

# Python

## Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

### Recap:

1. Arithmetic Operators
  - a. / (Division)
  - b. % (Modulus)
  - c. // (Floor Division)
  - d. +=, -=, \*=, /=, %=, \*\*= (Operator AND)

### Learning Outcomes:

1. String Slicing
  - a. Start:Stop:Step
  - b. Positive and Negative indexing
  - c. Inverting string

### Explanation Points:

- String Slicing
  - Highlight how a string is a list of individual characters
  - The first character of the string will be index 0
  - Highlight the full syntax in the format: [starting index: ending index: step]
  - Very important to note the full syntax, and get the student to understand the various steps (What does start, stop and step mean?)

### Breakdown of Lesson Plan:

Recap Lesson 3 Quiz <ul style="list-style-type: none"><li>• No help provided. Student must write answer first before being allowed to try typing on computer</li><li>• Student should score at least 75%</li></ul>	20 min (Test) 5 min (Explanation)
Lesson 4.1 (String Index)	10 min
Lesson 4.2 (String Slice Syntax - Positive)	15 min
Lesson 4.3 (String Slice Syntax - Negative)	15 min
Lesson 4.4 (String Slice – Advanced)	15 min
Lesson 4 Quiz	10 min

## Lesson 4

# Python

Module 1 Lesson 4

Name:

Date:



## Learning Objective

Lesson Number	Topics Covered
Quiz	Recap Quiz
4.1	String Slicing – Introduction
4.2	String Slicing – Step
4.3	String Slicing – Negative Index
4.4	String Slicing – Inverting
4.5	String Slicing – Practice
Quiz	End of Lesson Quiz
4 - Extra	More String Slicing

In this lesson, we will be observing how String Slicing and Indexes work in Python.

# Python

Module 1 Lesson 4



Name:

Date:

## Recap Lesson 3 Quiz

### Question 1

Use python to work the equation  $(10 + 1.6)$  to get to the answer in the output column

Output

<b>1</b>	
<b>2</b>	
<b>3</b>	

<b>1</b>	The answer is 11.6
<b>2</b>	<class 'float'>

### Question 2

Use python to work the equation  $(10 * 2 + 5 // 2)$ . Display the output in the format shown below.

Output

<b>1</b>	
<b>2</b>	
<b>3</b>	

<b>1</b>	The answer is
----------	---------------

### Question 3

Use python to work the equation  $(20 \% 4 * 2 + 6 + 3 * 5)$ . Display the output in the format shown below.

Output

<b>1</b>	
<b>2</b>	
<b>3</b>	

<b>1</b>	The answer is
----------	---------------

### Question 4

Name a variable cow and assign "moo" to it. Using multiplication, display the output shown below

Output

<b>1</b>	
<b>2</b>	
<b>3</b>	

<b>1</b>	moomoomoo
----------	-----------

## Recap Lesson 3 Quiz

# Python

Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

## Question 5

Use python string concatenation, name a variable name and assign "Jack" to it. Display the output shown below.

Output

1	
2	
3	

1	My name is Jack
---	-----------------

## Question 6

Mary has 300 sweets. She wishes to separate them into bags of sweets, where each bag contains 12 sweets. How many bags does she need, and how many sweets are left?

Print the answer of in the format of "Mary needs \_\_\_\_ bags. She has \_\_\_\_ sweets remaining."

*HINT: Create a variable called Sweets, assign the value 300 to it. Create a variable called Bags, assign the value of Sweets floor divided by 12 to it. Create a variable called Remaining, assign the value of Sweets modulo 12 to it. Using the variables created, print the answer in the following format*

## Question 7

Follow the following instructions carefully. Think of a number from 1 to 9. Set the value of the variable, x, as the number you are thinking of. Multiply x by 3, and set it as the new value of x. Add 3 to x, and set it as the new value of x. Next, multiply x by 3, and set it as the new value of x. Divide x by 9, and set the resultant value as the new value of x. Finally print x-1. What is the result?

*HINT: Use assignment operators*

## Question 8

Tom has 6 sweets. James has twice as many sweets as Tom. Jane has 2 less sweets than James. How many more sweets does Jane have as compared to Tom? Print out your answer in the form "Jane has \_\_\_\_ more sweets than Tom."

