MODULE 02

DATA SCIENCE BOOTCAMP

String Methods & Control Flow

Vertical Institute

Reminder

1. Please turn on Zoom camera the whole duration of classes.

2. At the start of all classes, please rename yourselves to: Name + Last 3 digits and letter of your NRIC. Example: John Tan (123A)

Agenda

- Attendance and Photo Taking
- String Indexing
- String Functions and Formatting
- Conditional Statements
- Loops
- Functions
- Recap, Attendance and Photo Taking

Attendance Photo Taking

MODULE 2: STRING METHODS & CONTROL FLOW

String Methods

What Are String Methods?

A Python string is a sequence of Unicode characters that is enclosed in the quotation marks.

Python has a set of built-in methods that you can use on strings

- split()
- lower()
- upper()
- isnumeric()
- join()
- len()

MODULE 2: STRING METHODS & CONTROL FLOW

String Indexing

String Indexing

Python, strings are ordered sequences of character data. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [ ] .

Declare a string variable

>>> s = ‘HELLO WORLD’

H	E	L	L	O		W	O	R	L	D
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String Indexing

In Python, strings are ordered sequences of character data. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [ ] .

Retrieve the first letter

>>>s[0] or s[-11]

‘H’

H	E	L	L	O		W	O	R	L	D
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String Indexing

In Python, strings are ordered sequences of character data. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [ ] .

Retrieve the letter ‘w’

>>>s[6] or s[-5]

’W’

H	E	L	L	O		W	O	R	L	D
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String Indexing

In Python, strings are ordered sequences of character data. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [ ] .

Retrieve 'Hello' by using [start_index:end_index:step_size]

Take note that at the end_index is not inclusive

>>>s[0:5] or s[-11:-6]

'HELLO'

H	E	L	L	O		W	O	R	L	D
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String Indexing

In Python, strings are ordered sequences of character data. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [ ] .

Retrieve alternate character starting with ‘H’ by using [start_index:end_index:step_size]

>>>s[0:6:2] or s[-11:-6:2]

‘HLO’

H	E	L	L	O		W	O	R	L	D
0	1	2	3	4	5	6	7	8	9	10
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

MODULE 2: STRING METHODS & CONTROL FLOW

String Functions and Formatting

String Concatenation

A common task you'll need to accomplish that involves merging or combining string.

```
>>> a = 'hello'
>>> b = 'world'
>>> a + b
'Helloworld'
```

Note: You cannot concatenate a string and an integer

String Splitting and Joining

.split() : Splits a string in a list where each word is a list item

```
>>> txt = 'I love Python'
>>> x = txt.split()
['I','love','Python']
```

.join() : Returns a string by joining all the elements of an iterable (list, string, tuple), separated by a string separator

```
split_txt = ['Python', 'is', 'fun']
>>> ' '.join(split_txt)
'Python is fun'
```


F-Strings

Also known as ‘formatted string literals’, f-strings are string literals that have an f at the beginning and curly braces containing expressions that will be replaced with their values

```
>>> name = “John”
>>> age = 35
>>> f”Hello, {name}. You are {age}.”

“Hello, John. You are 35.”
```


MODULE 2: STRING METHODS & CONTROL FLOW

Python Iterations, Control Flow, and Functions

MODULE 2: STRING METHODS & CONTROL FLOW

Conditional Statements

Why Do We Use Conditional Statements

Most of the simple codes executed so far consist of **sequential execution**, in which statements are always performed one after the next, in exact order specified.

However, the world is often more complicated than that. Most of the time, a program needs to skip over some statements, execute a series of statements repetitively, or choose between alternate sets of statements to execute.

This is where the **control structures** come in. A control structure directs the order of execution of the statements in a program (referred to as the program's **control flow**)

Python If Statements

If...else

Used for decision-making operations. It contains a body of code which runs when condition given in the if statement is true. If the condition is false, the optional else statement will then run.

Syntax	Example

If expression:
  statement(s)
else:
  statement(s)
	a = 10
	if a > 5:
	  print(“This is more than 5”)
	else:
	  print(“This is less than 5”)
	
	‘This is more than 5’

Python If Statements

If...elif...else

Allows you to check multiple expressions for True and executes a block of code as soon as one of the condition is evaluates to True

Syntax	Example

If expression:
  statement(s)
elif:
  statement(s)
else:
  statement(s)
	a = 10
	if a > 15:
	  print(“This is more than 15”)
	elif a > 9 :
	  print(“This is more than 9”)
	else:
	  print(“This is less than 5”)
	‘This is more than 9’

MODULE 2: STRING METHODS & CONTROL FLOW

Loops

For Loops

A for loop is used for iterating over a sequence (that is either a list, tuple, dictionary, set or a string. Loops are important in important as they **execute a block of code repeatedly.**

banks = [‘DBS’, ‘OCBC’, UOB’]

for x in banks:

print(x) Output: ‘DBS’

‘OCBC’

