

6: operators, if then

Thursday, January 13, 2022 9:37 PM

We're going to go over details which will really help you with Assignment 1

How to use input with multiple arguments

How to check if two values are the same

How to use If and Else

And a review on program arguments

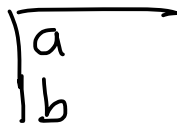
Assignment operator

Thursday, January 13, 2022 10:04 PM

$$a = \emptyset$$

$$a = a + 1$$

$$b = a + 2$$



A B C D what is b?

1 2 3 4

$$a = \emptyset$$

$$a = a + 1$$

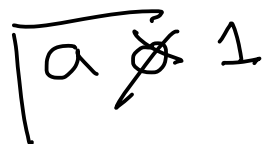
0 + 1
1

$$a \rightarrow \emptyset$$

still in RAM
flagged for overwrite
by garbage
collection

a ~~→~~ \emptyset
→ 1

for shorthand we'll write



a is not changed

$$b = a + 2$$

1 + 2
3

Order of Operations

Thursday, January 13, 2022 9:42 PM

Last time: Inside pavers 1st

What will this calc to?

1 min to
try by hand
use your
note sheet!

```
1 much_num = 5%(3**(6//4))
```

(1)

($3^1 \Rightarrow 3$)

5% 3

2

Parentheses are very helpful in getting
math to do what you expect
what about w/o pavers?

```
1 much_num = 5%(3**(6//4))
```

```
2 much_num2 = 5%3**6//4
```

Name	Value
much_n 2	
much_n 1	

$3^6 \Rightarrow 729$

$5 \% 729 \Rightarrow 5$

$5 // 4 \Rightarrow 1$

Wait! How to know?!

PEMDAS (kinda)

Parentheses

Exponents

Multiplication Division Modulo (left to right)

Addition Subtraction (Left to right)

Try

$$5 * 2 ** 1 // (3 \% 4)$$

$$\textcircled{3} \quad \textcircled{2} \quad \textcircled{4} \quad \textcircled{1} \Rightarrow 3$$

$$\Rightarrow 2$$

$$5 * 2 \Rightarrow 10$$

$$10 // 3 \Rightarrow 3$$

why do we care? Many programs have math, even if just 'simple addition' mult
So best know how to get the right answer!

Breakout room! 3 min

1 person share screen

7 2 51 11

Place operators so result is 7

$$(7^{**2})//(51\%11)$$

Operators

Tuesday, January 18, 2022 9:55 PM

< less than
> greater than
<= less than or equal to
>= greater than or equal to

```
print(3 < 4)
```

Evaluate to True or
False

What is
True or False?

Not an int, Not a string!

Pro Tip: to quickly check
short bits of code, use the
shell!

How do we check the type
of something?

```
>>> type(3<4)
```

```
<class 'bool'> a new type!
```

bool for boolean
Named after George Boole

Named after George Boole
True or False
Caps in python!

is-snowing = False

AI has us check to see
if the user has the right
answer, so booleans will
help

== Equal to



!= Not equal to

= Single is assignment

== Double is comparison

```
>>> 3 == 4
```

```
False
```

```
>>> 5 != 4
```

```
True
```

New operators; we need new
order of operations!

```
1 Parenthesis
2 Exponentiation (right to left)
3 Unary (like -5 )
4 Multiplication, division, modulus (right to left)
5 addition and subtraction
6 numerical comparisons == or !=
7 not
8 and
9 or
```

} Next class!

