IRIS 2022 Seismology Skill Building Workshop OSL

Quiz navigation

1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
31

Finish review

Started on	Saturday, August 27, 2022, 6:23 AM
State	Finished
Completed on	Sunday, September 4, 2022, 6:14 AM
Time taken	7 days 23 hours
Marks	26.80/31.00
Grade	**86.45** out of 100.00

Question 1
Correct
1.00 points out of 1.00
Flag question

Today we will start our Module on Jupyter notebooks and explore some of the advantages of being able to use and share them on the internet. There are a variety of sites that can host a Jupyter notebook, but we will run one of the Seismo-Live notebooks on the Binder cloud hosting site. To start with, go to the Seismo-Live website:

http://www.seismo-live.org/

On the webstite, you should choose "View Jupyter Notebooks". Next you will see a list of topic areas, and I would ask that you click on Python Introduction, then Python_Crash_Course, then click the Open button. This will open a static preview of the Jupyter Notebook.

In many cases as a scientist, the page that comes up could be all you need to learn about how someone is using code, text, and figures to explain their science. However, it is not interactive and you cannot "play" with their code in the notebook. To open an interactive version of the notebook, click on the OPEN LIVE ON BINDER button (not the OPEN ALL ON BINDER button). This should bring up a new browser window that has the Binder logo circling on it for a minute or so. Eventually, it should bring up an interactive Jupyter Notebook version of the page.

What is the name of the notebook, displayed in the upper left next to the Jupyter logo?

NOTE: The binder may take a long while to load.

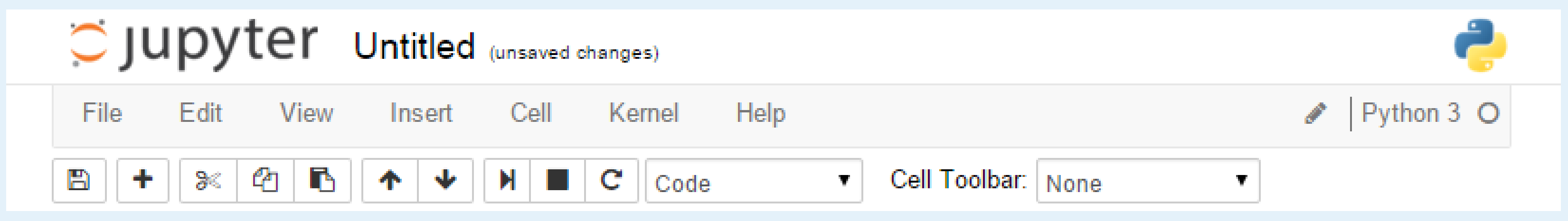
Select one:
- a. Scientific Python
- b. Seismo-Live
- c. Python Crash Course
- [✓ d. Python_Crash_Course] ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 2
Correct
1.00 points out of 1.00
Flag question

Right below the title of the notebook is the "header" that contains a menubar and a toolbar. This header remains fixed at the top of the screen, even as the body of the notebook is scrolled. The menubar and toolbar contain a variety of actions which control notebook navigation and document structure, but these are more designed for creating your own notebook than exploring someone else's, so we will not use these very much today.



IMPORTANT NOTE: Binder is an amazing resource, but since it is an open resource, it needs to be mindful of how many concurrent users there are. This means that if you are inactive for more than 10 minutes, it will shut off your connection and you will need to restart the live session from the static view webpage. To help you identify when this disconnection occurs, it will tell you with a browser pop up message and a red Not Connected message will appear on the right side of the header. So it would be wise to keep the static view of the notebook open while you are working on the interactive page in case you need to re-open the interactive page.

So just to be clear, what is the maximum amount of time in minutes you can get distracted while working on a Binder live notebook without losing your progress?

Answer: 10 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 3
Correct
1.00 points out of 1.00
Flag question

Below the header is the body of the notebook. The body is composed of "cells" of information. Each cell contains either markdown, code input, code output, or raw text. Cells can be included in any order and edited at-will, allowing for a large amount of flexibility for constructing a narrative.

