

SI Session 10/03

By Cody Ptacek

Activity: Syntax Review

Task

- Split up into groups (potentially groups of individuals)
- Each group will be assigned a topic from the following list:
 - for-in loops
 - Stand-alone if statements
 - If-else blocks
 - If-elif blocks
 - File IO read/write
 - File reading (using for-in loop)

Write the TEMPLATE for your assigned topic. Be prepared to explain how it works to the class.

While Loops

initialization

while looping_condition (using initialization):

code that runs while looping_condition is True

progression (of initialization)

Discussion: How to Use This Session

Given that we have a full week until your first test, here's how I would use this session.

- 1) Takes notes/pics of the activities and your outputs for future studying.
- 2) Record any topics that you struggled with. Spend additional time studying those topics.
 - a) If you can, record what was difficult for you.

NOTE:

This session, I'm largely asking all of you to try to do stuff on your own without working as a group. If you struggle to do these problems on your own, you may turn to a partner and ask for help. However, I DISCOURAGE you from doing so without first attempting it on your own.

YOUR TESTS ARE INDEPENDENT!

Activity: Interpreting File IO

Task

- Interpret the code on the following page with the provided txt file.
- Produce the output file produced by the code.
- Compare answers with a neighbor.
Discuss any differences with each other.

Activity: Interpreting File IO

example.txt:

This is

an

example file

for

use in this

SI Session!

```
input_file = open("example.txt", 'r')
output_file = open("output_file.txt", 'w')

input_list = []

for line in input_file:
    line = line.split()
    input_list += line

for element in input_list:
    element = element.lower()
    if element == "file":
        output_file.write("donut" + "\n")
    elif element == "SI":
        output_file.write("Practice" + " ")
    else:
        output_file.write(element + " ")

input_file.close()
output_file.close()
```

Actity: Interpreting File IO

this is an example donut

for use in this si session!

Activity: Different If Statements

Task

- On the following slides are four similar programs. The main difference is in the structures and types of if-statements.
- Read through the four different programs & provide the output for each.
 - Make sure you understand what makes these outputs different!

Activity: Different If Statements

```
# stand_alone_ifs.py
my_nums = range(0,11)
```

```
for num in my_nums:
    if num == 2:
        print("two")
    if num % 2 == 0:
        print("mod 2")
    if num == 7:
        print("3+4")
    print(num)
```

```
#if-else statement.py
my_nums = range(0,11)
```

```
for num in my_nums:
    if num == 2:
        print("two")
    else:
        print(False)
    if num % 2 == 0:
        print("mod 2")
    else:
        print(False)
    if num == 7:
        print("3+4")
    else:
        print(num)
```


Activity: Different If Statements

mod 2
0
1
two
mod 2
2
3
mod 2
4
5
mod 2
6
3+4
7
mod 2
8
9
mod 2
10

False
mod 2
0
False
False
1
two
mod 2
2
False
False
3
False
mod 2
4
False
False
5
False
mod 2

6
False
False
3+4
False
mod 2
8
False
False
9
False
mod 2
10

Activity: Different If Statements

```
# if-elif statement.py
```

```
my_nums = range(0, 11)
```

```
for num in my_nums:
```

```
    if num == 2:
```

```
        print("two")
```

```
    elif num % 2 == 0:
```

```
        print("mod 2")
```

```
    elif num == 7:
```

```
        print("3+4")
```

```
    print(num)
```

```
# if-elif-else statement.py
```

```
my_nums = range(0,11)
```

```
for num in my_nums:
```

```
    if num == 2:
```

```
        print("two")
```

```
    elif num % 2 == 0:
```

```
        print("mod 2")
```

```
    elif num == 7:
```

```
        print("3+4")
```

```
    else:
```

```
        print(num)
```

Activity: Different If Statements

mod 2
0
1
two
2
3
mod 2
4
5
mod 2
6
3+4
7
mod 2
8
9
mod 2
10

mod 2
1
two
3
mod 2
5
mod 2
3+4
mod 2
9
mod 2

Activity: Interpreting While Loops

```
my_string = "Let them eat cake!"
my_num = 0
modified_string = ""

while my_num < len(my_string):
    if my_num % 10 == 0:
        modified_string += "10"
    if my_num % 6 == 0:
        modified_string += "6"
    else:
        modified_string += my_string[my_num]
    my_num += 2
print(modified_string)
```

Activity: Interpreting While Loops

106tt6 10a6ae

Activity: Session Review

Task

- Write down the template for one of the topics you weren't assigned.
 - Write them all if you can.
- Answer the following questions:
 - What is the difference in output between using stand-alone if statements and if-elif blocks?
 - How to write to a file? Do you need to add anything else to the strings you are writing to a file?