

INTRODUCTION TO
COMPUTER PROGRAMMING

Tuples and Functions



OUTLINE

- List Methods
- Strings as Lists
- Tuples
- Functions



LIST METHODS

Some Python List Methods

append()	Add an element to the end of a list
insert()	Insert an item at the defined index
remove()	Removes an item from the list
pop()	Removes and returns an element at the given index
clear()	Removes all items from the list
Index()	Returns the index of the first matched item
Count()	Returns the count of number of items passes as an argument
Sort()	Sort items in a list in ascending order
Reverse()	Reverse the order of items in the list



LIST METHODS

List Method	Syntax	Result
append()	<code>myList.append(element)</code>	
insert()	<code>myList.insert(i, element)</code>	only updates the current list
remove()	<code>myList.remove()</code>	
pop()	<code>myList.pop(index)</code>	returns the element removed from the list (default is index of the last item -1)
clear()	<code>myList.clear()</code>	
index()	<code>myList.index(elem, start, end)</code>	the index of an element occurring from index start to end
count()	<code>letters.count(element)</code>	
sort()	<code>myList.sort(reverse = True)</code>	
reverse()	<code>myList.reverse()</code>	



Lists and Strings

- Recall that a string is a sequence of characters
- The `list()` method breaks a string into its individual letters and stores them as elements of a list
- The `split()` method breaks a string into individual words



STRING METHODS

List Method	Syntax	Result
capitalize() or upper()	myString.capitalize()	
casefold() or lower()	myString. casefold()	
count()	myString.count(substring)	
endswith()	myString.endswith(substring)	
find()	myString.find(substring)	Returns -1 if not found
index()	myString.index(substring, start, end)	
isalpha(), isnumeric(), isdecimal(), isdigit()		



TUPLES

- A **tuple** is a sequence of values, where each value in the tuple can be of any data type.
- Similar to lists, the elements of a tuple are comma separated. Elements are enclosed in parentheses.

- *Example:*

```
tupleA = 'a','s','d','f','g'  
tupleB = ('q',1,'#',2,'!')
```

```
t1 = 'a'    # what is the data type of t1?
```

```
t2 = ('a')  # what is the data type of t2?
```

```
t3 = 'a',   # a tuple with one element should be followed by a comma
```

? What is the result of tuple('apple') vs. list('apple')?



TUPLES ARE IMMUTABLE

- Elements in a tuple cannot be altered

? What happens if you change the value of an element in the tuple? `tupleA[1] = 'b'`?

- However, elements in a tuple can be replaced

```
tupleA = 'a','s','d','f','g'  
tupleA = ('A',) + tupleA[1:]
```



FUNCTION?

- How do we define a function in Math?

