

Python

Module 1 Lesson 2



Name: _____

Date: _____

Recap:

1. Variables
 - a. Variables and Naming Convention
 - b. Integers
 - c. Strings

Learning Outcomes:

1. Strings
 - a. Concatenation
 - b. Strings + Integers
2. Arithmetic Operators
 - a. + (Addition of Integers and Strings)
 - b. - (Subtraction of Integers)
 - c. * (Multiplication of Integers)

Explanation Points:

- Strings and Integers cannot be added together
 - Either fully string or fully integers. Show the difference between the two
- Changing Strings and Integers using str and int
 - Show what errors are formed if Python cannot change
- Recognizing Strings
 - Highlight the importance of "" and " in recognizing Strings
- Intermediate Variables
 - Show how variables can be made of equations of other variables.

Breakdown of Lesson Plan:

Recap Lesson 1 Quiz <ul style="list-style-type: none">No help provided. Student must write answer first before being allowed to try typing on computerStudent should score at least 75%	15 min (Test) 5 min (Explanation)
Lesson 2.1 (Concatenation)	10 min
Lesson 2.2 (Concatenation)	10 min
Lesson 2.3 (Addition)	15 min
Lesson 2.4 (Subtraction)	15 min
Lesson 2.5 (Multiplication)	10 min
Lesson 2 Quiz	10 min

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2

Learning Objective

Lesson Number	Topics Covered
Quiz	Recap Quiz
2.1	String Concatenation
2.2	Addition
2.3	Subtraction
2.4	Multiplication
2.5	Division
Quiz	End of Lesson Quiz

In this lesson, we will be learning how to print our answers in a legible format, as well as observe the various Arithmetic Operators commonly used in Programming.

Python

Module 1 Lesson 2



Name:

Date:

Recap Lesson 1 Quiz

Question 1

Create 2 variables named `mark1` and `mark2`. Assign the values 500 and 300 to them. Print the values of `mark1` and `mark2`.

Output

1	
2	
3	
4	

1	
2	
3	
4	

Question 2

Create variables named `john` and `jane`. Assign the value 50 and 100 to them. Print the value of your variables to get an output of 50.

Output

1	
2	
3	
4	

1	50
---	----

Question 3

Create a variable named `age` and assign the value 50. Print the datatype of `age`.

Output

1	
2	

1	
---	--

Question 4

Create a variable named `moneyface` and assign the value "100" to it. Print `moneyface`.

Output

1	
2	

1	
2	

Python

Module 1 Lesson 2

Name:

Date:



Recap Lesson 1 Quiz

Question 5

Fill in the column with "Allowed" or "Not Allowed"

		Is it allowed?
1	Jake	
2	class	
3	elif	
4	Break	
5	John	
6	Numeroftimes	
7	except	
8	howmuch	
9	eturn	
10	returnme	
11	correctme	
12	difference	
13	dylan	
14	amidone	

Question 6:

Which of the follow are **not** datatypes that you have learnt?

A: Integers

B: Boolean

C: Float

D: String

Question 7:

Strings are easy to recognize because they -

A: contain " or ""

B: have decimal places

C: contains numbers

D: contain only alphabets

Question 8:

Which of the following is a float?

A: "7.0"

B: 9

C: 11

D: 3.0

Python

Module 1 Lesson 2

Name: _____

Date: _____



Recap Lesson 1 Quiz

Question 9A:

Katty wants to write a simple program to wish her friend Happy Birthday. She wants to print the following words:

“Happy Birthday Amy!”

What datatype should she use to store this value? Why?

--

Question 9B

Try writing a program such that the output is as shown!!

1	
2	
3	
4	

Output:

1	Happy Birthday Amy!
2	<class 'str'>

HINT: Remember the type() function will show the datatype of the variable.

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.1

At times, it is important for us as programmers to output a statement that makes sense. Imagine, as a user, you run a program only to face an output that looks like:

1	55000
2	29000

What exactly does 55000 and 29000 even mean?

It would be useful if we are able to add some context into the statements, for example:

1	Profit Goal : 55000 USD
2	Current Profits : 29000 USD

Now it is much clearer what these 2 integers are supposed to represent.

However, the sharp eyed student would notice something. In order to print this, we need to have the statement as a string! But how do we do that?

Therefore, in today's lesson, let us try learning how to create outputs that makes sense. The method to do so is called String Concatenation.

