**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20 may 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **python** | **USN:** | **4al16ec074** |
| **Topic:** | **string concatenation, string multiplication and format command with string** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\Nelson\Desktop\o1.PNG**  **C:\Users\Nelson\Desktop\o2.PNG** |
| Printing  Analyze and run the Command below.  Make changes and develop a state of mind to tear apart the code for learning.  print ("Hello World!")  print ("I'd much rather you 'not' worry about the errors.")   1. print ('I "said" do not worry about errors in Python.') 2. Think of 3 ways and break the code : 3. Single quotes or double quotation can be used, but to wrap around single quotes use double and vice versa.   print ("most of the time comment is after to command to undo/redo something.")   1. print("This will run.")   Think of 3 ways to break the code  Read the below aloud:  # A comments, this is so you can read your program later.  # Anything after the # is ignored by python.   1. # hhhhhhhh 2. print("I could have code like this.")   Variables - Int and Str also system variables and str functions  Let us print few things and understand variable types.  Run the below code on your notebook.  Integer vs. float type and what happens at interaction!   1. a = 5.01 2. b=1 3. c=a\*b 4. type(c)   This code calculates and then prints.   1. print("Carefully observe the formatting of the output:") 2. print("Balls", 25 + 30 / 6) 3. print("Stick", 100 - 14 \* 3 % 4) 4. print("Oh, did you observe the difference?.") 5. print("Is it true that 3 + 8 < 12 - 7?") 6. print("Oh, that's why it's False.")          What is happening with dot and whole numbers…  Lets run and find out, then think, then play and then break   1. lines= 100 2. word = 4.0 3. pages=100 5. lines pages= lines- pages 6. diff= lines - word 7. division = lines/word 9. print ("There are", lines, "liens available.") 10. print ("There are only", pages, "pages available.") 11. print("There will be", division , "lines per word") 12. print("Difference", diff, "lines minus word.") 13. # Observe the interaction between the decimal and the non decimal variables!   Find type of variable  mobile2=857906879575   1. type (mobile2) 2. mobile = "857906879575" 3. type(mobile)   decimal number=98.12   1. type (decimal number)   Let us play with some system files…  from sys import argv    2. # read the WYSS section for how to run this 3. first, second, third = argv 5. #print("The script is called:", script) 6. print("Your first variable is:", first) 7. print("Your second variable is:", second) 8. print("Your third variable is:", third)   What was commented and why it was commented, read the error and see what happens.   1. from sys import argv 3. script, user named = argv 4. prompt = '' 6. print (f"Hi {username}, I'm the {script} script.") 7. print("I'd like to ask you a few questions.") 8. print(f"Do you like me {username}?") 9. likes = input(prompt) 11. print(f"Where do you live {username}?") 12. lives = input(prompt) 14. print("What kind of computer do you have?") 15. computer = input(prompt) 17. print(f""" 18. Alright, so you said {likes} about liking me. 19. You live in {lives}. Not sure where that is. 20. And you have a {computer} computer. Nice. 21. """)   from sys import argv   1. script,xxx, filename = argv 3. txt = open(filename) 5. print(f"Here's your file {filename}:") 6. print(txt.read()) 8. print("Type the filename again:") 9. file\_again = input("> ") 11. txt\_again = open(file\_again) 13. print(txt\_again.read())   Reading from a system file of azure.   1. a = 'this is a string' 2. #a[10] = 'f' 3. b = a.replace('string', 'longer string')   What happens if we force python to break a string?  What will Python do?   1. s = 'python' 2. list(s)   .format put the thing that are in parenthesis in curly brackets.   1. formatter = "{} {} {} {}" 3. print(formatter.format(True, False, False, True)) 4. print(formatter.format(formatter, formatter, formatter, formatter)) 5. ment after is ignored 7. # You can also use a comment to "disable" or comment out a piece of code: 8. # print ("This won't run.") 10. print("This will run.")   What is happening we spread across a string to multiple lines?  Line and tab are important and act delimiters or boundaries ..   1. print(""" 2. x 4. x 6. x 7. """) |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | | | |
|  | | | |
|  | | | |