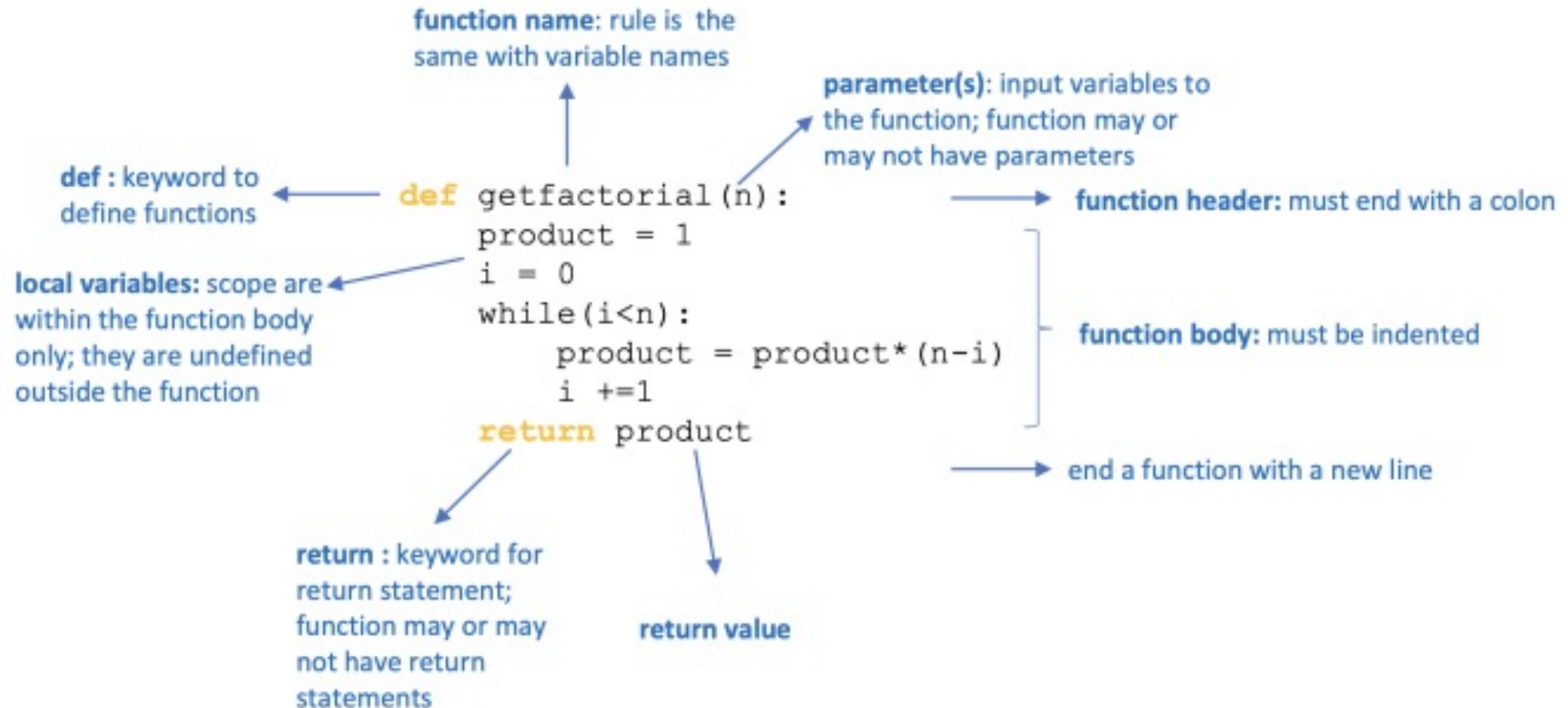


CONSIDER THIS:

- Suppose that you have a program that computes the average of scores of a student in 3 subjects: Math, English and Science
- The teacher uses such program to input the scores of the students and the program will have to output the average grade of a student.
- The teacher will have to repeat this process for all students
- Intuitively, you will have to run the same exact lines of code repeatedly for each set of inputs.



FUNCTIONS



FUNCTION CALL

- An expression that uses a function

```
1 x = input('Enter a number for x: ')
2 y = input('Enter a number for y: ')
3
4 factx = getfactorial(x)
5 print('The factorial of x is', factx)
6 facty = getfactorial(y)
7 print('The factorial of x is', facty)
```

`factx = getfactorial(x)`

- `x` is called the argument
- `factx` will hold the value of the returned by the function



FLOW OF EXECUTION

- A general rule is to define the functions before you call them
- Python scripts are executed from top to bottom, but function definitions do not alter the flow of execution.
 - The statements within the function body are only executed during function calls
 - Think of functions as a detour in the flow of execution; but once the function is finished, the flow of execution returns to where it left off.



PYTHON BUILT-IN FUNCTIONS

- `x = input()`
- `X = input("Enter a value for x:")`
- `xtype = type(x)`
- `print("Hello Sarah")`



INTRODUCTION TO
COMPUTER PROGRAMMING

Functional Programming and Python Libraries



Dictionary: Reverse Lookup

- Given a dictionary d and a key k , the value associated to a key in the dictionary is $v = d[k]$
- **What if you want to find k ?**



Dictionary: Reverse Lookup

- Given a dictionary `d` and a key `k`, the value associated to a key in the dictionary is `v = d[k]`
- **What if you want to find `k`?**
- PROGRAMMING ACTIVITY:
Write a function called `reverse_lookup()` that will return a list of keys that are mapped to a value `v`. Return an empty list if there are no keys mapped to `v`.