1. Concatenation (short cut method) – Use “,” symbol
 - a. To join two strings together or 1 string and 1 integer, you can utilize the “,” symbol.
2. Concatenation – Use “+” symbol
 - a. To join two strings together by utilizing the “+” symbol.
 - b. To join one string and one integer, you can utilize the “+” symbol.

A very useful tool to help us with String Concatenation is to use type casting.

Type Casting: Converts from one datatype to another. We can convert the integer to string by adding “str” or string to integer by adding “int”. The format is as shown:

`int('7')`

or

`str(59000)`

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.1

Let us begin by comparing the difference between the two String Concatenation methods, using (,) vs using (+). Note: Pay careful attention to the spacing!

Task 1:

Output

1	teacher="is naggy"
2	print("teacher" + teacher)

1	
---	--

Task 2:

Output

1	teacher="is naggy"
2	print("teacher" , teacher)

1	
---	--

What is the difference between Tasks 1 and 2?

--

Task 3:

Output

1	Check = "crazy"
2	print("Am I " + Check + "?")

1	
---	--

Task 4:

Output

1	Check = "crazy"
2	print("Am I" , Check , "?")

1	
---	--

When should we use "+" and when should we use ","?

--

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.1

One downside to using the (+) method to String Concatenation is that both values must be strings. Let us look at this example:

Task 5:

1	Orange = 15
2	print("James ate " + Orange + " Oranges")

What is the error shown in Task 5?

We can work around this kind of situations by using Type Casting! Let us see this in action:

Task 6:

1	Orange = 15
2	print("James ate " + str(Orange) + " Oranges")

Interestingly enough, using (,) does not have such an issue.

Task 7:

1	Orange = 15
2	print("James ate", Orange, "Oranges")

With all these in mind, let us try some simple String Concatenation Practices

Task 8:

Write a program that prints your age in the format:

"I am currently ____ years old."

1	
2	

Task 9:

Jack is 178 cm tall. Write a program that prints his height in the format:

"Jack is ____ cm tall."

1	
2	

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.1

Task 10:

James gets \$10 as pocket money. Write a program that prints his pocket money in the format:

“James receives \$___ as pocket money.”

1	
2	

HINT: How do you print the “\$” sign without spacing?

Task 11:

In some ATMs, the screen shows the balances of the users in the format:

“Current Account: \$100,000”

Let us try replicating this formatting!

1	
2	

HINT: How do you print the “,” sign without spacing?

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.2

While printing lines is useful, Python also allows programmers to do more complex calculations and algorithms. Let us start by learning some simple Arithmetic Operators. In this lesson, we will be looking at the “+” symbol.

Python Arithmetic Operators → Arithmetic Operators are special symbols in Python that carry out arithmetic or logical computation. (e.g. “+”, “-”, “/”, “**”)

Operator	Meaning	Example
+	Add two operands	x + y
-	Subtract right operand from the left	x - y
*	Multiply two operands	x * y
/	Divide left operand by the right one {always results into float (decimal place)}	x / y
%	Modulus - remainder of the division of left operand by the right {always results into integer}	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line (round down) {always results into integer}	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Operator = +

Operands = 2 and 3

Output = 5

1	combine=2 + 3
2	print(combine)

1	5
---	---

Note that when doing calculations, it is safer and neater to utilize variables to help with our calculations. We can compare these two programs to see.

1	Addition = 5 + 7
2	print(Addition)

1	print(5+7)
---	------------

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.2

Task 1:

James runs an orchard that grows apples, pears and peaches. He has a total of 25 apple trees, 31 pear trees and 29 peach trees. Write a simple program that can find the total number of fruit trees James has. Print your answer in the format:

“James has a total of ____ fruit trees.”

1	Apple_Tree=
2	Pear_Tree =
3	Peach_Tree =
4	Total_Tree =
5	
6	print(“James has a total of”, Total_Tree, “fruit trees.”)

Teachers’ Remark:

Some students may ask why do we have to write the entire program like this, when we can easily calculate it using a calculator or by hand? This is because the examples provided here are simple, and are meant to guide you through the thought processes and concepts. When applied in real life, the number of terms would far exceed what can be feasibly done manually. At that point in time, a firm understanding of the basics would make understanding how to write your programs easier.

