Assignment_0

September 22, 2023

1 Assignment 0 – *Basics of Python and Data Science* (55 marks)

1.0.1 Question 1 – Data science terminology (15 marks)

Read this news article. - a) (5 Marks) Is the main subject in this article an example of machine learning, statistics, or the data science process? Why? - b) (5 Marks) Make up an example of a data science process which involves the model discussed in this article. - c) (5 Marks) In your opinion, is the model discussed in this article an example of artificial intelligence?

1.0.2 Question 2 – Operators & Variables (14 marks)

- a) (2 Marks) In a print statement, what happens if you leave out one of the parentheses, or both?
- b) (2 Marks) If you are trying to print a string, what happens if you leave out one of the quotation marks, or both?
- c) (2 Marks) You can use a minus sign to make a negative number like -2. What happens if you put a plus sign before a number? What about 2++2?
- d) (2 Marks) In math notation, leading zeros are ok, as in 09. What happens if you try this in Python? Try 011.
- e) (2 Marks) Create a variable called "my_favorite_song" which contains your favorite song as a string
- f) (2 Marks) We could assign a variable as follows `n = 42` . What happens when you try 42 = n? How about x = y = 1?
- g) (2 Marks) In math notation you can multiply x and y like this: xy. What happens if you try that in Python? Create variables `x` and `y` and try it

[68]:
```python
# 2 a)

# In the print statement, if the outer parentheses is left out, then the␣
↪following error is displayed:
# "SyntaxError: incomplete input"
print('hi'

# In the print statement, if the inner parentheses is left out, then the␣
↪following error is displayed:
# "SyntaxError: unmatched ')'"
```

```
print'hi')

# In the print statement, if both parentheses are left out, then the following␣
↪error is displayed:
# "SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
↪"
print'hi'
```

```
  Cell In[68], line 5
    print('hi'
          ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

[65]:
```
# 2 b)

# If you leave out one of the quotation marks when trying to print a string,␣
↪then the following error is displayed:
# "SyntaxError: unterminated string literal (detected at line _)"
print('hi)

# If you leave out both of the quotation marks when trying to print a string,␣
↪then the following error is displayed:
# "NameError: name 'string_content' is not defined" -> string_content is the␣
↪string content that you are trying to print.
print(hi)
```

```
  Cell In[65], line 5
    print('hi)
          ^
SyntaxError: unterminated string literal (detected at line 5)
```

[35]:
```
# 2 c)
# In Python, you can use a plus sign before a number to indicate that the␣
↪number is positive.
# However, the plus sign in this case is not necessary because Python assumes␣
↪numbers without a sign are positive by default.

# For 2++2, pthon interpets this as adding two postiive values, since it␣
↪assumes that the first 2 is positive by default,
# and as for the plus signs, the first + sign is interpreted as a positive␣
↪sign, and the second + sign is the addition operator.
# As a result, the expression becomes 2 + 2, which equals 4.
# code to demonstrate:
print(+2)
print(2++2)
```

```
2
4
```

[36]:
```
# 2 d)
# Leading zeros in python results in the following error:
# "SyntaxError: leading zeros in decimal integer literals are not permitted;␣
↪use an 0o prefix for octal integers"
011
```

```
  Cell In[36], line 4
    011
    ^
SyntaxError: leading zeros in decimal integer literals are not permitted; use a␣
↪0o prefix for octal integers
```

[40]:
```
# 2 e)
my_favorite_song = "golden hour"
```

[69]:
```
# 2 f)

# The expression 42 = n causes an error because we are essentially assinging␣
↪value (n) to a literal (4).
# For varibale assignment, the equal sign (=)  expects a variable name on the␣
↪left-hand side. ->(n = 42 is the correct way)
# The error displayed is: "SyntaxError: cannot assign to literal here. Maybe␣
↪you meant '==' instead of '='?"
42=n
```

```
  Cell In[69], line 6
    42=n
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='
```

[70]:
```
# 2 f)