A B C D Breakout Rooms

```
1 4 == (3 + 1) > (False != (4 >= 5 ))|
```

A True

B False

C Error

D 4

Prompt is in the google drive breakout room prompts folder

3 min. Work it, don't put it in Thonny! If you get done early, check in with your classmates about how they're doing

If

Tuesday, January 18, 2022 10:30 PM

A1 requires us to check
If there are system arguments.
How?

```
if len(sys.argv) > 1:
    print("arguments")
```

True or False
colon
new line
Tab
4 spaces
one or more lines of code for when boolean expression is True

```
1 # Author: Scott Wehrwein
2 # Date: 10/9/2019
3 # Example demonstrating the if statement
4
5 # The behavior of the code below depends on the
6 # value of this variable:
7
8 is_raining = True #we're in the PNW, it's always raining
9
10 # Make a raincoat recommendation based on raininess:
11 if is_raining:
12     # indented code block: one or more statements to be
13     # executed only if the condition (is_raining) evaluates
14     # to True:
15     print("Wear a raincoat")
16     print("Have a great day!")
17
18
19 # Statements after the indented code block are
20 # executed afterwards as usual
21
22 # Wish the user well
23 print("Have a great day!")
```

Else

Tuesday, January 18, 2022 10:35 PM

```
1 # Author: Caroline Hardin, modified from code by Scott Wehrwein
2 # Date: 1/26/2020
3 # Example demonstrating the if/else statement
4
5 is_raining = input("Is it raining? yes or no ")
6 is_raining = is_raining == "yes" #caps could be a problem! more later|
7
8 print("Welcome to the Bel-Weather app")
9
10 # Recommend whether to wear a raincoat or not.
11 if is_raining:
12     print("You should wear a raincoat!")
13 else:
14     print("No raincoat required")
15
16 # Wish the user well
17 print("Thanks for using Bel-Weather!")
```

Program arguments revised

Tuesday, January 18, 2022 10:37 PM

```
#get the two numbers.  
#temp: get it from input TODO: fix to get from program arguments  
num1 = input("what is num1 ")  
num2 = input("what is num2 ")  
  
#print out what is num1 plus num2 ?  
  
#put in a different file to test out  
test = "what do we get if we add " + int(num1) + " to " + num2 + " ? "  
  
answer1 = input("test")
```

Spot the bugs!

(run after each fix)

So annoying to keep typing in
num1 and num2!
why can't we do this?

```
4 num1 = 4#:  
5 num2 = 5#:
```

hard coding your user doesn't
want to change your code to use
your program. Might not even be
able to!

```
5 import sys  
6 num1 = sys.argv[1]#:  
7 num2 = sys.argv[2]#:  
8
```

cd C:\Users\hardinc3.CSCI\OneDrive - Western Washington University

Cd desktop
python3.10 testingPython.py

```
C:\Users\hardinc3.CSCI\OneDrive - Western Washington University\Desktop>python3.10 testingPython.py 5 6  
what do we get if we add 5 to 6 ?
```

Quick up arrays and change arguments
This will Save you time
We're doing this so your Hw takes
less time
Learn it and use it!

Using input with multiple arguments

Tuesday, January 18, 2022 9:33 PM

A1 asks you to work with
Strings and inputs.

```
>>> %Run A1_modulo_soln_s20.py 2 20
What do you think 2 modulo 20 is? 2
Good job!
2 divided by 20 is 0 remainder 2

What do you think 20 modulo 2 is? 5
oops
20 divided by 2 is 10 remainder 0

your score is: 1
```

how?

Step 1: pseudocode

```
2 #get the two numbers.
3 #temp: get it from input TODO: fix to get from program arguments
4 num1 = input("what is num1")
5 num2 = input("what is num2")
6
7 #print out what is num1 plus num2 ?
8 answer1 = input("what is", num1, "plus", num2)
9 #what will happen when we run this?
```

Yes an error!

```
what is num14
what is num25
Traceback (most recent call last):
  File "C:\Users\hardinc3.CSCI\OneDrive - Western Washington U
8, in <module>
    answer1 = input("what is", num1, "plus", num2)
TypeError: input expected at most 1 arguments, got 4
```

Input only takes 1 argument
You can't use multiple arguments

input only takes 1 argument
You can only pass it 1 thing
2 ways to fix this

1) Use print

```
8 print("what is", num1, "plus", num2)
9 answer1 = input("?")
```

but that's on 2 lines. How
Not end the print with a newline
but a space?