Task 2:

John has 3 separate saving accounts. In his first account, he has \$140000. In the second account, he has \$256789. In the third account, he has \$53426. Write a simple program that can find the total amount of money John has. Print your answer in the format:

“John has \$ ____ in total.”

1	Account1=
2	Account2=
3	Accoun3=
4	
5	
6	print(

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.2

Task 3:

Patricia wants to buy some presents for Christmas. She is planning to buy a stuffed bear for \$15, a pencil box for \$5 and a Christmas Wreath for \$7. Write a simple program to help her calculate how much she will need in order to buy these 3 presents. Print your answer in the format:

“Patricia will need \$___ in total.”

1	
2	
3	
4	
5	
6	

Task 4:

Mrs Mandy has a total of 6 classes. She is planning to buy a pencil (1) for each of her students. The number of students in each class is shown in the table below. Write a simple program to find out how many pencils she will need in total. Print your answer in the format:

“Mrs Mandy will need ___ pencils in total.”

Class	Student
A	15
B	21
C	24
D	19
E	23
F	20

1	
2	
3	
4	
5	
6	

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.3

The next operator to look at would be the minus “-” symbol.

Python Arithmetic Operators → Arithmetic Operators are special symbols in Python that carry out arithmetic or logical computation. (e.g. “+”, “-”, “/”, “**”)

Operator	Meaning	Example
+	Add two operands	x + y
-	Subtract right operand from the left	x - y
*	Multiply two operands	x * y
/	Divide left operand by the right one {always results into float (decimal place)}	x / y
%	Modulus - remainder of the division of left operand by the right {always results into integer}	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line (round down) {always results into integer}	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Operator = -

Operands = 12 and 2

1	combine=12 - 2
2	print(combine)

Output = 10

1	10
---	----

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.3

Task 1:

James has \$100 in his bank. He withdraws \$5 to pay for lunch, \$20 for a taxi ride and \$13 for a gift. How much does he have left in his bank? Write a simple program to find out! Print your answer in the format:

“James has \$___ left.”

1	Bank = 100
2	Lunch = 5
3	Taxi = 20
4	Gift = 13
5	Bank = Bank – Lunch – Taxi – Gift
6	
7	print(

Why do you think we should overwrite the value in Bank (as seen in line 5?)

Task 2:

Kane has 200 sweets. He gives 50 to Jamie, 15 to Kane and 17 to Tom. Write a simple program to find out how many sweets each person has. Print your answer in the format:

“Kane will have ___ sweets, Jamie will have ___ sweets, Kane will have ___ sweets and Tom will have ___ sweets.”

1	Kane = 200
2	Jamie
3	Kane
4	Tom
5	Kane = Kane
6	
7	print(

HINT: Remember that Jamie, Kane and Tom have their own variables!

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.4

The next Arithmetic Operator we will look at is the multiplication **"**"** symbol. In Python, we do not use **x** to represent multiplication, as **x** is also a letter. Therefore, we use the **"**"** symbol.

Python Arithmetic Operators → Arithmetic Operators are special symbols in Python that carry out arithmetic or logical computation. (e.g. **"+"**, **"-"**, **"/"**, **"**"**)

Operator	Meaning	Example
+	Add two operands	x + y
-	Subtract right operand from the left	x - y
*	Multiply two operands	x * y
/	Divide left operand by the right one {always results into float (decimal place)}	x / y
%	Modulus - remainder of the division of left operand by the right {always results into integer}	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line (round down) {always results into integer}	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Operator = *

Operands = 2 and 3

Output = 6

1	combine=2 * 3
2	print(combine)

1	6
---	---

In this lesson, we will also take a look at how we can multiply strings too.

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.4

Task 1:

Tom estimates his daily expenditure to be \$15 each day. He wants to estimate his spending over 5 days. Write a simple program to find his expenditure over 5 days. Print your answer in the format:

“Tom would have spent \$___ over 5 days.”

1	Daily_spent = 15
2	Days = 5
3	Total_spent =
4	
5	print(

This question can also be answered using Addition. However, Multiplication is better in this case. Why?

--

Task2:

Another interesting thing we can do with multiplication is to multiply a string!

1	String = "Coding"
2	print(String)
3	print(String * 5)

Run the code above and find out the output!

Output:

1	
2	

Task 3:

Now try it yourself! Print the following output and show it to your teacher!