*HINT: Create a variable called Tom with a value of 6. Create a variable James and assign Tom multiply by 2 to it. Create a variable Jane and assign James minus 2 to it. Create a variable More, assign Jane minus Tom to it. Using the variable More, print the answer in the following format.*

# Python

Module 1 Lesson 4

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## Lesson 4.1

In this lesson, we will be tackling the concept of String Slicing. While not apparently useful, the concept of indexes and slicing will be very important as you learn more about programming in Python.

Here are some common concepts that may be useful before we begin:

1. The length of a string represents the number of characters it contains
2. Each character in a string has an index
3. You can use String Slicing to extract a few characters from a string

To begin, let us understand what an index even is.

Indexing: 1. Index starts from 0 in Python.

2. First element has an index 0, second element has index 1, so on.

3. Negative Index starts from -1

4. To access an element by its index we need to use square brackets

Hello

0 1 2 3 4  
-5 -4 -3 -2 -1

Therefore, the first term in a string has an index of 0, and each subsequent term has an index that increases by 1. Using this, we can select the specific indexes from a string to print a completely new string.

To do string slicing, there is a certain format we need to follow, which is as such:

(variable) [start:stop:step]

For example:

Output

1	String = "I Love Coding"
2	print(String[2:6])

1	Love
---	------

We begin to understand string slicing by recognizing where to [start].

The number represented by [start] refers to the index of the string where we begin.

0	1	2	3	4	5	6	7	8	9	10	11	12
I		L	o	v	e		C	o	d	i	n	g

In the example above, we started with the index 2, which is represented by L.

On the other hand, the [stop] refers to the index where we stop at (non-inclusive). For the example above, we stopped at 6, which is a spacing. Therefore, our output would read "Love".

# Python

## Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

### Lesson 4.1

Type the following and fill in the output after pressing F5

*\*The numbers represent the lines.*

**For the following questions, the indexes below may be useful.**

0	1	2	3	4	5	6	7	8	9	10	11	12
I		L	o	v	e		C	o	d	i	n	g

Task 1:

Output

1	String = "I Love Coding"
2	print(String[0])

1	
---	--

Task 2:

Output

1	String = "I Love Coding"
2	print(String[5])

1	
---	--

Task 3:

Output

1	String = "I Love Coding"
2	print(String[0:5])

1	
---	--

Task 4:

Output

1	String = "I Love Coding"
2	print(String[7:12])

1	
---	--

**An interesting thing to note is we can also choose to leave some of the terms blank. Let us look at some examples:**

Task 5:

Output

1	String = "I Love Coding"
2	print(String[:7])

1	
---	--

Task 6:

Output

1	String = "I Love Coding"
2	print(String[7:])

1	
---	--

# Python

Module 1 Lesson 4

Name:

Date:



## Lesson 4.2

### Recap:

Indexing: 1. Index starts from 0 in Python.

2. First element has an index 0, second element has index 1, so on.

3. Negative Index starts from -1

4. To access an element by its index we need to use square brackets

Hello

0 1 2 3 4  
-5 -4 -3 -2 -1

String Slicing Format:

(variable) [start:stop:step]

Moving along, sometimes we want to skip some indexes. To do this, we use the [step] portion of the String Slicing.

For example:

Output

1	String = "Logic Is Life"
2	print(String[2:9:2])

1	gcl
---	-----

To understand how to [step], we first need to remember what the [start:stop] provides.

0	1	2	3	4	5	6	7	8	9	10	11	12
L	o	g	i	c		l	s		L	i	f	e

In the example above, we started with the index 2 and stopped at the index 9. Therefore,

2	print(String[2:9])
---	--------------------

Would have provided an answer of "gic ls".

With this in mind, we can look at [step] as jumping indexes by a certain value. In this example, we are jumping every 2 indexes. Therefore, we are only printing indexes 2, 4, 6 and 8. This prints the letters:

"gcl"

Let us try some examples.

# Python

Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

## Lesson 4.2

Type the following and fill in the output after pressing F5

*\*The numbers represent the lines.*

**For the following questions, the indexes below may be useful.**

0	1	2	3	4	5	6	7	8	9	10	11	12
L	o	g	i	c		l	s		L	i	f	e

Task 1:

Output

1	String = "Logic Is Life"
2	print(String[0:12:2])

1	
---	--

Task 2:

Output

1	String = "Logic Is Life"
2	print(String[0:12:3])

1	
---	--

Task 3:

Output

1	String = "Logic Is Life"
2	print(String[2:10:2])

1	
---	--

Task 4:

Output

1	String = "Logic Is Life"
2	print(String[4:13:2])

1	
---	--

Task 5:

Output

1	String = "Logic Is Life"
2	print(String[:12:3])

1	
---	--

Task 6:

Output

1	String = "Logic Is Life"
2	print(String[4::2])

1	
---	--



# Python

Module 1 Lesson 4

Name:

Date:



## Lesson 4.3

### Recap:

Indexing: 1. Index starts from 0 in Python.

