



Basic Python #3

AI Mentorship



Outline

1. Python: For Loop
2. Python: Range
3. Python: For Loop - Program Example
4. Python: While Loop
5. Python: Continue
6. Python: Break
7. Python: Nested Loop



Python: For Loop

Python has two primitive loop commands:

1. `for` loops
2. `while` loops

A `for` loop is used for iterating over a sequence

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```



Python: Range

The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

`range(start, stop, step)`

```
x = range(6)
for n in x:
    print(n)
```

```
x = range(3, 6)
for n in x:
    print(n)
```

```
x = range(3, 20, 2)
for n in x:
    print(n)
```



Python: For Loop - Program Example

Create program that receive **n** customer name and age data, then print all of the customer data,
Where **n** is an integer.



Python: While Loop

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

```
while True:  
    # execute this
```

```
i = 1  
while i < 6:  
    print(i)  
    i += 1
```



Python: Continue

With the `continue` statement we can stop the current iteration, and continue with the next

```
for i in range(5):  
    if i == 3:  
        continue  
    print(n)
```



Python: Break

With the `break` statement we can stop the loop even if the while condition is true.

```
for i in range(5):  
    if i == 3:  
        break  
    print(n)
```




Python: Nested Loop

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop".

```
for i in range(2):  
    for j in range(3):  
        print("i:{},j:{}".format(i,j) ,end=" ")  
    print()
```