- ***Markdown cells*** - These are used to build a nicely formatted narrative around the code in the document. The majority of the notebook you are interacting with today is composed of markdown cells.

- ***Code cells*** - These are used to define the computational code in the document. They come in two forms: the *input cell* where the user types the code to be executed, and the *output cell* which is the representation of the executed code. Depending on the code, this representation may be a simple text value, or something more complex like a plot or an interactive widget.

- ***Raw cells*** - These are used when text needs to be included in raw form, without execution or transformation.

You can identify individual cells by clicking anywhere in the body of the notebook. This should highlight the cell with a thin blue box around the cell. Which type of cell is the first one in the Python_Crash_Course notebook?

Select one:
- a. code output
- b. raw text
- c. code input
- [✓ d. markdown] ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 4
Correct
1.00 points out of 1.00
Flag question

Which cell number is the first to have the code input type?

Answer: 4 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 5
Correct
0.67 points out of 1.00
Flag question

The beauty of a live Jupyter notebook is that you can interact with the code in the notebook. To run the code, you just need to click the play button on the left side of the code cell, or in the header underneath the notebook title.

Before you click the play button, notice the text a little further left of the play button that should say:

```
In [ ]:
```

What happens inside the [ ] symbol after you push the play button?

Select one:
- a. It changes to [1]
- b. It changes to [*] for a moment, then it changes to [2]
- c. It changes to [*] and then stops
- d. Nothing
- [✓ e. It changes to [*] for a moment, then it changes to [1]] ✓

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 6
Correct

Once you have run this code, the key libraries we will need for this Jupyter notebook have been loaded. There will not be a code output cell afterwards because there is no output for this portion of code. The next cell after the code shows a bunch of markdown text with links to help pages for learning about Python. You are welcome to review these if you would like a refresher or to learn some more about Python, but I think you should have enough knowledge of Python from previous assignments that you don't need to review these now. The next cell after this starts the instructions for introducing

1.00 points out of 1.00

Flag question

key Python concepts with code. The code after the Numbers heading is where you can focus your attention next. Go ahead and click the play button to see what the output of this code is. Which of the following does it display as output?

Select one or more:

- [ ] a. 3.0 + 4j
- [ ] b. (9.0+16j)
- [x] c. (-7+24j) ✓ 1 of 2 correct answers
- [ ] d. 3
- [x] e. 3.0 ✓ 1 of 2 correct answers

Check

Correct
Marks for this submission: 1.00/1.00.

Question 7
Correct
1.00 points out of 1.00
Flag question

Since folks may not have a strong background in complex numbers, we can modify this code to help us understand them a bit better. A complex number has a real part (the regular number like 3.0) and an imaginary part (the part with a j like 4j). You can think of j as the square root of -1. We call it imaginary because there is no real number you can square to get -1. To check and make sure you understand this, click on the code cell and change the line of code that sets the complex number to be:

```
c = 0.0 + 2j
```

What is the answer when it prints the e variable (the square of the c variable)?

Select one:

- a. (4+0j)
- [✓ b. (-4+0j)] ✓
- c. 4
- d. -4

Check

Correct
Marks for this submission: 1.00/1.00.

Question 8
Correct
0.80 points out of 1.00
Flag question

The next code cell demonstrates how Python handles text strings and a variety of tricks with them. Which of the following are full lines of output from this code cell when you run it?

Select one or more:

- [x] a. New York 1 2 ✓ 1 of 5 correct answers
- [ ] b. w Y
- [ ] c. New
- [x] d. new york ✓ 1 of 5 correct answers
- [ ] e. New York
- [x] f. N k ✓ 1 of 5 correct answers
- [x] g. York ✓ 1 of 5 correct answers
- [x] h. I am in New York ✓ 1 of 5 correct answers

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.80/1.00**.

Question 9
Correct
1.00 points out of 1.00
Flag question

Which of the following would produce an output of "new"? Feel free to try these in the code cell to see which one is the correct answer.