2. First element has an index 0, second element has index 1, so on.

3. Negative Index starts from -1

4. To access an element by its index we need to use square brackets

Hello

0 1 2 3 4  
-5 -4 -3 -2 -1

String Slicing Format:

(variable) [start:stop:step]

What happens if the string is too long? Sometimes, if we want to get the last index, but we have difficulties finding how long a string is, Negative Indexing will help!

For example:

### Output

1	String = "Coding Is Easy!"
2	print(String[-5:-1])

1	Easy
---	------

To understand how to [step], we first need to remember what the [start:stop] provides.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
C	o	d	i	n	g		l	s		E	a	s	y	!

Unlike positive indexing, negative indexes begins from the right, and starts from -1. In the example above, we begin with index -5, which is an "E". We stop at index -1, which is the "!". Therefore, we will print "Easy".

When doing string slicing, we can do a mix of both positive and negative indexing to get the desired string.

# Python

## Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

### Lesson 4.3

Type the following and fill in the output after pressing F5

*\*The numbers represent the lines.*

**For the following questions, the indexes below may be useful.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
C	o	d	i	n	g		l	s		E	a	s	y	!

Task 1:

Output

1	String = "Coding Is Easy!"
2	print(String[0:-1])

1	
---	--

Task 2:

Output

1	String = "Coding Is Easy!"
2	print(String[-8:-1])

1	
---	--

Task 3:

Output

1	String = "Coding Is Easy!"
2	print(String[-7:-9])

1	
---	--

Task 4:

Output

1	String = "Coding Is Easy!"
2	print(String[-1:-6])

1	
---	--

Task 5:

Output

1	String = "Coding Is Easy!"
2	print(String[::-1])

1	
---	--

Task 6:

Output

1	String = "Coding Is Easy!"
2	print(d[-7:])

1	
---	--

### Lesson 4.4

# Python

Module 1 Lesson 4

Name:

Date:



## Recap:

Indexing: 1. Index starts from 0 in Python.

2. First element has an index 0, second element has index 1, so on.

3. Negative Index starts from -1

4. To access an element by its index we need to use square brackets

Hello

0 1 2 3 4  
-5 -4 -3 -2 -1

String Slicing Format:

(variable) [start:stop:step]

The last concept of String Slicing we will cover is inverting a string. We have evaluated what happens if we use negative indexing in the [start:stop] areas. What happens if our [step] is negative?

For example:

Output

1	String = "I Love Coding
2	print(String[-1:0:-1])

1	!gnidoC evol
---	--------------

Using negative number in [step] gets much trickier to understand. To begin, let us see where the [start] and [stop] bring us to.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
I		L	o	v	e		C	o	d	i	n	g	!

From the example, we start at -1 and stop at 0. This should be new to you, since most of the times we start at the left of the stop. However, this time our start happens to be on the right of our stop. Do not panic, you are still on the right track.

When our [step] is negative, you can interpret it like that of a number line, whereby we move between the terms in a negative direction. This means we begin from -1, and increase by -1 all the way until we reach 0.

Therefore, we will get:

"!gnidoC evol "

## Lesson 4.4

Type the following and fill in the output after pressing F5

# Python

## Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

*\*The numbers represent the lines.*

**For the following questions, the indexes below may be useful.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13
-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
I		L	o	v	e		C	o	d	i	n	g	!

Task 1:

Output

1	String = "I Love Coding!"
2	print(String[-1:-7:-1])

1	
---	--

Task 2:

Output

1	String = "I Love Coding!"
2	print(String[-1:-6:-1])

1	
---	--

Task 3:

Output

1	String = "I Love Coding!"
2	print(String[-1::-1])

1	
---	--

Task 4:

Output

1	String = "I Love Coding!"
2	print(String[-1::-2])

1	
---	--

Task 5:

Output

1	String = "I Love Coding!"
2	print(String[5:0:-1])

1	
---	--

# Python

Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

## Lesson 4.5

Let us practice what we have learnt. For these tasks, there will be no index provided. You should try making your own indexes if needed.

