



# Simple Programming

## Lesson 4: For and While Loops (Part 1)

<http://www.yahmad.co.uk/>

# Lesson Overview

## Objectives

Understand the use of for loops to iterate through a list.

Understand how the while loop works with different conditions.

Understand the use of variables in loop.

## Outcomes

**For Loops Tasks**

**While Loops Tasks**

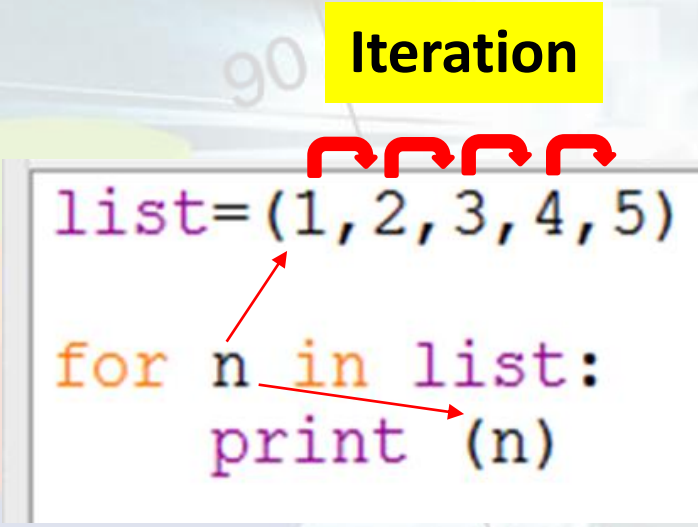
**Loops Extension**

# Loops Overview

## For Loop

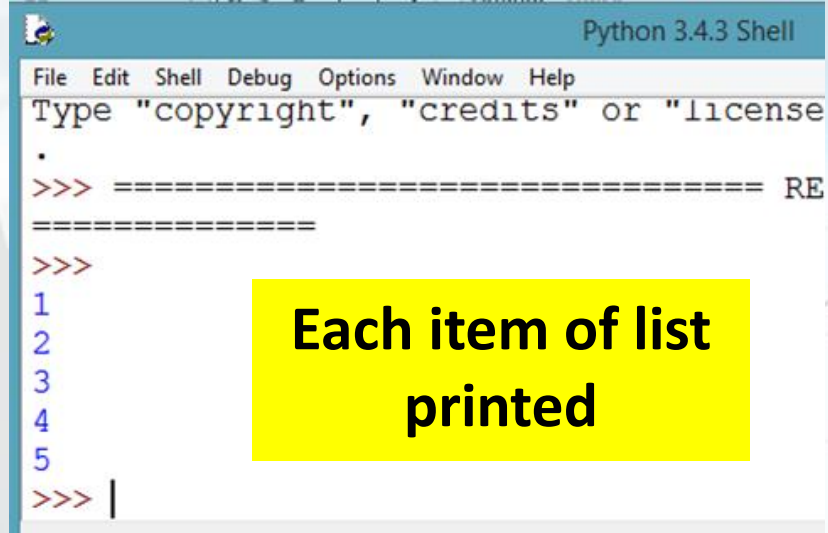
The **For loop** will **iterate** through every **item** of the **list**. The loop will **end** once it has gone **through** every item on the list.

### Iteration



The diagram shows a code block with the following text: `list=(1, 2, 3, 4, 5)` and `for n in list:` followed by `print (n)`. Above the list definition, four red curved arrows point to the right, indicating the sequence of iterations. A red arrow points from the variable `n` in the loop to the list definition, and another red arrow points from `n` to the `print (n)` statement.

```
list=(1, 2, 3, 4, 5)
for n in list:
    print (n)
```



The screenshot shows a terminal window titled "Python 3.4.3 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The prompt is `Type "copyright", "credits" or "license"`. The user has entered `>>>` and the shell has responded with `===== RE` and `=====`. The user has entered `>>>` and the shell has responded with `1`, `2`, `3`, `4`, and `5` on separate lines. The user has entered `>>>` and the shell has responded with a vertical bar `|`.

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license"
.
>>> ===== RE
=====
>>>
1
2
3
4
5
>>> |
```

Each item of list  
printed

**n** is a **variable** which will **store** the values of the list.

Each **cycle** of the loop will **print** a number stored in the **variable**.

# Loops Overview

## While Loop

The **While Loop** is based on conditions. If the condition is **True** then the **while Loop** will continue to **loop**.

If attempt == False  
Loop will continue

```
attempt="false"
while attempt=="false":
    try:
        year = int(input("Enter your year group (7,8,9): "))
    except:
        print ("You did not enter a valid year group.")
    else:
        if year == 7:
            print ("You are in Year "+str(year))
            print ("Your Head of Year is Mr Brown.")
            attempt="true"
```

If attempt is **not equal to False** then  
the **loop will stop**

If the **correct year is selected** then  
**attempt variable will change to True.**

# Task 1: Iteration using For Loops

1. Create a **list** with at least 5 numbers.
2. Create a **for loop** which will **print each item of the list individually**.