Select one:

- a. print(location[3].lower())
- b. print(location[:2].lower())
- c. print(location[2:].lower())
- [✓ d. print(location[:3].lower())] ✓ Correct. Make sure to run this in the code cell.
- e. print(location[3:].lower())
- f. print(location[2].lower())

Check

Correct
Marks for this submission: 1.00/1.00.

Question 10
Correct
1.00 points out of 1.00
Flag question

For the **Exercise** section immediately following the output from the last question, go ahead and put `name = "mike"` in the first empty code cell and then run it. What would you need to type in the second empty cell to get it to print my name with the first letter capitalized?

Select one:

- [✓ a. print(name.capitalize())] ✓ Correct. Make sure to run this in the code cell.
- b. print(name.upper[])
- c. print(upper(name))
- d. print(name.capitalize[])
- e. print(name.upper())
- f. print(capitalize(name))
- g. print(upper[name])
- h. print(capitalize[name])

Check

Correct
Marks for this submission: 1.00/1.00.

Question 11
Correct
1.00 points out of 1.00
Flag question

The next code cell demonstrates how you can use lists in Python (sort of like NumPy arrays, but not quite). Which of the following are outputs of this code cell when you run it?

Select one or more:

- [ ] a. [(-4+0j), 1, 2]
- [x] b. [1, 2.0, 2j, 1, 2, 3, 'hello', 'you'] ✓ 1 of 5 correct answers.
- [x] c. [2j, 1, 2] ✓ 1 of 5 correct answers.
- [x] d. 1 ✓ 1 of 5 correct answers.
- [x] e. [2, 3, 'hello'] ✓ 1 of 5 correct answers.
- [x] f. [1, 2.0, 2j] ✓ 1 of 5 correct answers.
- [ ] g. [1, 2.0, (-4+0j)]
- [ ] h. [1, 2.0, (-4+0j), 1, 2, 3, 'hello', 'you']
- [ ] i. [1, 2.0, (3+4j)]

Check

Correct
Marks for this submission: 1.00/1.00.

Question 12
Correct

Which of the following commands could be added to the end of the code cell to print:

```
Thank You
```

1.00 points out of 1.00

Flag question

without any brackets?

Select one:

- [✓ a. print ("Thank",everything[-1].capitalize())] ✓ Correct. Make sure to run this in the code cell.
- b. print ("Thank",everything[:-1])
- c. print ("Thank",everything[-1:].capitalize())
- d. print ("Thank",everything[-1:].upper())
- e. print ("Thank",everything[-1].upper())
- f. print ("Thank",everything[:-1].upper())
- g. print ("Thank",everything[:-1].capitalize())
- h. print ("Thank",everything[-1:])
- i. print ("Thank",everything[-1])

Check

Correct
Marks for this submission: 1.00/1.00.

Question 13
Correct
1.00 points out of 1.00
Flag question

The next code cell demonstrates how you can use dictionaries in Python, one of the last things we learned about in our ObsPy tutorials. Which of the following commands would print:

```
Muster
```

without any brackets?

Select one:

- a. print (information["name"][:6])
- b. print (information["surname"][1:6])
- c. print (information[1:6]["name"])
- d. print (information[1:6]["surname"])
- e. print (information["name"][1:6])
- [✓ f. print (information["surname"][:6])] ✓ Correct. Make sure to run this in the code cell.
- g. print (information[:6]["surname"])
- h. print (information[:6]["name"])

Check

Correct
Marks for this submission: 1.00/1.00.

Question 14
Correct
1.00 points out of 1.00
Flag question

The next code cell introduces the concept of functions. We have not had an opportunity to create our own function within Python code yet, so this is a helpful illustration of how to do that. The markup cell before the code indicates that functions are a great way to conquer a big problem by dividing it into a series of smaller, more manageable ones. We have already been using functions in previous tutorials, but we have imported them instead of creating them. This code illustrates how you can define them using the `def` command. Note that the `return` command is used to define what information is produced when the function is called.