Task 1:

1	String = "The Logic Coders"
2	print(String[4:-5])

Output

1	
---	--

Task 2:

1	String = "The Logic Coders"
2	print(String[2:-2])

Output

1	
---	--

Task 3:

1	String = "The Logic Coders"
2	print(String[:])

Output

1	
---	--

Task 4:

1	String = "The Logic Coders"
2	print(String[::-1])

Output

1	
---	--

Task 5:

1	String = "The Logic Coders"
2	print(String[4:7])

Output

1	
---	--

Task 6:

1	String = "The Logic Coders"
2	print(String[0:3])

Output

1	
---	--

# Python

Module 1 Lesson 4

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## Lesson 4.5

**For the following questions, the output is provided. Try to find the correct String Slicing format to get these outputs.**

Task 7:

1	String = "The Logic Coders"
2	print(String[

Output

1	Logic
---	-------

Task 8:

1	String = "The Logic Coders"
2	print(String[

Output

1	The Logic
---	-----------

Task 9:

1	String = "The Logic Coders"
2	print(String[

Output

1	TeLgc
---	-------

Task 10:

1	String = "The Logic Coders"
2	print(String[

Output

1	Coders
---	--------

Task 11:

1	String = "The Logic Coders"
2	print(String[

Output

1	Ce
---	----

### EXTRA:

**Your teacher may give you some challenges related to String Slicing! See if you can solve them!**

# Python

Module 1 Lesson 4



Name: \_\_\_\_\_

Date: \_\_\_\_\_

## End of Lesson 4 Quiz

Question 1:

Use python string slicing and the variable *Name*, come up with a program to print "GIKM".

Name = "ABCDEFGHJKLMNOPQRS"

*HINT: Find pattern. Use [start:stop:step]*

Question 2:

Use string slicing to get the first 5 characters of the string and last 5 characters of the string

Alphabet = "ABCDEFGHJKLMNOP"

Print "The first 5 letters are \_\_\_\_"

Print "The last 5 letters are \_\_\_\_"

*HINT: For first 5 letters use zero-indexing. For last 5 letters use negative indexing.*

Question 3:

Define variable **String** as "Onomatopoeia". Using string slicing, print "pea".

Question 4:

Use string slicing with negative indexing to print "dangerous" from the string named

year2020= 'virus is dangerous'

Question 5:

Print "pine" from the variable x shown below.

x="porcupine"

Question 6:

Print "vert" from the variable y shown below.

y='Inverted Words'

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## End of Lesson 4 Quiz

**For the following questions, select the best answer:**

### Question 7

Which of the following represents the String Slicing Format?

A: [Start:Step:Stop]

B: [Stop:Start:Step]

C: [Start:Stop:Step]

D: [Step:Start:Stop]

### Question 8

Which of the following format uses Negative Slicing?

A: String[0:10]

B: String[-6:-1]

C: String[:7]

D: String[7:]

### Question 9

When talking about positive indexes, what is the index of the first letter in a string?

A: 0

B: 1

C: -1

D: String.FirstIndex()

### Question 10

What is the last index of a string?

A: 0

B: 1

C: -1

D: 15



# Python

Module 1 Lesson 4

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## Lesson 4 EXTRA

Here are some extra problems that highlight interesting things about String Slicing. See if you can figure out why the answers are as such:

Task 1:

What happens if our start goes after our stop?

1	String = "The Logic Coders"
2	print(String[9:3])

Output

1	
---	--

Task 2:

What happens if we place 2 string slicing back-to-back?

1	String = "The Logic Coders"
2	print(String[4:9][0])

Output

1	
---	--

Task 3:

What happens if we place 2 string slicing back-to-back?

1	String = "The Logic Coders"
2	print(String[4:9][0:2])

Output

1	
---	--

Task 4:

What happens if we place 2 string slicing back-to-back?

1	String = "The Logic Coders"
2	print(String[4:9][::-1])

Output

1	
---	--

# Python

Module 1 Lesson 4

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## Lesson 4 EXTRA

Task 5:

What if we stop outside our maximum String Index?

1	String = "The Logic Coders"
2	print(String[0:10000])

Output

1	
---	--

Task 6:

What if our step is outside our maximum String Index?

1	String = "The Logic Coders"
2	print(String[0:10000])

Output

1	
---	--

Task 7:

What happens if our start goes before our stop when our step is negative?

1	String = "The Logic Coders"
2	print(String[0:5:-1])

Output

1	
---	--

Task 8:

What happens if our start is outside our maximum String Index?

1	String = "The Logic Coders"
2	print(String[1000])

Output

1	
---	--