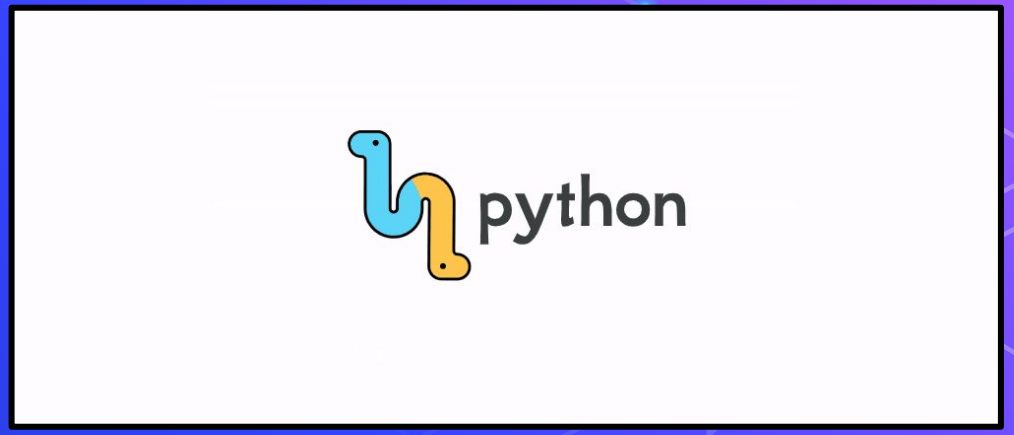
Learning Python

Table of Contents



Revisiting Problems...

→ Problem #1

Ask the user for 2 numbers.

If number 1 > number 2 ~ print “Number 1 is larger”

Else if number 1 = number 2 ~ print “Numbers are the same”

Else ~ print “Number 1 is smaller”

→ Solution #1

Does anyone in the chat have a solution for it? If so drop your code below!

Revisiting Problems...

→ Solution #1

Here is a version of one of the many correct answers...

```python
#Problem #1 Ask the user for 2 numbers. If number 1 > number 2 ~ print "Number 1 is Larger" - Else if number 1 = number 2 ~ print "Numbers are the same" - Else ~ print "Number 1 is smaller"
number1 = int(input("Enter an integer value for number 1"))
number2 = int(input("Enter an integer value for number 2"))

if (number1 > number2):
    print("Number 1 is larger")

elif (number1 == number2):
    print("Numbers are the same")

else:
    print("Number 1 is smaller")

print("\n" + "Program Finished!")
```

```
Enter an integer value for number 1 50
Enter an integer value for number 2 25
Number 1 is larger

Program Finished!
```

```
Enter an integer value for number 1 25
Enter an integer value for number 2 25
Numbers are the same

Program Finished!
```

```
Enter an integer value for number 1 25
Enter an integer value for number 2 50
Number 1 is smaller

Program Finished!
```

Revisiting Problems...

→ Solution #1

What if you entered “letters” and not “numbers”?

```
Enter an integer value for number 1 25
Enter an integer value for number 2 Fruits
Traceback (most recent call last):
```

```
Enter an integer value for number 1 25
Enter an integer value for number 2 Fruits
Traceback (most recent call last):
                                                    line 44, in <module>
    number2 = int(input("Enter an integer value for number 2"))
ValueError: invalid literal for int() with base 10: ' Fruits'
```

Revisiting Problems...

→ Problem #2

Set a word in a variable (ex. Your Name)

Use one of the Python Loops to print each letter in the word one by one. If 10 letters have been outputted then exit the program using exit()

→ Solution #2

Does anyone in the chat have a solution for it? If so drop your code below!

Revisiting Problems...

→ Solution #2

Here is a version of one of the many correct answers...

C: > Users > LouisVuitton > Desktop > Prob2.py > ...

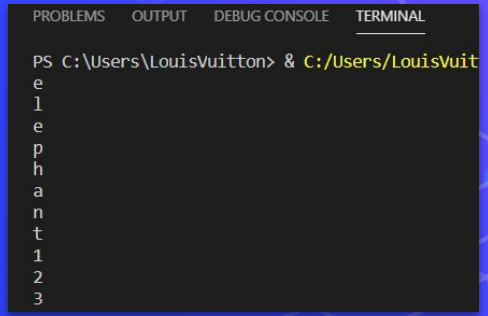
```python
####problem 2####
#→ Set a word in a variable.(ex. Your name)
#Use one of the Python loops taught to print each letter in the word one by one.
# If 10 letters have been outputted then exit the program using exit()

word = "elephant1234"

letter_sum = 0

for i in word:
    if letter_sum <= 10:
        print(i)
        letter_sum = letter_sum + 1

    else:
        exit()
```



Learning Lists...

→ What are lists?

- A list is one of the many types of ***sequence*** variables
- Data structure that holds a ***sequence*** of elements
- Each element/value inside of a list is called an “item”
- Lists can be changed and mutated (add or remove items)
- Similar to “ArrayLists” in Java

→ Examples of lists!

- Assortment of files on a PC
- Song playlists
- Browser bookmarks
- Emails

```
#Sequence Variables
h = ["apple", "banana", "cherry"] #list type variable
i = ("apple", "banana", "cherry") #tuple type variable
j = range(7) #range type variable
```

Different Types Of Sequence Variables

Example Of Lists...