What is the largest number produced by this code?

Answer: 216 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 15
Correct
1.00 points out of 1.00
Flag question

Which of the following commands could be added at end of the code cell to output the number 64?

Select one or more:

- a. print(do_more_stuff(2, 3, 2))
- b. print(do_stuff(4, 3))
- [✓ c. print(do_stuff(4, 16))] ✓ 1 of 2 correct answers.
- d. print(do_stuff(16, 3))
- e. print(do_more_stuff(3, 2, 2))
- [✓ f. print(do_more_stuff(2, 2, 3))] ✓ 1 of 2 correct answers.

Check

Correct
Marks for this submission: 1.00/1.00.

Question 16
Correct
1.00 points out of 1.00
Flag question

The next code cell demonstrates how to import functions in Python, which is a concept we have used frequently in our previous Python tutorials. I do like how this code block illustrates the various ways to import functions into Python. I think it can get a little confusing for new users because there is not one set way to import libraries and functions, although I have been trying to illustrate common approaches in the previous tutorials.

Let's see the values of the variables in this code cell by adding the following command at the end:

```
print (a * c)
```

What is the output of this command?

Answer: -1.0 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 17
Correct
1.00 points out of 1.00
Flag question

Which of the following commands would print a value of 1.0?

Select one:

- a. print (sine (pi / 2))
- b. print (math.sine (pi))
- c. print (math.sine (pi / 2))
- d. print (sin (pi / 2))
- [✓ e. print (math.sin (pi / 2))] ✓
- f. print (sine (pi))
- g. print (sin (pi))
- h. print (math.sin (pi))

Check

Correct
Marks for this submission: 1.00/1.00.

Question 18
Correct
1.00 points out of 1.00
Flag question

The next code cell will print out all of the functions available in the math library that you imported. Which of the following functions are available in this library?

Select one or more:

- a. root
- [✓ b. tanh] ✓ 1 of 4 correct answers.
- [✓ c. exp] ✓ 1 of 4 correct answers.
- [✓ d. log] ✓ 1 of 4 correct answers.

- [ ] e. power
- [✓ f. log10] ✓ 1 of 4 correct answers.

Check

Correct
Marks for this submission: 1.00/1.00.

Question 19
Correct
1.00 points out of 1.00
Flag question

Let's move down to the next section on Control Flow. The first code cell in this section introduces the use for the `for` command. A for loop is used for iterating over a sequence (like a list, a dictionary, or even a string of text characters). The for loop can execute a set of statements, once for each item in the sequence.

What would the output be if you add a command at the end of the first code cell in the Control Flow section that looks like this?

```
print (item)
```

Select one:
- a. b
- b. a a
- c. b b
- d. a
- e. item
- [✓ f. c] ✓
- g. c c

Check

Correct
Marks for this submission: 1.00/1.00.

Question 20
Correct
1.00 points out of 1.00
Flag question

The second code cell illustrates a common use of the for command where it is coupled with the range() function. To help illstrate how this works, add a command at the end of the code cell (and outside the `for` loop) that looks like this:

```
print (range(4))
```

What does it produce?

Select one:
- [✓ a. range(0, 4)] ✓
- b. {0 1 2 3}
- c. [0 1 2 3]
- d. [0, 1, 2, 3]
- e. {0, 1, 2, 3}

Check

Correct
Marks for this submission: 1.00/1.00.

Question 21
Correct
1.00 points out of 1.00
Flag question

To make sure we illustrate how the for loop works on a string, let's add the following two lines to the end of the code cell (again, outside the first for loop):

```
for x in "banana":
        print(x)
```

Make sure to include those spaces before the print(x) command because this is how Python knows the command is to be executed as part of the loop. What does this pair of commands output?