Output:

1	MorningMorningMorning
---	-----------------------

Python

Module 1 Lesson 2

Name: _____

Date: _____



Lesson 2.5

The last operator we will be looking at is the division symbol `"/"`. Do note that the division operator always gives a float.

Python Arithmetic Operators → Arithmetic Operators are special symbols in Python that carry out arithmetic or logical computation. (e.g. `"+"`, `"-"`, `"/"`, `"*"`)

Integer – Whole number

Float – Number with decimals

Operator	Meaning	Example
<code>+</code>	Add two operands	<code>x + y</code>
<code>-</code>	Subtract right operand from the left	<code>x - y</code>
<code>*</code>	Multiply two operands	<code>x * y</code>
<code>/</code>	Divide left operand by the right one {always results into float (decimal place)}	<code>x / y</code>
<code>%</code>	Modulus - remainder of the division of left operand by the right	<code>x % y</code> (remainder of <code>x/y</code>)
<code>//</code>	Floor division - division that results into whole number adjusted to the left in the number line (round down)	<code>x // y</code>
<code>**</code>	Exponent - left operand raised to the power of right	<code>x**y</code> (x to the power y)

Arithmetic Operators → Division

Operator = `/`

Operands = 20 and 2

Output = 10.0

1	combine=20 / 2
2	print(combine)

1	10.0
---	------

Python

Module 1 Lesson 2



Name: _____

Date: _____

Lesson 2.5

Task 1:

Teacher Mandy wants to share 500 sweets between 25 students. Write a simple program to find out how many sweets each student gets. Print your answer in the format:

“Each student will get ____ sweets.”

1	Sweets = 500
2	Students = 25
3	Sweets_Per_Student =
4	
5	print(

With the knowledge of these 4 operators, let us try a challenging example:

Task 2

Work out the following equations and print in the format “The answer of *equation* is ____”

Eg. “The answer of $5 * 6$ is 30”

- a. $5 * 6$
- b. $6 * 2 - 3 + 7$
- c. $(2 + 2) - (4 * 3)$
- d. $(2 - 2) + (1 * 10)$
- e. $12 * 12 + (3 - 2)$
- f. $10 * 10 - 5$
- g. $2 + 5 * 2 - 2$

Here is an example of one solution, $21 * 55$

1	print(“The answer of $21 * 55$ is”, $21 * 55$)
---	---

Output:

1	The answer of $21 * 55$ is 1155
---	---------------------------------

Python

Module 1 Lesson 2

Name: _____

Date: _____



End of Lesson 2 Quiz

Question 1

Create 2 variables `number1` and `number2`, assign values of 5 and 10 to the variables. Print the datatype of `number1` and the sum of `number1` and `number2` in the format "The sum is 15"

Output

1	
2	
3	
4	

1	
2	The sum is 15

Question 2

Create 2 variables `height1` and `height2`, assign the values of "35" and '89' to the variables. Print the datatype of `height1` and the sum of `height1` and `height2`.

Output

1	
2	
3	
4	

1	
2	

Question 3

Create 2 variables `collection1` and `collection2`, assign values of 100 and 200 to them. Create a third variable `totalcash` and assign `collection1 + collection2` to it. Print the value of `totalcash`.

Output

1	
2	
3	
4	

1	Total cash is \$300
---	---------------------

Python

Module 1 Lesson 2



Name: _____

Date: _____

End of Lesson 2 Quiz

Question 4

Create 2 variables `totalmoney` and `spending`, assign values of 400 and 100 to them. Create a third variable `broke` and assign `totalmoney - spending` to it. Print the value of `broke` in the format of "I am broke with ____ dollars"

Output

1	
2	
3	
4	

1	
---	--

Question 5

Create 2 variables `monster1`, `monster2`, `monster3` and assign any values to them. Create a third variable `totalmonsters` and assign `monster1 + monster2 + monster3` to it. Print the value of `totalmonsters` in the format of "Total monsters caught is ____"

(Use any values to get to the output)

Output

1	
2	
3	
4	
5	

1	Total monsters caught is
---	--------------------------

Question 6

Create 3 variables `monster1`, `monster2`, and `monster3`, assign "Too", "Many", "Monsters" to them. Create a fourth variable `monsterscaught` with the value `monster1 + monster2 + monster3`. Print the value of `monsterscaught`.

Output

1	
2	
3	
4	
5	

1	
---	--