Tutorial-01

July 20, 2024

Note: all tutorials for COMP7023 need to be completed in Python

1 Q1 – Variables

1. Assign the value 9 to the variable `x`
2. Assign the value 7 to the variable `y`
3. Find the sum of `x` and `y` and assign to a new variable `z`
4. Find the product of `x` and `y` and assign to a new variable `z2`
5. Find the ratio between `x` and `y` and assign to a new variable `z3`. What type of variable is `z3`?
6. Create a list of the first 5 square values and assign to the variable `L`.
7. Create a string "Monty Python the Flying Circus" and assign to the variable `x`

2 Q2 – Operators

2.1 Basic arithmetic

Calculate the following in Python:

1. $5 - 8$
2. 6×3
3. $19 \div 7$
4. reminder of $19 \div 7$
5. integer result of $19 \div 7$
6. 8^2
7. 6^4
8. $(3 + 4) \times 7$
9. 2^{4+7}

2.2 Assignment

Given `x` = 10, calculate the following:

1. add 3 to `x`, and assign to `x`
2. divide `x` by 2, and assign to `z`
3. given `y = 5`, calculate $x \times y$ and assign to `x`
4. decrement `x` by 1

3 Q3 – Lists, tuples, dictionaries, sets

3.1 Lists

1. Create a list of the first 5 prime numbers, and assign to `L`
2. What's the first element of `L`
3. Append to the list the next 3 prime numbers
4. Get from `L` the last 5 elements
5. What's element 3 of `L`
6. Get every 2nd element from `L`, starting from the second element
7. What's the length of the list?
8. Create a copy of `L` and assign to `L2`
9. Replace element 7 of `L2` with the value 88,
10. Did you change the values in `L`?

3.2 Tuples

1. Create a tuple for two values: 3, 4, and assign to the variable `x`
2. Create a tuple for two items: the value 3, and the string "Two", and assign to `x`
3. Create a tuple for the first 5 positive odd integers
4. Create a tuple to include a short two element list, a string, and a float number
5. Unpack the three items from the previous part into separate variables
6. Can you think of a situation when you should use tuple for your data instead of list?

3.3 Dictionaries

1. Create a dictionary mapping the first 5 numbers to their English words
2. Create a dictionary mapping the English words for number 1-5 to their French equivalent (or a language of your choice)
3. Find the French word for 3 using the two dictionaries you created above
4. Add a new item in the first dictionary for the number 10

3.4 Sets

1. Create a set with the names of the Monty Pythons: Graham Chapman, John Cleese, Terry Gilliam, Eric Idle, Terry Jones, and Michael Palin
2. Create a new set with names of the Pythons who have passed away: Graham Chapman, Terry Jones
3. Use set operation to find the Pythons who are still alive
4. Provide the Python code to find if "Carol Cleveland" is in the set of Monty Pythons

4 Q4 – Control Flow

4.1 `if` loops

Write `if` statements to perform the following tasks:

1. If x is even, subtract 1 from x.
2. If x is equal to y, print "finished"
3. If x is negative, make it positive

4. If x is equal to "degrees", convert y to radians, than compute sin(y), otherwise compute sin(y)
5. If x is greater than y and y is greater than z, print "sorted"
6. If x is greater than 50 and y is equal to "residential" or x is greater than 100 and y is equal to "freeway", print "speeding!", otherwise print "good speed".

4.2 `for` loops

Write `for` loops to perform the following tasks:

1. Print the numbers 1 to 10, one number at a time.
2. Sum the numbers 1 to 10.
3. Count how many items in `z = [12, 34, 45, 23, 16]` are greater than x (Set x to your favourite number).
4. Print a statement for every letter in `ascii_lowercase` stating that each letter is your favourite. The variable `ascii_lowercase` from the `string` library contains all the lower case letters of the alphebet.
5. For the set {Linux, OSX, Windows} compare all pairs to determine which is greater. For each pair, print which is greater than which (this requires a for loop in a for loop).

4.3 `while` loops

Write `while` loops to perform the following tasks:

1. Print a list of numbers that are multiples of 10 smaller than 100
2. Print random numbers generated using `uniform(0,1)` from the `random` library until a number is printed that is greater than 0.8. Break the code once you've printed 5 numbers.
3. Sum the items in the list `z = [12, 45, 47, 82, 9, 10]`
4. Ask for user inputs of numbers, stop is the user inputs 0. (The function for user input is `input()`)
5. Find the factorial of a given number.

