>>> for i in range(0, 10, 2):
...     print(i)
...
# 0
# 2
# 4
# 6
# 8
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

The range() function

You can even use a negative number for the step argument to make the for loop count down instead of up.

```
>>> for i in range(5, -1, -1):
...     print(i)
...
# 5
# 4
# 3
# 2
# 1
# 0
```

https://programmingwithmosh.com/wp-content/uploads/2019/02/Python-Cheat-Sheet.pdf

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun