Intro to Coding with Python_main()

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (<u>https://jcrouser.github.io/</u>)

Plan for Today

•Themain() function

A reminder from the syllabus

• "References"

- You should use resources when you need help!
- And you must cite them! (Give them credit for helping you)
- In-line citations to any resources you used, including page numbers (if a printed resource) or a direct URL (if an online resource).

• Ex.

Example

```
*documentations.py - /Users/jcrouser/Google Drive/Teaching/Course Material/CSC111/CSC111/demos/documentations....
#
         Names: Jordan Crouser & Morganne Crouser
#
          Date: 26 September 2018
#
     Filename: demo.py
  Description: This is a demonstration of how to
#
#
                 propertly attribute help on a
#
                CSC111 assignment
name = input("Enter your name: ")
formatted_string = "{0:>10}".format(name)
print(formatted_string)
# REFERENCES
# I googled how to use the str.format(...) method
# and found the Python documentation here:
# https://docs.python.org/3/library/stdtypes.html#str.format
                                                          Ln: 17 Col: 60
```

Recap

• So far, we've been writing code in files as if we were writing it on the console:



• When we do this, the Python interpreter executes everything from the **top down**

An alternative

• It is better practice to write the code you want to execute inside a main () function, e.g.



• This lets the interpreter "read ahead" and then execute

• **Remember**: the interpreter reads from the top down, which means that it reads the **definition** first



 Then it reads each line inside the definition, but these don't get executed yet



• At this stage, we've given python a "recipe" for what we want it to do when we call **main()**



• If we stop here, **nothing will actually happen**

The real work happens only when we actually call the main () function



• When we do, python goes to the **main()** function definition and follows the instructions it finds there

Discussion

Why bother?

Just one more thing...

🗅 🗙 mai	n.py* × salutations.py*
1	#!/usr/bin/env python3
2	# -*- coding: utf-8 -*-
3	ппп
4	Ab Mosca
5	02.20.2024
6	
7	Importing functions demo
8	<i>nnn</i>
9	
10	det nello():
	print("Hello!")
12 12	dof goodbyo()
13 17	print("Good bye!")
15	
16	<pre>def main():</pre>
17	<pre>print("salutations main() running")</pre>
18	hello()
19	goodbye()
20	
21	main()
22	

- Suppose I have code I wrote elsewhere
- What happens if someday we want to use the code in this file as **part of another program**?

Just one more thing...

C I	ans me as part of another program.		
		🗅 🗙 ma	in.py* × salutations.py
		2	# -*- coding: utf-8 -*-
		 ∕I	Ab Masca
		- 1 5	AD 1103Ca 02_20_2024
	02.20.2024	6	0212012024
		7	Main function and importing notes
		8	""
		9	
		10	import salutations
		11	
		12	<pre>def talkToUser(start):</pre>
		13	it start:
		14	<pre>print("welcome to my program!") colutations halls()</pre>
		15 16	
		17	salutations goodbye()
		18	Sacacacións goodbyc()
	hello()	19	<pre>def main():</pre>
	goodbye()	20	talkToUser(1)
		21	
	<pre>main()</pre>	22	main()
		23	

• What happens if someday we want to use the code in this file as **part of another program**?

Just one more thing...

this file as **part of another program**? × main.py* × salutations.py #!/usr/bin/env pyt # -*- coding: utf-8 -*-2 3 Ab Mosca 5 02.20.2024 Main function and importing notes 8 9 import salutations 10 11 12 def talkToUser(start): 13 if start: print("Welcome to my program! ") 14 salutations.hello() print("Good by 15 16 else: salutations.goodbye() 17 print("salutat 18 19 def main(): 20 talkToUser(1) 21 22 main() 23

• What happens if someday we want to use the code in

Just one more thing...

<pre>fmain.py* X salutations.py #!/usr/bin/env pyt # -*- coding: utf-8 -*- 3</pre>			
<pre>#!/usr/bin/env pyt # -*- coding: utf-8 -*-</pre>	main.py* × salutations.py*	🗅 🗙 mai	n.py* × salutations.py
# -*- Cooling: dtr-3# -*Ab MoscaAb Mosca502.20.202467Main function and importing notes""""""9import salutations10def hello():11print("Hello!"12def goodbye():14print("Good by1516print("salutat18hello()19def main():10111213141515161718191910101112131415151617171819101011121314151516171718191910111112131415151617171819191910101112131415151617181919 <th><pre>#!/usr/bin/env pyt # up coding: utf</pre></th> <th>2</th> <th># -*- coding: utf-8 -*-</th>	<pre>#!/usr/bin/env pyt # up coding: utf</pre>	2	# -*- coding: utf-8 -*-
Ab Mosca4Ab Mosca02.20.2024502.20.202417Main function and importing notes1110110111112112112112112112112112112112112112112113114151516151715181619161917101810191111121113121414151516161717181719171018111911101211131114121512161317141816191719181919191010101110121013111412151416151715181619161916<	# -*- couing: uti-	<u></u> כ ⊿	
Ab hosed302.20.202402.20.20246Importing function810import salutationsdef hello():11print("Hello!"12def goodbye():14print("Good by15salutations.hello()16ef main():17print("salutathello()19goodbye()20talkToUser(1)main()23	Ab Mosca	4 5	AD MOSCA A2 20 2021
Importing function7Main function and importing notesImporting function8"""9import salutationsdef hello():11print("Hello!"12def talkToUser(start):if start:def goodbye():14print("Good byprint("Good by15salutations.hello()else:def main():print("salutathello()goodbye()20talkToUser(1)21main()23	02 20 202 <u>4</u>	5	02.20.2024
<pre>Importing function Importing function Import and Importing functs Import and Importing functs Import and Import and Importing functs Import all Import and Import a</pre>	0212012024	7	Main function and importing notes
<pre> """ 9 import salutations def hello(): print("Hello!" 12 def talkToUser(start): if start: def goodbye(): print("Good by 15 salutations.hello() else: def main(): print("salutat hello() goodbye() 20 talkToUser(1) main() 22 main() 23 main() main() 23 main() 23 main() main()</pre>	Importing function	8	
<pre>def hello(): print("Hello!" def goodbye(): print("Good by def main(): print("salutat hello() goodbye() main() main() 23</pre> import salutations def talkToUser(start): if start: print("Welcome to my program!") salutations.hello() else: salutations.goodbye() def main(): rint("salutations.goodbye() def main(): rint() zalutations.goodbye() def main(): rint() zalutations.goodbye() rint() zalutations.goodbye() rint() zalutations.goodbye() rint() zalutations.goodbye() rint() zalutations.goodbye() rint() zalutations.goodbye() rint() zalutations.goodbye() zalutations.good		9	
<pre>def hello(): print("Hello!" 11 def talkToUser(start): if start: def goodbye(): print("Good by 15 salutations.hello() 16 else: def main(): print("salutat hello() goodbye() 20 talkToUser(1) main() 23 </pre>		10	import salutations
<pre>print("Hello!" 12 def talkToUser(start): 13 if start: print("Welcome to my program! ") salutations.hello() 16 else: def main(): print("salutat hello() goodbye() 20 def main(): talkToUser(1) 21 main() 23</pre>	<pre>def hello():</pre>	11	
<pre>13 if start: print("Welcome to my program!") salutations.hello() else: def main(): print("salutat hello() goodbye() 20</pre>	<pre>print("Hello!"</pre>	12	<pre>def talkToUser(start):</pre>
<pre>def goodbye(): 14 print("Welcome to my program!") salutations.hello() l6 else: def main(): 17 salutat hello() goodbye() 20 def main(): talkToUser(1) main() 22 main() 23</pre>		13	if start:
<pre>print("Good by 15 salutations.hello() 16 else: def main(): 17 salutations.goodbye() print("salutat 18 hello() 19 def main(): goodbye() 20 talkToUser(1) 21 main() 22 main() 23</pre>	<pre>def goodbye():</pre>	14	<pre>print("Welcome to my program! ")</pre>
<pre>def main(): print("salutat hello() goodbye() main() </pre>	print(<i>"Good by</i>	15	salutations.hello()
<pre>def main(): 17 salutations.goodbye() print("salutat hello() goodbye() 20 talkToUser(1) 21 main() 23</pre>		16	else:
<pre>print("salutat 18 hello() goodbye() 20 talkToUser(1) 21 main() 22 main() 23</pre>	def main():	17	salutations.goodbye()
<pre>metto() goodbye() 20 talkToUser(1) 21 main() 22 main() 23 </pre>	print("salutat	18	
main() 22 main() 23	nello()	19	der main():
main() 22 main()	goodbye()	20	
	main()	21	main()
		23	

What will happen if I run main.py?

What will happen if I run main.py?

/Use	rs/abmosca/Documents/Documents/Westfield/classes/CAIS117-S24/website/demos/main.py	Source Console Object A a =
	× main.py × salutations.py	
	2 # -*- coding: utf-8 -*-	Usage
	4 Ab Mosca 5 02.20.2024 6	Here you can get help of any Help Variable Explorer Plots Files
	7 <i>Main function and importing notes</i> 8 """ 9	□ × Console 2/A □ = 14.0.6]
	10 import salutations 11	Type "copyright", "credits" or "license" for more information.
	<pre>def talkToUser(start): if start: print("Welcome to my program! ") salutations.hello() else: salutations.goodbye() 18 10 dof main():</pre>	<pre>IPython 8.15.0 An enhanced Interactive Python. In [1]: runfile('/Users/abmosca/Documents/Documents/ Westfield/classes/CAIS117-S24/website/demos/main.py', wdir='/Users/abmosca/Documents/Documents/Westfield/ classes/CAIS117-S24/website/demos') salutations main() running</pre>
	<pre>19 def main(): 20 talkToUser(1) 21</pre>	Hello! Good bye!
	22 main() 23 24	Hello!

Ju th

Discussion

 What we need: a way to tell python to behave one way when we run it as a "stand-alone" program, and a different way when we import it

Ideas?

Python convention

We can use an if statement to tell python to call the main() function only if the program is being run directly

tly	🗅 🗙 ma	in.py × salutations.py*
	1	<pre>#!/usr/bin/env python3</pre>
	2	# -*- coding: utf-8 -*-
	3	
	4	Ab Mosca
	5	02.20.2024
	6	Transaction from the second second
	/	Importing functions demo
	ð o	
	10	dof bollo()
	10 11	<pre>def net(0(): print("Hollo("))</pre>
	12	
	13	def goodbye().
	14	print("Good bye!")
	15	
	16	<pre>def main():</pre>
	17	<pre>print("salutations main() running")</pre>
	18	hello()
	19	goodbye()
	20	
	21	ifname == "main":
	22	main()
	23	

Python convention

This is a little bit confusing: we named the function we created to hold our program main()

🗅 🗙 ma	in.py × salutations.py*
	<pre>#!/usr/bin/env python3 # vk codings utf 8 vk</pre>
	# -*- Courny: uti-o -*-
	Ab Mosca
	02.20.2024
	Importing functions demo
8	<i>n n n</i>
	det hello():
	print("Hello!")
	dof goodbyo()
	print("Good bye!")
16	<pre>def main():</pre>
17	<pre>print("salutations main() running")</pre>
	hello()
	goodbye()
21	ifname == "main":
22	main()
22	

Python convention

In our if statement, we're asking whether some variable called _____name____ is equal to the string "____main___" (not to mention I don't recall initializing anything called _____name___...)

🗅 🗙 ma	in.py × salutations.py*
	#!/usr/bin/env python3
	# -*- coding: utf-8 -*-
	Ab Mosca
	02.20.2024
	Importing functions demo
	det hello():
	print("Hello!")
	<pre>def goodbye(): print("Cood byo!")</pre>
	print("Good bye!")
	dof main().
	print("salutations main() running")
	hello()
20	goodbye()
20	if name == " main ":
22	main()
23	

To the documentation!





15-minute exercise

 Write a program that contains a main() function, which contains instructions for printing out the phrase:

"Today is not Friday :-("

Use an if statement combined with checking the value of the <u>name</u> variable to call main() only when the program is run directly

• Add an **else** statement so that whenever the program ("module") is **import**ed, it prints out the phrase:

"Maybe today...?"

Discussion

What did you come up with?

Takeaways

- Programs ("modules") that are well-organized are easier to read, more versatile, and potentially more efficient
- The first step we'll take toward organizing our code is to include a main() function, which includes the instructions we want our program to run
- To make it easier to **import** code we write now into later modules, we will follow the convention of including:

at the end of each module

Helpful tip: have a starter template

```
Interview of the starter.py - /Users/jcrouser/Google Drive/Teaching/Course Material/CSC111/CSC111/demos/start...
#-
#
         Names: Jordan Crouser & <PARTNER>
#
          Date: <DATE>
#
     Filename: starter.py
# Description: This is a demonstration of how to
#
                 organize your starter code (incl.
                 a main function and scope check)
#
         _____
def main():
    # This is where my code will go
if __name__ == "__main__":
    main()
# REFERENCES
                                                 Ln: 3 Col: 21
```