```
list=(1,2,3,4,5)

for n in list:
    print (n)
```

```
>>>
1
2
3
4
5
>>> |
```

**n** is the **name of the variable** which will **hold values of the list** through **each cycle** of the loop. This variable can be given any name.

3. Create a **for loop** which will **print each letter of a text string individually**.

```
File Edit Format Run Options Window |
for n in "Hello World":
    print (n)
```

```
>>>
H
e
l
l
o
W
o
r
l
d
>>> |
```

## Extension:

- Create a **for loop** which will **print the names of four students**.  
Create a **for loop** which will **print every day of the week**.

## Task 2: For Loop to Highest and Lowest Values

1. Create a list containing 8 numbers between 0-100.
2. Set the **highest variable to 0** and the **lowest to 100**.
3. Create a **for loop** which will set the **highest and lowest values into variables**.

n	high
34	0
56	34
2	56
12	56
7	56
98	56
45	98
35	98



```
2.loopsmax.py - C:/Users/
File Edit Format Run Options Window Help
list=[34,56,2,12,7,98,45,35]
high=0
low=100

for n in list:
    if n > high:
        high=n
    if n < low:
        low=n

print (high)
print (low)
```

```
File Edit Shell Deb
Python 3.4.3
, 22:43:06)
32
Type "copyri
more informa
>>> =====
=====
>>>
98
2
>>>
```

## Task 3: Multiples of 5

1. Create a **for loop** which will find the multiples of 5 from the list.
2. If the **number in the list can be divided by 5 with no remainder** then it is a **multiple of 5**.

```
list=[34,56,2,12,7,98,45,35]
```

```
for n in list:  
    if n % 5 ==0:  
        print (str(n)+" is a multiple of 5.")  
    else:  
        print (str(n)+" is not a multiple of 5.")
```

If the **remainder is equal to 0** then the **number in the list is a multiple of 5**.

```
>>> 52/5  
10.4  
>>> 52%10  
2
```

```
>>> =====  
>>>  
34 is not a multiple of 5.  
56 is not a multiple of 5.  
2 is not a multiple of 5.  
12 is not a multiple of 5.  
7 is not a multiple of 5.  
98 is not a multiple of 5.  
45 is a multiple of 5.  
35 is a multiple of 5.  
>>>
```

/

Using the slash (/) will work out the divided value – **10.4**

%

Using the slash (%) will work out the remainder – **2**

## Task 4: Quiz Menu

1. Create a **for loop** which will **add** and **count** the **values of the list**.
2. Create a **print statement** to print the **total numbers**, **sum** and the **average** for the list.

```
list=[34,56,2,12,7,98,45,35]
x=0
count=0

for n in list:
    x=x+n
    count=count+1

print ("Numbers in the list: "+str(count))
print ("The total sum of the list: "+str(x))
print ("Average: "+str(x/count))
```

```
>>>
Numbers in the list: 8
The total sum of the list: 289
Average: 36.125
>>> |
```

**X = 0**

**X=0+34 = 34**

**X=34+56 = 90**

**X=90+2 = 92**

**+n every loop  
cycle**

The value in **n** will be added to **X** in every cycle of the loop.

**Count = 0**

**Count=0+1 = 1**

**Count=1+1 = 2**

**Count=2+1 = 3**

**+1 every loop  
cycle**



## Task 5: Students Scores

1. Create the list which contains **3 students** and **two sets of marks**.
2. Create **variables** in the **for loop** which will hold the three values of the tuple.
3. Print the student Name, Maths and ICT score.

a

b

c

```
student = [("Bob", 56, 74), ("Billy", 98, 45), ("James", 17, 65)]

for a,b,c in student:
    print ("Student Name: "+a)
    print ("Maths Score: "+str(b))
    print ("ICT Score Score: "+str(c))
```

```
>>>
Student Name: Bob
Maths Score: 56
ICT Score Score: 74
Student Name: Billy
Maths Score: 98
ICT Score Score: 45
Student Name: James
Maths Score: 17
ICT Score Score: 65
>>> |
```

**Lists:** Is a sequence of Python Objects which **can be changed**

= [ 1 , 2 , 3 , 4 ]

**Tuples:** Is a sequence of Python Objects which **not be changed**

= [ ( ), ( ), ( ) ]

## Task 6: Highest Achiever

1. Total the marks for Maths and ICT exams into a variable called **t**.
2. Use an **if** to work out if the **total mark is greater than the highest**.
3. Set the **variables** for **high** and the **name**.

```
student = [("Bob", 56, 74), ("Billy", 98, 45), ("James", 17, 65)]
high=0
for a,b,c in student:
    t=b+c
    if t > high:
        high=t
        name=a
    print ("Student Name: "+a)
    print ("Maths Score: "+str(b))
    print ("ICT Score Score: "+str(c))
    print ("Total Score: "+str(t), "Average Score: "+str(t/2))
    print ("-----")

print("The highest total was acheived by: "+name)
print("The highest total was: "+str(high))
```

## Task 6: Highest Achiever Output

```
>>>
Student Name: Bob
Maths Score: 56
ICT Score Score: 74
Total Score: 130 Average Score: 65.0
-----
Student Name: Billy
Maths Score: 98
ICT Score Score: 45
Total Score: 143 Average Score: 71.5
-----
Student Name: James
Maths Score: 17
ICT Score Score: 65
Total Score: 82 Average Score: 41.0
-----
The highest total was acheived by: Billy
The highest total was: 143
>>>
```

**Highest Score**

The **highest score** and the **student name** has been printed.

# Task 7: Year Group (Try and Except)

1. Edit your year group task from the previous lesson.
2. The loop will **run while attempt** is set to **false**. The **attempt will change to true** once the **correct option has been selected**.
3. Enter **try** and **except** to catch any errors with the program.