Select one:
- a. the word banana on one line
- b. the numbers 0 1 2 3 4 with one number on each line
- c. the numbers 0 1 2 3 4 with one number on each line, starting with the last number and listing the numbers in reverse order
- d. the numbers 0 1 2 3 4 with on one line
- e. the numbers 0 1 2 3 4 with on one line, starting with the last number and listing the numbers in reverse order
- [✓ f. the word banana with one letter on each line] ✓
- g. the word banana with one letter on each line, starting with the last letter and listing the letters in reverse order
- h. the word banana on one line, starting with the last letter and listing the letters in reverse order

Check

Correct
Marks for this submission: 1.00/1.00.

Question 22
Correct
1.00 points out of 1.00
Flag question

The next code cell helps to introduce if/then/else statements. We have not used these statements yet in Python, but they are very common in scientific programming, so it is good for you to get a chance to interact with them. What does the code output if you set the age to be 10?

Select one:
- [✓ a. Older than 10] ✓
- b. 10
- c. Younger than 10
- d. Wait what?
- e. (nothing)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 23
Correct
1.00 points out of 1.00
Flag question

In the next code cell there are two loops that create the same list of numbers. What is the output if we add this command at the end of the code?

```
print(b[-1])
```

Select one:
- a. {9}
- b. {10}
- [✓ c. 8] ✓
- d. 10
- e. [9]
- f. [8]
- g. [10]
- h. 9
- i. {8}

Check

Correct
Marks for this submission: 1.00/1.00.

Question 24
Correct
1.00 points out of 1.00

How would we change the last loop in this script to output odd values instead of even?

Select one:
- a. b.append(i).odd()

Flag question

- b. print (b.odd())
- c. odd(b.append(i))
- [✓ d. if i % 2:] ✓ Correct. Go ahead and run this command instead of `if not i % 2`
- e. b = [i for i in a if i % 2]
- f. print (odd(b))

Check

Correct
Marks for this submission: 1.00/1.00.

Question 25
Correct
0.00 points out of 1.00
Flag question

The next section and corresponding code cell is on Error Messages. Take a minute to make sure you read the markdown cell this time because it is good advice about error messages and how to use them. Try running the code - what happens? As it is originally written, nothing will happen because the # comment symbol has prevented the do_something(1, 2) command from being run. The program can read the function without causing any errors. Go ahead and remove the # and try running the code with this command enabled. The error message should provide information about what went wrong. What caused the error?

Select one:
- a. The `something_else` function is called with incorrect amount of information.
- [✓ b. The variable `something_else` has not been given a value.] ✓
- c. The variable `do_something` has not been given a value.
- d. The `do_something` function is called with incorrect amount of information.

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question 26
Correct
0.00 points out of 1.00
Flag question

The Python error message provides line numbers for the code to help you identify what went wrong and which step caused the problem. Which of the following is correct about when the operation stopped?

Select one:
- a. Line 4 when the program first receives a call to a function with a flaw in it.
- [✓ b. The program reads all the way to line 4, then it tries to execute the earlier function and fails.] ✓
- c. Line 3 when the program reaches the end of the function definition.
- d. Line 1 when the program tries to define a function without a correct command.
- e. Line 2 when the program first reads an incorrect command and fails.

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question 27
Correct
1.00 points out of 1.00
Flag question

The next section is designed to introduce users to the Scientific Python Ecosystem that includes NumPy, SciPy, Matplotlib, and pandas. We've had a chance to explore most of these, but this notebook has some different uses of these libraries than we have seen before, which is nice.

The first part of this section is on NumPy. The code demonstrates the creation of an array using the linspace() function to create a linear array - in this case an array with a million samples in it! The code squares each sample and then adds them all together with the sum() function. Go ahead and run the code cell to see what the sum is. Considering how much math it has to do, how long does it take?

NOTE: If the code cell does not run, try changing 1E6 to 1000000

Select one:
- a. About a minutes
- [✓ b. Really short, less than a second or so.] ✓ Yep, isn't that crazy!
- c. About 10 seconds
- d. It takes forever
- e. About 10 minutes

Check

Correct
Marks for this submission: 1.00/1.00.