Type in the chat but don't push
enter yet.... 1 min

```
8 print("what is", num1, "plus", num2, end=" ")
```

Way #2) Make a new String
with everything so we can give
input a single value

Strategy: use a scratch pad python
file to test syntax clear of the
test of your (possibly buggy) code

```
1 #put in a different file to test out some syntax
2 num2 = input("what is num 2")
3 test = "what happens if we add" + 2 + "a variable" + num2
```

What will happen if we run this?

what will happen if we run this?
error!

How do we fix it
`str()`

```
3 test = "what happens if we add" + str(2) + "a variable" + num2
```

You can add strings together to get a
bigger string!
which you could then pass to input
function

```
4 answer1 = input(test) works!
```

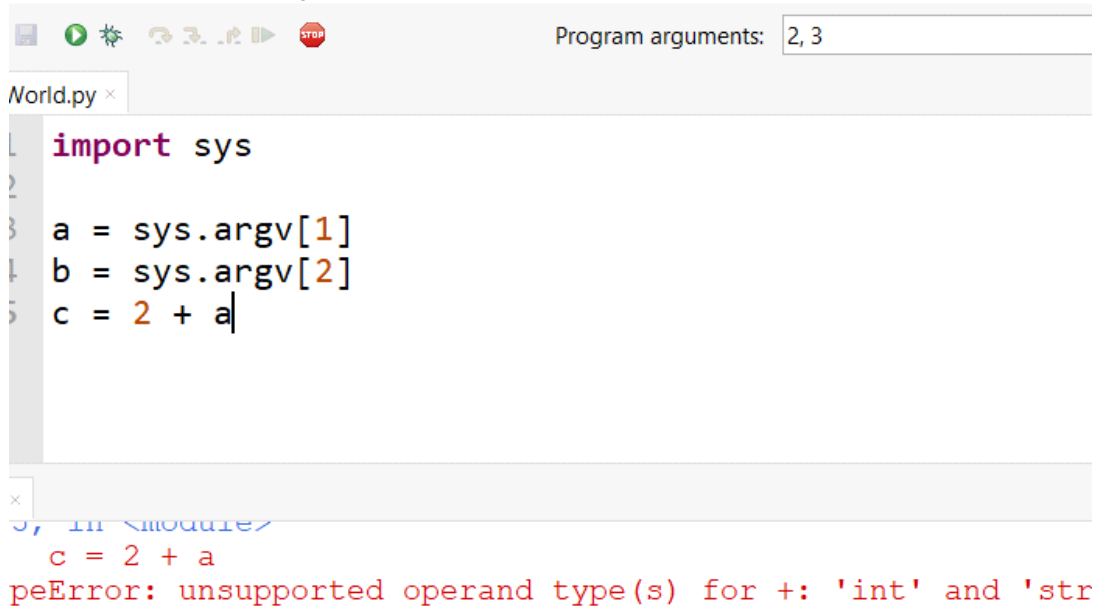
check the variable answer1

Summary:

- 2 ways to Compose a long input prompt
- 1) use `print` and change the end
 - 2) make a new string

Exercises: sys args, print, //, %

Thursday, January 13, 2022 9:37 PM



The screenshot shows a Python IDE window titled "World.py". The code in the editor is as follows:

```
1 import sys
2
3 a = sys.argv[1]
4 b = sys.argv[2]
5 c = 2 + a
```

The "Program arguments:" field at the top right contains the values "2, 3". Below the code editor, a console window displays the following error message:

```
c = 2 + a
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Code Execution. For each fragment of code, write into the box on the right what the program will print to the screen. Write into the box ONLY what the program will print. If the program will not print anything, write the word NOTHING in the box. If the code will generate an error, write ERROR.

1. `Print("Hello World")`

2. `print("Hello World")`

3. `print(int(str("43")))`

4. `day=12
year="Saturday"
print("mon", year, sep="Day")`

5. `firstNum = 1
secondNum = 4
print(firstNum // secondNum)
print(secondNum / firstNum)
print(firstNum % secondNum)
print(secondNum % firstNum)`