```
attempt="false"
while attempt=="false":
    try:
        year = int(input("Enter your year group (7,8,9): "))

    except:
        print ("You did not enter a valid year group.")

    if year == 7:
        print ("You are in Year "+str(year))
        print ("Your Head of Year is Mr Brown.")
        attempt="true"

    elif year == 8:
        print ("You are in Year "+str(year))
        print ("Your head of year is Ms Jones")
        attempt="true"

    elif year == 9:
        print ("You are in Year "+str(year))
        print ("Your head of year is Mr Ahmad")
        attempt="true"

    else:
        print ("You did not enter 7, 8, 9")
```

Loop will continue to loop  
while attempt = false

Attempt is set to True

Attempt is set to True

Attempt is set to True

```
Enter your year group (7,8,9): 11
You did not enter 7, 8, 9
Enter your year group (7,8,9): a
You did not enter a valid year group.
You did not enter 7, 8, 9
Enter your year group (7,8,9): |
```

Using **try** and **except** will catch any errors and continue the loop.

# Task 8: Try and Except

1. Enter **try** and **expect** to catch any errors with the program.
2. If the user tries **typing in a letter** then **the except print** will be shown.

```
num="false"
pin=0
while num=="false":
    try:
        pin=int(input("Enter the four digit pin: "))

    except:
        print ("You can only type numbers.")

    if pin !=1234:
        print ("You have entered the incorrect pin.")

    else:
        print ("You have entered the correct password.")
        num="true"
```

```
Enter the four digit pin: 2
You have entered the incorrect pin.
Enter the four digit pin: aaaa
You can only type numbers.
You have entered the incorrect pin.
Enter the four digit pin:
```

Without the **except** then you will get an error message.

```
Enter the four digit pin: aa
Traceback (most recent call last):
  File "C:/Users/yahmad/Documents/Python/Lesson 4/8. while loop password - pt1a.py", line 5, in <module>
    pin=int(input("Enter the four digit pin: "))
ValueError: invalid literal for int() with base 10: 'aa'
>>> |
```

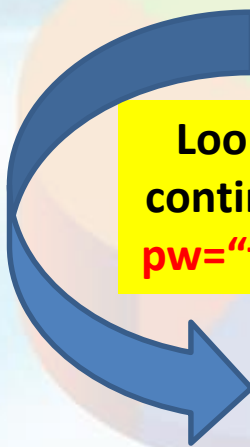
## Task 9: While Loop Password Part 1

1. Create a **while loop** which will continue Loop until the correct password has been entered..
2. Declare the **password in a variable**.
3. Declare the **pw** variable as **False**. When the password has been entered correctly then the variable will change to **True** and **break the loop condition**.

```
password="apple"  
pw="false"
```

Loop to continue if **pw is false**

```
while pw=="false":  
    login=(input("Enter the password to login: "))  
  
    if login != password:  
        print ("You have entered the incorrect password.")  
  
    else:  
        print ("You have entered the correct password.")  
        pw="true"
```



Loop to  
continue if  
**pw="false"**

Loop will stop if **pw** is set to **True**

## Task 9: While Loop Password Pt2 - Attempts

1. Create a **while loop** which will allow the user **3 attempts** to enter a password.
2. Declare the **password in a variable**.
3. Declare the **pw** variable as **False**. When the password has been entered correctly then the variable will change to **True** and **break the loop condition**.

0, 1, 2

```
attempt=0  
password="apple"  
pw="false"
```

Attempts is set to **0**.  
**1** will be **added** in each cycle

<3

Both conditions need to be **true**

Loop to continue if **pw is false**

Loop to  
continue if  
attempts is  
less than **3**

```
while attempt <3 and pw=="false":  
    login=(input("Enter the password to login: "))  
  
    if login != password:  
        print ("You have entered the incorrect password.")  
        attempt=attempt+1 +1  
        print ("This is your "+str(attempt)+" attempt")  
        if attempt ==3:  
            print ("You have been blocked")  
  
    else:  
        print ("You have entered the correct password.")  
        pw="true"
```

Loop will stop if **pw** is set to **True**

## Extension

**Create your own program using a while loop. Include suitable variables and user input.**





# Plenary – Refer to the Lesson Objectives

## Objectives

Understand the use of for loops to iterate through a list.

Understand how the while loop works with different conditions.

Understand the use of variables in loop.

## Plenary Task (Q&A)

Peer assess each other scripts.

Discuss the levels pupils have achieved for this task.

**Question:** What is the purpose of If and Nested Statements?