# x = y = 1 is a valid expression since it is simply assigning the same value␣
↪(1) to both variables 'x' and 'y'.
x = y = 1
```

```
[71]: # 2 g)
x=2
y=4
print(xy)
# the code above results in the following error: "NameError: name 'xy' is not
 ↪defined"
# The reasoning for this error is that python interpertes 'xy' as a variable
 ↪name rather than multiplying x by y.
# To multiply x by y in python, 'x*y' should be used instead 'xy', since '*' is
 ↪the multiplication operator.
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[71], line 4
      2 x=2
      3 y=4
----> 4 print(xy)
      5 # the code above results in the following error: "NameError: name 'xy'
 ↪is not defined"
      6 # The reasoning for this error is that python interpertes 'xy' as a
 ↪variable name rather than multiplying x by y.
      7 # To multiply x by y in python, 'x*y' should be used instead 'xy', sinc
 ↪'*' is the multiplication operator.

NameError: name 'xy' is not defined
```

1.0.3 Question 3 – Lists and Dictionaries (16 marks)

- a) (3 marks) Create a list of strings which represent your top 3 interests
- b) (3 marks) Access the last two elements of the list in one line of code
- c) (1 mark) Print the data type of the list

Use the following dictionary:

```
# Example dictionary representing information about a person
person = {
    "first_name": "John",
    "last_name": "Doe",
    "age": 30,
    "city": "New York",
    "email": "johndoe@example.com",
    "is_student": False,
    "hobbies": ["reading", "running", "cooking"],
    "address": {
        "street": "123 Main St",
        "city": "New York",
        "zip_code": "10001"
    }
}
```

- c) (5 marks) Print the first name, the zip_code and hobbies of the above person using the above dictionary `person`
- d) (2 marks) Set the city to be Toronto in the dictionary
- e) (2 marks) Print only the dictionary keys, then print only the dictionary values

[10]:
```
# 3 a)
topInterests = ["hockey","technology","crypto"]

# 3 b) using slicing to get last 2 elements
topInterests[-2:]

# 3 c)
print(type(topInterests))
print('\n')

person = {
    "first_name": "John",
    "last_name": "Doe",
    "age": 30,
    "city": "New York",
    "email": "johndoe@example.com",
    "is_student": False,
    "hobbies": ["reading", "running", "cooking"],
    "address": {
        "street": "123 Main St",
        "city": "New York",
        "zip_code": "10001"
    }
}

# 3 d)
print(person["first_name"])
print(person["address"]['zip_code'])
print(person["hobbies"])
print('\n')

# 3 e) set city as toronto
person["address"]['city']="Toronto"

# 3 f)
print(person.keys())
print('\n')
print(person.values())
```

```
<class 'list'>
```

```
John
10001
['reading', 'running', 'cooking']
```

```
dict_keys(['first_name', 'last_name', 'age', 'city', 'email', 'is_student',
'hobbies', 'address'])
```

```
dict_values(['John', 'Doe', 30, 'New York', 'johndoe@example.com', False,
['reading', 'running', 'cooking'], {'street': '123 Main St', 'city': 'Toronto',
'zip_code': '10001'}])
```

1.0.4 Question 4 – Loops and if statements (10 marks)

- a) (5 marks) Use a for loop to print the numbers from 1 to 10, each on a separate line.
- b) (5 marks) Modify your loop so that it still iterates through the numbers 1 to 10, but it prints only the odd numbers from 1 to 10 and it prints your name instead of the even numbers. Your output should look like:

```
1
Kelly
3
Kelly
5
Kelly
7
Kelly
9
Kelly
```

[6]:
```
for i in range (1,11):
    print (i)
print( '\n')

for i in range(1,11) :
    if (i%2!=0):
        print(i)
    else:
        print ('Priya')
```

```
1
2
3
4
```

```
5
6
7
8
9
10
```

```
1
Priya
3
Priya
5
Priya
7
Priya
9
Priya
```