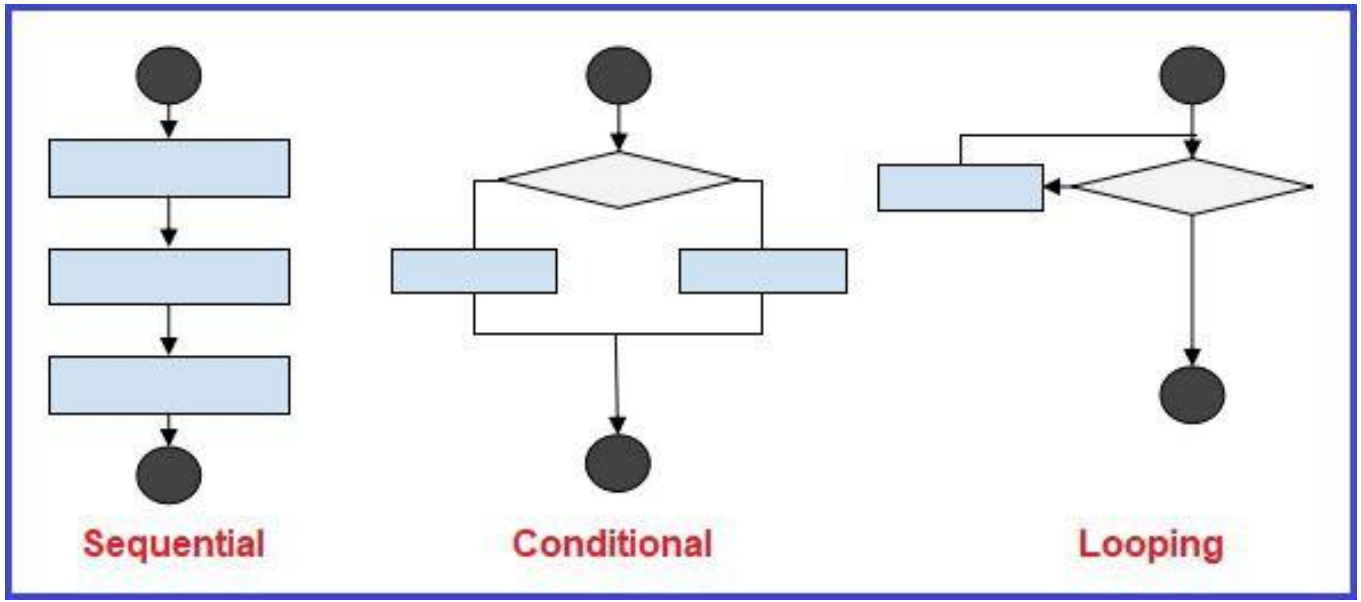
‘UOB’

Range ( )

- Returns the sequence of the given number between the given range
- When user call range() with one argument, user will get a series of numbers that starts at 0
  and includes every whole number up to, but not including, the number that user have provided as the stop.
- Three ways you can call range():

Expression	Output
for i in range(5): print(i)	0 1 2 3 4
for i in range(1,5): print(i)	1 2 3 4
for i in range(0, 30, 3): print(i)	0 3 6 9 12 15 18 21 24 27

Summary of the control flows



MODULE 2: STRING METHODS & CONTROL FLOW

Functions

Python Functions

A function is a set of statements that takes input, do some specific computation and produce output.

The main idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs.

Types of Python Functions

Built-in functions: The built-in functions are functions that are already pre-defined in Python

User-defined functions: The user-defined functions are those defined by the user to perform a specific task

In this section, we will be focusing on building a user-defined function

Advantages of Using Functions

These are the following advantages of Python functions:

- Using functions, we can avoid rewriting the same logic/code again and again
- We can call Python functions multiple times in a program and anywhere in a program
- We can track a large Python program easily when it is divided into multiple functions
- Reusability is the main achievement of Python functions
- However, function calling is always an overhead in a python program

Steps for User-Defined Functions (UDF)

You can use functions to bundle a set of programming instructions that you want to use repeatedly and able to call when needed.

Steps to defining a function:

1. Use the keyword **def** to declare the function and follow this up with the function name.
2. Add parameters to the function: they should be within the parentheses of the function. End your line with a colon.
3. Add statements that the functions should execute. Be aware of the indentation.
4. End your function with a return statement if the function should output something. Without the return statement, your function will return an object None.

User-Defined Functions (UDF)

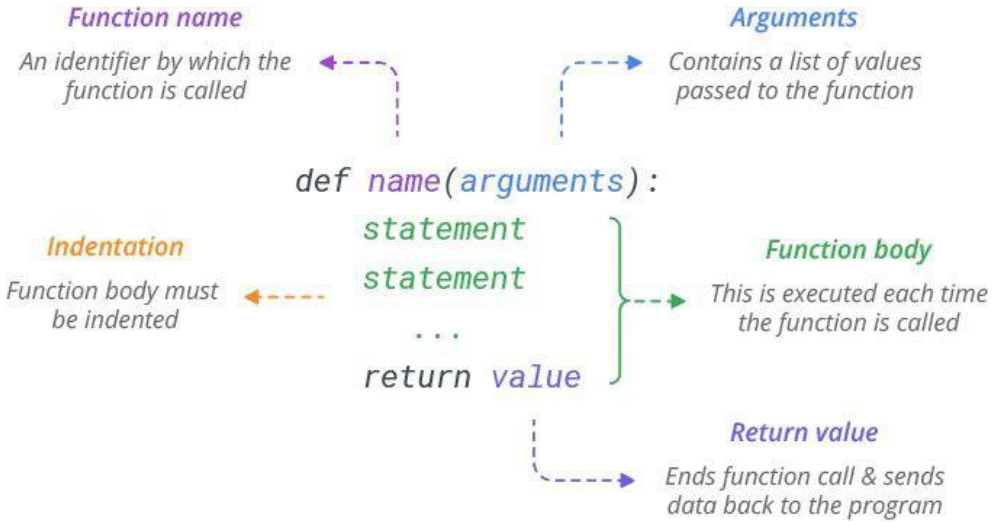
Defining summation function. def sum(a,b):

return a + b

Calling the function you defined.

>>> sum(1,5)

Output 6



Recap time!

What are your favorite takeaways on String Methods & Control Flow?

Let's share with each other!

Some things to take note...

Link and resource could be accessed in the Learning Portal.

https://elearn.verticalinstitute.com/users/sign_in

Attendance Photo Taking

MODULE 02

DATA SCIENCE BOOTCAMP

Thank you!

Vertical Institute