Python Libraries

- In programming, a **library** is a collection of precompiled codes with some well-defined operations that can be used in your program.
- The Python library has a collection of code that can be used repeatedly in many programs.
- Python libraries play a role in Machine Learning, Data Science and Data Visualization



Python **math** library

```
# Importing math library
import math

A = 16
print (math.sqrt (A) )
```

- The methods are the functions available inside the library
- List of methods in the math library:
https://www.w3schools.com/python/module_math.asp



Practice Exercise

- Write a function that will accept the 3 lengths of a triangle and determine if the triangle is a right triangle.
 - Print “Right Triangle if a right triangle is formed
 - If the triangles don’t form a right triangle, print the list of angles arranged from the angle opposite the longest side to the shortest side of the triangle.

Hint: Import the `math` library of python so you can use `math.sin()` and `math.cos()`

Hint: To find the sine of degrees, it convert the degrees to radians first using the `math.radians()`



Hint : Right Triangle?

- A triangle with sides a, b and c follows the Pythagorean theorem, which states that:

$$c^2 = a^2 + b^2$$

- To solve a triangle given three sides:
 1. Calculate 2 angles in the triangle using Cosine Law
 2. Calculate the 3rd missing angle. Note that the sum of angles in a triangle is 180 degrees.

$$\cos(C) = \frac{a^2 + b^2 - c^2}{2ab}$$

$$\cos(A) = \frac{b^2 + c^2 - a^2}{2bc}$$

$$\cos(B) = \frac{c^2 + a^2 - b^2}{2ca}$$



INTRODUCTION TO
COMPUTER PROGRAMMING

Dictionaries



Dictionaries in Python

- A dictionary is a built-in data type in Python.
- It is widely used due to its efficiency and easy implementation.

A collection of indexes, called keys, and their corresponding values

- In a dictionary, each key is associated with a single value. This association is called a **key-value** pair and corresponds to one item in the dictionary.



Dictionaries in Python

- In a dictionary, each key is associated with a single value. This association is called a **key-value** pair and corresponds to one item in the dictionary.

$$\{key_1: value_1, key_2: value_2, key_3: value_3 \dots, key_n: value_n\}$$

where $key_1, key_2, \dots, key_n$ are the keys of any immutable data type (ex. string, integer, tuple) and $value_1, value_2, \dots, value_n$ are value that can be of any data type.



Accessing Dictionary Contents

- To values of items in the dictionary can be accessed using the key enclosed in brackets.

```
EngTag = {'one': 'isa', 'two': 'dalawa', 'three': 'tatlo'}  
EngTag['two']  
>> 'dalawa'
```



Example

? Create a dictionary that stores the population of the following countries in southeast Asia.#

Countries in South-Eastern Asia

Country	Population (2020)
<u>Indonesia</u>	273,523,615
<u>Philippines</u>	109,581,078
<u>Vietnam</u>	97,338,579
<u>Thailand</u>	69,799,978



Solution

```
# METHOD 1. Tedious and requires multiple lines
Countries = dict()
Countries['Indonesia'] = 273523615
Countries['Philippines'] = 109581078
Countries['Vietnam'] = 97338579
Countries['Thailand'] = 69799978
Countries['Myanmar'] = 54409800
```

```
# METHOD 2. Create a list of keys and list of values then use zip() function
country = ['Indonesia', 'Philippines', 'Vietnam', 'Thailand', 'Myanmar']
population = [273523615, 109581078, 97338579, 6979997, 54409800]
# to convert lists to dictionary
Countries = dict(zip(country, population))
Countries.keys()
Countries.values()
```



Dictionary Methods

- **len()** function
- **dictionary.keys()** and **dictionary.values()**
- **in** operator

```
myDictionary = {"zero": "apple", "one": "banana", "two": "grape"}  
  
"two" in myDictionary    # True  
"three" in myDictionary  # False
```

Self Test: How will you check if a value is contained in the dictionary?



Traversing a Dictionary

Recall: How do you traverse the elements in a list?

- The **for** statement can be used to traverse the keys of a dictionary.

```
myDictionary = {"zero": "apple", "one": "banana", "two": "grape"}  
  
for key in myDictionary:  
    print(key, myDictionary[key])
```