Question 28
Correct
0.00 points out of 1.00
Flag question

As written, the code does not output anything from the second part where it creates an array of random values, then takes the fast-fourier transform (fft), and then the inverse of the fast-fourier transform (ifft). To see how this part of the code works, add a `print(x)` command after the random() command and then again at the end of the code. This will help you to see the random values right after they are created and then after the fft and ifft has been applied. What do these print commands show?

Select one:
- a. An array of 100 complex point numbers before, and 100 different complex numbers afterwards
- b. An array of 100 floating point numbers before, and 100 different floating point numbers afterwards
- c. An array of 100 complex point numbers before, and 100 very similar floating point numbers afterwards
- d. An array of 100 complex point numbers before, and 100 different floating point numbers afterwards
- e. An array of 100 complex point numbers before, and 100 very similar complex numbers afterwards
- [✓ f. An array of 100 floating point numbers before, and 100 very similar complex numbers afterwards] ✓
- g. An array of 100 floating point numbers before, and 100 very similar floating point numbers afterwards
- h. An array of 100 floating point numbers before, and 100 different complex numbers afterwards

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question 29
Correct
1.00 points out of 1.00
Flag question

The next section is on SciPy, which is a library we have not had the chance to use yet. It provides many user-friendly and efficient numerical routines, such as routines for numerical integration, interpolation, optimization, linear algebra, and statistics. You can read more about it here if you would like to use it for your own science:

https://docs.scipy.org/doc/scipy/reference/

The code provided for you in this notebook is designed to demonstrate how interp1d() can be used to mathematically interpolate a 1-dimensional function from an input series of data points. The code starts by creating a series of whole numbers from 0 to 10, squares these numbers, divides by 9, takes the negative, and then the cosine of this value. Pretty complicated, eh? Go ahead and add a line after the y variable is created to print the y values. How would you describe the y function?

Select one:
- a. Rapidly oscillates between -1 and 1 before declining to 0.
- b. Declines gradually from 1 to -1.
- c. Rapidly oscillates between -1 and 1.
- d. Declines gradually from 1 to 0.
- e. Rapidly oscillates between -1 and 1 before asymptotically approaching 1.
- [✓ f. Declines gradually and then rapidly oscillates between -1 and 1.] ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 30
Correct
1.00 points out of 1.00
Flag question

Then the last part of the code performs the interpolation. Go ahead and add a line after the y variable is created to print the f2 values. Approximately how many data points are interpolated per original data point?

Select one:
- a. 1

○ b. 100

○ c. 101

○ d. 5

[✓ e. 10] ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question **31**
Correct
0.33 points out of 1.00
Flag question

The last section of the notebook we will examine is on Matplotlib (the rest of the notebook are Exercises that you are welcome to try on your own). The code it provides for Matplotlib shows how you can create plots in a Jupyter notebook. I will take a moment to stress how important this is and how this is so attractive to scientists. In essence, you can show people how you make the figures in your analysis and resulting publication. What a great way to share science! Ok, so for this code, it creates a sine wave and plots it (sound familiar?). Go ahead and run the code to see this. Since you have some experience with the Matplotlib pyplot library, I would like you to adjust the code to plot the sine curve with only 20 data points shown as circles connected with a line. Which of the following revisions to the plt.plot() call would accomplish this?

Select one:

○ a. plt.plot(np.sin(np.linspace(0, 2 * np.pi, 10)), color="green", label="Some Curve", 'o')

○ b. plt.plot(np.sin(np.linspace(0, 20 * np.pi, 2000)), color="green", label="Some Curve", 'o')

○ c. plt.plot(np.sin(np.linspace(0, 2 * np.pi, 20)), color="green", label="Some Curve", 'o')

○ d. plt.plot(np.sin(np.linspace(0, 20 * np.pi, 2000)), color="green", label="Some Curve", marker='o')

○ e. plt.plot(np.sin(np.linspace(0, 2 * np.pi, 10)), color="green", label="Some Curve", marker='o')

[✓ f. plt.plot(np.sin(np.linspace(0, 2 * np.pi, 20)), color="green", label="Some Curve", marker='o')] ✓

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Finish review