5 Q5 – Functions

5.1 Mathematical Functions

Write Python functions to compute the y value of a mathematical function when given its x value. They should all have inputs of x and return y.

1. $y = 3x + 9$
2. $y = 2x^2 + 4x + 1$
3. $y = sin(x)$
4. $y = \frac{1}{1+e^{-x/5}}$

nb. It is best to use `numpy` for maths operations and creating arrays. To test the functions, create an x varible as a numpy array between -50 and 50, like this: `x = numpy.arange(-50,50)`.

5.2 Simple Functions

Write Python functions for the following tasks:

1. A function that takes two inputs, sums them, and returns the sum.

2. A function that takes one input, x, returns as a tuple the sin, cos and tan values of x
3. A function that takes a list and print out each item in the list
4. A function that takes two inputs, x and y, where y has a default value of 3, returns the product
5. A function that returns the factorial of the input value

6 Q6 – Iterators

1. Write a program that display characters that are present at an even index number in the string variable `word`.
2. Write a program that takes a string variables `word` and prints out for every character in the word "The `i`th character in `word` is `character`", where `i`, `word` and `character` are items you need to insert.
3. Print out "The French word for `english number` is `french number`" for the numbers 1-5. Hint: create two list of words and use the function `zip`. You're welcome to use another language.
4. Given a list of fruits, their prices and the quantities that you purchased, print out "You bought `x` `fruit name` for $ `price`" for each fruit in the list, where `x`, `fruit name` and `price` are items you need to insert.
5. Python dictionaries has an iterator called `item()`. Given a dictionary 'knights = {'Sir Bedevere': 'the Wise', 'Sir Lancelot': 'the Brave', 'Sir Galahad': 'the Pure', 'Sir Robin': 'the Not-Quite-So-Brave-as-Sir-Lancelot'}, print out the knights with their nicknames, e.g. "Sir Bedevere the Wise" on separate lines

7 Q7 – List comprehension

Use list comprehension to:

1. Create a list of cubics for the first 9 positive integers
2. Split a string into separate characters
3. Create a identical list from the first list
4. Given a list `knights = ['Sir Bedevere', 'Sir Lancelot', 'Sir Galahad', 'Sir Robin']`, create a new list with all lower case letters of the names.
5. Create a new list from `knights` with all the knights expect "Sir Robin" who will be replaced with "King Arthur"

8 Q8 – Strings

8.1 Strings as lists

The indexing, slicing, etc. of strings behaves exactly like that of lists. With that in mind:

1. Write the code to extract the odd indices from a string to create a new string
2. Create a new string made of the middle three characters of another string
3. Append to the string "Data Science" a second string "is fantastic!"
4. Combine the strings `ascii_lowercase` and `ascii_uppercase` from the `string` library, such that we have a string that has one uppercase letter folllowed by it's lower case version, i.e. "AaBbCc…" (hint: try the `zip` iterator.)

8.2 String manipulation

The following paragraph is from Wikipedia's page on Python about the name "Python":

Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture; for example, the metasyntactic variables often used in Python literature are spam and eggs instead of the traditional foo and bar. The official Python documentation also contains various references to Monty Python routines.

1. Count how many times the word "Python" appears in the paragraph. (hint: there's a `count()` method to strings.)
2. Replace all instances of the word "Python" with the word "Snake".
3. Find all instances of the word "Python". (hint: you'll need either a `for` loop, or use list comprehension; also look into the method `startswith()` for strings.)
4. Split the paragraph into the individual sentences, and save them into a list. So you should have three items in the list.

Note: there are many different ways to break up a long string in Python during input. You can concatenate strings using `+`, you can use the triple double quote, you can use `\` as line continuation. I leave this as an exercise to find out how best to break the paragraph up.

9 Q9 – Putting it together

9.1 Palindrom

Write a function to check if the input number is a palindrome. Test it with a few numbers.

Note: a palindrome number is a number that is the same after resersing the order of the digits. For example, 919 is a palindrome number.

9.2 Prime numbers

The basis of most computer security algorithms relies on the fact that it is difficult to factorise large numbers. Write a loop to identify the set of prime numbers that are less than 1000. Start by first writing the code to check if one integer is prime, the build up to checking all intergers less than 1000.

9.3 Counting items in a string

Count all letters, digits, and special symbols from a given string. (Hint: Python strings has methods like `isalpha` check if you have a letter for example.)

Test with a string such as "1ntr@dU5^nT#P7$hon" for example.