

LAB # 05

ITERATION WITH LOOPS

Objective

To get familiar with the different types of loops.

Exercise

- A. Point out error, if any, and paste the output also in the following Python programs.

Code 01:

```
Task1.py
1 for(;;)
2 {
3     printf("SSUET")
4 }
5

Shell
>>> %Run Task1.py
Traceback (most recent call last):
File "C:\Users\BINARY\OneDrive\Desktop\New folder (2)\Task1.py", line 1
  for(;;)
          ^
SyntaxError: invalid syntax
>>>
```

Assistant

SyntaxError: invalid syntax
Task1.py, line 1
Python doesn't know how to read your program.
Small ^ in the original error message shows where it gave up, but the actual mistake can be before this.
Did you forget the colon?

Warnings

May help you find the cause of the error.
Task1.py
Line 1 : invalid syntax [syntax]
Was it helpful or confusing?
General advice on dealing with errors.

The error is we always give the range in for loop. We use indentation instead of curly braces after the condition end, we use colon too. Third we use f-string in the brackets not outside. After error solve ↓.

Output:

```
Task1.py
1 for i in range(1,11):
2     print(f"SSUET", end = ("!! "))

Shell
>>> %Run Task1.py
SSUET!! SSUET!! SSUET!! SSUET!! SSUET!! SSUET!! SSUET!! SSUET!! SSUET!! SSUET!!
```

Assistant

Warnings

May be ignored if you are happy with your program.
Task1.py
Line 2 : Using an f-string that does not have any interpolated variables
Was it helpful or confusing?

Code 02:

The screenshot shows a Python code editor with a file named `Task1.py`. The code contains the following lines:

```
1 count = 4
2 while n < 8
3     count = count + 3
4     print(count)
```

In the `while` loop, there is a missing colon after `n < 8`. The code editor highlights this error with red text and a small red arrow pointing to the missing colon. To the right of the code, the `Assistant` panel displays the following error message:

SyntaxError: expected ':'
Task1.py, line 2
Python doesn't know how to read your program.
Small ^ in the original error message shows where it gave up, but the actual mistake can be before this.
Did you forget the colon?

The `Warnings` section also shows a note about the missing colon:

File "C:\Users\BINARY\OneDrive\Desktop\New folder (2)\Task1.py", line 2
while n < 8
^
SyntaxError: expected ':'
Task1.py
Line 2 : expected ':' [syntax]

At the bottom, there is a link to `Was it helpful or confusing?` and `General advice on`.

There is a Syntax Error because `n` variable is not defined. We use `count` instead of `n`

After the condition we use colon. We don't use colon in increment condition. In `print` condition if we use extra indentation, it gives us the error. After error solve ↓.

Output:

The screenshot shows the same `Task1.py` file with the following corrected code:

```
1 count = 4
2 while count < 8:
3     count = count + 3
4     print(count)
```

The code runs successfully in the `Shell` pane, displaying the output:

```
7
10
```

To the right, the `Assistant` panel provides feedback:

The code in `Task1.py` looks good.
If it is not working as it should, then consider using some general debugging techniques.
Was it helpful or confusing?

Code 03:

The screenshot shows a Python code editor with a file named `Task1.py`. The code contains the following lines:

```
1 for v in range(4:8)
2     print(v)
```

The range function is used with invalid parameters: `range(4:8)`. The code editor highlights this error with red text and a small red arrow pointing to the colon. To the right of the code, the `Assistant` panel displays the following error message:

SyntaxError: invalid syntax
Task1.py, line 1
Python doesn't know how to read your program.
Small ^ in the original error message shows where it gave up, but the actual mistake can be before this.
Did you forget the colon?

The `Warnings` section also shows a note about the invalid range:

File "C:\Users\BINARY\OneDrive\Desktop\New folder (2)\Task1.py", line 1
for v in range(4:8)
^
SyntaxError: invalid syntax
Task1.py

There is a syntax error. While we give start & end range we use comma instead of colon. After the condition end we always use colon. Here is no indentation in print condition so this is error. After error solve ↓.

