

In-class worksheet 14

Mar 7, 2019

Complete the following tasks in python. You may do them either from the ipython command line (qt-console) or by creating code chunks within this jupyter notebook.

1. Working with numbers

Perform the following basic operations in python. Note that you will have to print your result with the function `print()` if you want to see the output in your ipython notebook.

1. Add two numbers
2. Multiply two numbers
3. Assign a number to a variable and then print the variable
4. Assign two numbers to two different variables, then assign the product of those two variables to a third. Print the third variable.

```
In [1]: # Your solutions go here. Create a separate code chunk for each question
        # using the + icon in the toolbar.

        # 1. Add 3 and 5
        3+5 # without printing, nothing visible happens
        print(3+5) # this generates output.

8
```

```
In [2]: # 2. Multiply 3 and 5
        print(3*5)

15
```

```
In [3]: # 3. Assign 3 to variable 'a' and print
        a = 3
        print(a)

3
```

```
In [4]: # 4. Assign 3 to variable 'a', 5 to variable 'b', product to 'c'
        a = 3
        b = 5
        c = a*b
        print(c)

15
```

2. Working with strings

Define a variable called `mystring`, which contains a lengthy string of some kind (random letters, your address, song lyrics, a haiku, whatever). Perform the following tasks with this variable:

1. Print the contents of your variable.
2. Create a new variable called `complete_string` which contains the phrase "Here is my string: " followed by the contents of your `mystring` variable. (Hint - use the `+` operator). For example, if your `mystring` variable contains the text "Time flies like an arrow; fruit flies like a banana.", then `complete_string` should read, "Here is my string: Time flies like an arrow; fruit flies like a banana."
3. Use indexing to select the first letter of your string, the first five letters, or the last seven letters.
4. Attempt to use indexing to replace the character in the 5th position of `mystring` with the letter "X". How did that go?
5. Use the `.split()` method to split your string into a list.
6. In your variable `mystring`, replace the all occurrences of a letter of your choice with a number. Which string method can achieve this goal?

```
In [5]: # Your solutions go here. Create a separate code chunk for each question
        # using the + icon in the toolbar.
```

```
# 1. Make variable and print
mystring = "Time flies like an arrow; fruit flies like a banana."
print(mystring)
```

```
Time flies like an arrow; fruit flies like a banana.
```

```
In [6]: # 2. Make variable 'complete_string'
        complete_string = "Here is my string: " + mystring
        print(complete_string)
```

```
Here is my string: Time flies like an arrow; fruit flies like a banana.
```

```
In [7]: # 3. Select pieces of string through indexing

        # first letter (remember, python counts from 0)
        selection = mystring[0]
        print(selection)
```

```
# first five letters
selection = mystring[0:5]
print(selection)
```

```
# last seven letters
selection = mystring[-7:]
print(selection)
```

```
T
Time
banana.
```

```
In [8]: # 4. Try to replace character through indexing
        # mystring[4] = 'X' # does not work, python strings cannot be changed in place
```

```
In [9]: # 5. Split the string
        mystring.split()
```

```
Out[9]: ['Time',
        'flies',
        'like',
        'an',
        'arrow;',
        'fruit',
        'flies',
        'like',
        'a',
        'banana.']
```

```
In [10]: # 6. Replace letter 'i' with 'l' using the `.replace()` function
         mystring.replace('i', 'l')
```

```
Out[10]: 'Tlme flles llke an arrow; fruit flles llke a banana.'
```

3. Working with lists

Define a list variable called `animals` which contains the following six entries: "monkey", "giraffe", "shark", "caterpillar", "squid", and "jellyfish". Perform the following tasks with this variable:

1. Select the first animal in the list.
2. Select the middle two animals (numbers 3 and 4) in the list.
3. Select the last animal in the list.
4. Use indexing to change the second animal in the list to "cat".
5. Use the list method `.append()` to add "spider" to the end of the list.
6. Use the operator `in` to determine if "dog" is in the `animals` list. Also try with "monkey".
7. Use the `.join()` method to create a string which contains each animal in the `animals` list separated by the string "; " (semicolon and a space). Read the documentation of `.join()` or google it to solve this problem. Hint: `.join()` is a string function that takes a list as its argument.

```
In [11]: # Your solutions go here. Create a separate code chunk for each question
         # using the + icon in the toolbar.

         # 1. Make list, select first animal
         animals = ["monkey", "giraffe", "shark", "caterpillar", "squid", "jellyfish"]
         animals[0]
```

```
Out[11]: 'monkey'
```

```
In [12]: # 2. Middle two animals (remember, python counts from 0, and we need to
         # index one past the last element we want to select)
         animals[2:4]
```

```
Out[12]: ['shark', 'caterpillar']
```

```
In [13]: # 3. Last animal
         animals[-1]
```

```
Out[13]: 'jellyfish'
```

```
In [14]: # 4. Change second animal to cat
animals[1] = "cat"
print(animals)

['monkey', 'cat', 'shark', 'caterpillar', 'squid', 'jellyfish']

In [15]: # 5. Append "spider" to the list
animals.append("spider")
print(animals)

['monkey', 'cat', 'shark', 'caterpillar', 'squid', 'jellyfish', 'spider']

In [16]: # 6. Check if "dog" is in the list
"dog" in animals

Out[16]: False

In [17]: # check if "monkey" is in the list
"monkey" in animals

Out[17]: True

In [18]: # 7. Join the strings in the list together
"; ".join(animals)

Out[18]: 'monkey; cat; shark; caterpillar; squid; jellyfish; spider'
```