To get started, let's create a list that contains items of the string data type:

```
sea_creatures = ['shark', 'cuttlefish', 'squid', 'mantis shrimp', 'anemone']
```

When we print out the list, the output looks exactly like the list we created:

```
print(sea_creatures)
```

```
Output
['shark', 'cuttlefish', 'squid', 'mantis shrimp', 'anemone']
```

Indexing Lists...

Indexing Lists

Each item in a list corresponds to an index number, which is an integer value, starting with the index number `0`.

For the list `sea_creatures`, the index breakdown looks like this:

'shark'	'cuttlefish'	'squid'	'mantis shrimp'	'anemone'
0	1	2	3	4

The first item, the string `'shark'` starts at index `0`, and the list ends at index `4` with the item `'anemone'`.

Indexing Lists...

```
print(sea_creatures[1])
```

```
Output
cuttlefish
```

```
sea_creatures[0] = 'shark'
sea_creatures[1] = 'cuttlefish'
sea_creatures[2] = 'squid'
sea_creatures[3] = 'mantis shrimp'
sea_creatures[4] = 'anemone'
```

Indexing Lists...

If we call the list `sea_creatures` with an index number of any that is greater than 4, it will be out of range as it will not be valid:

```
print(sea_creatures[18])
```

```
Output
IndexError: list index out of range
```

Indexing Lists...

We can concatenate string items in a list with other strings using the `+` operator:

```
print('Sammy is a ' + sea_creatures[0])
```

```
Output
Sammy is a shark
```

Modifying Items In Lists...

```
sea_creatures[1] = 'octopus'
```

Now when we print `sea_creatures`, the list will be different:

```
print(sea_creatures)
```

```
Output
['shark', 'octopus', 'squid', 'mantis shrimp', 'anemone']
```

Example Of Overwriting

Adding Items In Lists...

```
42  #Adding List Items
43  print("Section 6")
44  sea_creatures = ['shark', 'octopus', 'blobfish', 'mantis shrimp', 'anemone']
45  print("Current sea creatures list", sea_creatures)
46  print()
47
48  add_creature = input("Please enter the name of the creature you would like to add ")
49  sea_creatures.append(add_creature)
50  print()
51
52  print("New list of sea creatures", sea_creatures)
53  print()
```

```
Section 6
Current sea creatures list ['shark', 'octopus', 'blobfish', 'mantis shrimp', 'anemone']

Please enter the name of the creature you would like to add Jellyfish

New list of sea creatures ['shark', 'octopus', 'blobfish', 'mantis shrimp', 'anemone', 'Jellyfish']
```

Removing Items In Lists...

```
sea_creatures =['shark', 'octopus', 'blobfish', 'mantis shrimp', 'anemone', 'yeti crab']

del sea_creatures[1]
print(sea_creatures)
```

Output

```
['shark', 'blobfish', 'mantis shrimp', 'anemone', 'yeti crab']
```

Lists vs. Tuples...

→ What are tuples?

- Tuple is like a list it can hold a sequence of elements
- Tuple's use round brackets instead of square brackets
- Tuple can not be changed and mutated (**NO** adding or removing items)
- Similar to "Arrays" in Java

→ Differences!

- Tuple = Fixed size upon declaration (static)
- Tuple = Arrays (Java)
- Lists = Dynamic Size
- Lists = ArrayLists (Java)

```
#Sequence Variables
h = ["apple", "banana", "cherry"] #list type variable
i = ("apple", "banana", "cherry") #tuple type variable
j = range(7) #range type variable
```

Learning Dictionaries...

→ What are dictionaries?

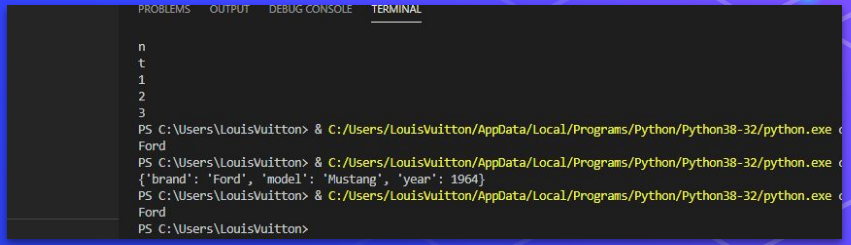
- Dictionaries are used to store data values in **key:value** pairs.
- Unordered, Changeable, No duplicates
- Dictionaries are written with curly brackets, and have keys and values

```
1
2   thisdict = {
3     "brand": "Ford",
4     "model": "Mustang",
5     "year": 1964
6   }
7
8   print(thisdict)
9
10
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

Find value using specific key...

```
1
2   thisdict = {
3     "brand": "Ford",
4     "model": "Mustang",
5     "year": 1964
6   }
7
8   print(thisdict["brand"])
9
```



Changing values in a dictionary...

```
# Changing and adding Dictionary Elements
my_dict = {'name': 'Jack', 'age': 26}

# update value
my_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(my_dict)

# add item
my_dict['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}
print(my_dict)
```

Iterating through a dictionary...

```
# Iterating through a Dictionary
squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}
for i in squares:
    print(squares[i])
```

Practice Problems

→ Problem #1

Ask the user to enter 5 items (ex. fruits) ~ Try to use a loop!

Create a list that stores these 5 items.

→ Problem #2

Create a dictionary containing 5 different cuisines(Chinese, Italian, Indian etc..) name your favorite dish in each cuisine as the “value” to the “key”. Then print each favorite dish one by one using a for loop.

The End

Any Questions?