Output:

The screenshot shows a Python code editor interface. On the left, the code file 'Task1.py' contains the following code:

```
1 for v in range(4,8):
2     print(v)
```

In the 'Shell' tab, the command '%Run Task1.py' is run, resulting in the output:

```
4
5
6
7
```

To the right, there is an 'Assistant' panel with the following text:

The code in [Task1.py](#) looks good.
If it is not working as it should, then consider using some general debugging techniques.

[Was it helpful or confusing?](#)

B. What will be the output of the following programs:

Code 04:

The screenshot shows a Python code editor interface. On the left, the code file 'Task1.py' contains the following code:

```
1 i = 2
2 while i < 10:
3     if i % 2 == 0:
4         print(i)
5     i = i + 1
```

In the 'Shell' tab, the command '%Run Task1.py' is run, resulting in the output:

```
2
4
6
8
```

Output:

The screenshot shows a Python code editor interface. On the left, the code file 'Task1.py' contains the following code:

```
1 i = 2
2 while i < 10:
3     if i % 2 == 0:
4         print(i)
5     i = i + 1
```

In the 'Shell' tab, the command '%Run Task1.py' is run, resulting in the output:

```
2
4
6
8
```

Code 05:

The screenshot shows a code editor window titled "Task1.py". The code contains the following Python script:

```
i = 1
while i>0:
    print(i)
    i = i + 1
```

To the right of the code, there is an "Assistant" panel. It displays a "SyntaxError: invalid syntax" message with "Task1.py, line 4" highlighted. Below the message, it says "Python doesn't know how to read your program." and provides a note: "Small ^ in the original error message shows where it gave up, but the actual mistake can be before this."

This is Infinite loop.

Output:

The screenshot shows a terminal window titled "Shell". The output of the command "%Run Task1.py" is displayed, showing the following sequence of numbers:

```
1358
1359
1360
1361
1362
1363
1364
```

Code 06:

The screenshot shows a code editor window titled "Task1.py". The code contains the following Python script:

```
for v in range(3,9,2):
    print(v)
```

Output:

The screenshot shows a terminal window titled "Shell". The output of the command "%Run Task1.py" is displayed, showing the following sequence of numbers:

```
>>> %Run Task1.py
3
5
7
```

C. Write Python programs for the following.

1. Write a program that prints the first 10 natural numbers and their sum using ‘for loop’.

Code 07:

```
Task1.py
1 total = 0
2 print("The first 10 natural numbers are:")
3 for i in range(1, 11):
4     print(i, end=" ")
5     total += i
6 print("\nThe Sum is:", total)
```

Output:

```
Shell x
>>> %Run Task1.py
The first 10 natural numbers are:
1 2 3 4 5 6 7 8 9 10
The Sum is: 55
```

2. Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form.

Code 08 “Using for loop”:

```
Task1.py
1 print(f"\tTable")
2 Table = eval(input("Enter your Num: "))
3 for x in range(1,11):
4     print(f"{Table} x {x} = {Table*x}")
```

Output:

```
Shell
Table
Enter your Num: 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

Code 08 “Using while loop”:

```
Task1.py
1 print(f"\tTable")
2 Table = eval(input("Enter your Num: "))
3 start = 1
4 while start < 11:
5     print(f"{Table} x {start} = {Table*start}")
6     start = start + 1
```

Output:

```
Shell
Table
Enter your Num: 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

3. Write a program that take vowels character in a variable named “vowels” then print each vowels characters in newline using ‘for loop’.

Code 09:

```
Task1.py
1 vowels = "aeiou"
2 for v in vowels:
3     print(v)
4
```

Output:

```
Shell %
>>> %Run Task1.py
a
e
i
o
u
```