4. Working with dictionaries

Create a dictionary called `molecules` with the following three key-value pairs: "hair" - "keratin"; "DNA" - "nucleotides"; "protein" - "amino acids". Perform the following tasks with this variable:

1. Print out the dictionary and examine the order of the key-value pairs. Is this what you expected?
2. Try to use indexing to select the second key-value pair from the dictionary. Does this work? Now try to index the value for "DNA".
3. Add another key-value pair, "ribosomes"-"RNA", to the `molecules` dictionary. Then, add another key-value pair, "RNA" - "nucleotides". Did this work? What does this tell you about the allowed uniqueness for dictionary keys and values?
4. Now, add yet another key-value pair, "ribosomes"-"rRNA" to the `molecules` dictionary. What happened? Can you see why?
5. Print out just the keys and just the values of the dictionary, using the dictionary methods `.keys()` and `.values()`.
6. Create a list of *tuples*, which in each tuple is the (key, value) pair. Use the `zip()` function for this. (You may need to look up this function). Save this list of tuples to a variable. Then re-cast the list of tuples back into a dictionary.

```
In [19]: # Your solutions go here. Create a separate code chunk for each question
# using the + icon in the toolbar.

# 1. Make dictionary and print it out
molecules = {"hair":"keratin", "DNA":"nucleotides", "protein":"amino acids"}
print(molecules)

{'DNA': 'nucleotides', 'protein': 'amino acids', 'hair': 'keratin'}

In [20]: # 2. Indexing goes by key, not by number
# molecules[1] # causes an error
```

```
In [21]: molecules["DNA"] # this works
```

```
Out[21]: 'nucleotides'
```

```
In [22]: # 3. Adding key-value pairs
molecules["ribosomes"] = "RNA" # this adds a new key "ribosomes"
print(molecules)

{'DNA': 'nucleotides', 'ribosomes': 'RNA', 'protein': 'amino acids', 'hair': 'keratin'}
```

```
In [23]: # The following adds a new key "RNA". It doesn't matter that "RNA" was
# already used as a value.
molecules["RNA"] = "nucleotides"
print(molecules)

{'RNA': 'nucleotides', 'DNA': 'nucleotides', 'ribosomes': 'RNA', 'protein': 'amino acids', 'hair': 'keratin'}
```

```
In [24]: # 4. Add yet another key-value pair, "ribosomes"->"rRNA"

# This overwrites the previous "ribosomes" entry, keys need to be unique
molecules["ribosomes"] = "rRNA"
print(molecules)

{'RNA': 'nucleotides', 'DNA': 'nucleotides', 'ribosomes': 'rRNA', 'protein': 'amino acids', 'hair': 'keratin'}
```

```
In [25]: # 5. Extract keys and values from dictionary
print(molecules.keys())
print(molecules.values())

dict_keys(['RNA', 'DNA', 'ribosomes', 'protein', 'hair'])
dict_values(['nucleotides', 'nucleotides', 'rRNA', 'amino acids', 'keratin'])
```

```
In [26]: # 6. Turn dictionary into list of tuples, and then back into dictionary

# First, create list of keys and values
# We need the `list()` function to actually create a list we can print
keys_values = list(zip(molecules.keys(), molecules.values()))
print(keys_values)

# Back to dictionary
dict(keys_values)

[('RNA', 'nucleotides'), ('DNA', 'nucleotides'), ('ribosomes', 'rRNA'), ('protein', 'amino acids'), ('hair', 'keratin')]
```

```
Out[26]: {'DNA': 'nucleotides',
'RNA': 'nucleotides',
'hair': 'keratin',
'protein': 'amino acids',
'ribosomes': 'rRNA'}
```