[Jahangirnagar University](https://www.coursera.org/programs/jahangirnagar-university-na-qyc9d)

[Skip to Main Content](https://www.coursera.org/learn/python-crash-course/exam/In52m/module-2-graded-assessment/attempt?redirectToCover=true#main)

* 

MD. TAZEL HOSSAN

[Crash Course on Python](https://www.coursera.org/learn/python-crash-course/home/welcome)

[Week 2](https://www.coursera.org/learn/python-crash-course/home/week/2)

Module 2 Graded Assessment

[Prev](https://www.coursera.org/learn/python-crash-course/lecture/SeY4Y/in-margas-words-why-i-like-python)

[Next](https://www.coursera.org/learn/python-crash-course/discussionPrompt/LSF3i/basic-python-syntax-review)

**QUIZ • 50 MIN**

**Module 2 Graded Assessment**

**Submit your assignment**

**DUE**Jun 29, 12:59 PM +06

Try again

**Receive grade**

**TO PASS**80% or higher

**Grade**

100%

View Feedback

We keep your highest score

Module 2 Graded Assessment

Graded Quiz • 50 min

**Due** Jun 29, 12:59 PM +06

**Congratulations! You passed!**

**TO PASS**80% or higher

Keep Learning

**GRADE**

100%

**Module 2 Graded Assessment**

**LATEST SUBMISSION GRADE**

100%

1.Question 1

Complete the function by filling in the missing parts. The color\_translator function receives the name of a color, then prints its hexadecimal value. Currently, it only supports the three additive primary colors (red, green, blue), so it returns "unknown" for all other colors.

**1 / 1 point**

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

def color\_translator(color):

    if color == "red":

        hex\_color = "#ff0000"

    elif color == "green":

        hex\_color = "#00ff00"

    elif color == "blue":

        hex\_color = "#0000ff"

    else:

        hex\_color = "unknown"

    return hex\_color

print("big">"small")

print(color\_translator("blue")) # Should be #0000ff

print(color\_translator("yellow")) # Should be unknown

print(color\_translator("red")) # Should be #ff0000

print(color\_translator("black")) # Should be unknown

print(color\_translator("green")) # Should be #00ff00

print(color\_translator("")) # Should be unknown





RunReset

**Correct**

Well done! You're breezing through the if-else clauses!

2.Question 2

What's the value of this Python expression: "big" > "small"

**1 / 1 point**



True



False



big



small

**Correct**

You nailed it! The conditional operator > checks if two values are equal. The result of that operation is a boolean: either True or False. Alphabetically, "big" is less than "small".

3.Question 3

What is the elif keyword used for?

**1 / 1 point**



To mark the end of the if statement



To handle more than two comparison cases



To replace the "or" clause in the if statement



Nothing - it's a misspelling of the else-if keyword

**Correct**

You got it! The elif keyword is used in place of multiple embedded if clauses, when a single if/else structure is not enough.

4.Question 4

Students in a class receive their grades as Pass/Fail. Scores of 60 or more (out of 100) mean that the grade is "Pass". For lower scores, the grade is "Fail". In addition, scores above 95 (not included) are graded as "Top Score". Fill in this function so that it returns the proper grade.

**1 / 1 point**

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

def exam\_grade(score):

    if score>95:

        grade = "Top Score"

    elif score>=60:

        grade = "Pass"

    else:

        grade = "Fail"

    return grade

print(exam\_grade(65)) # Should be Pass

print(exam\_grade(55)) # Should be Fail

print(exam\_grade(60)) # Should be Pass

print(exam\_grade(95)) # Should be Pass

print(exam\_grade(100)) # Should be Top Score

print(exam\_grade(0)) # Should be Fail





RunReset

**Correct**

Good job! You're getting the hang of it!.

5.Question 5

What's the value of this Python expression: 11 % 5?

**1 / 1 point**



2.2



2



1



0

**Correct**

Excellent! "%" is the modulo operator, which returns the remainder of the integer division between two numbers. 11 divided by 5 equals 2 with remainder of 1.

6.Question 6

Complete the body of the ***format\_name*** function. This function receives the ***first\_name*** and ***last\_name***parameters and then returns a properly formatted string.

Specifically:

If both the ***last\_name*** and the ***first\_name*** parameters are supplied, the function should return like so:

1

2

print(format\_name("Ella", "Fitzgerald"))

Name: Fitzgerald, Ella





If only ***one*** name parameter is supplied (either the first name *or* the last name) , the function should return like so:

1

2

print(format\_name("Adele", ""))

Name: Adele





or

1

2

print(format\_name("", "Einstein"))

Name: Einstein





Finally, if both names are blank, the function should return the empty string:

1

2

print(format\_name("", ""))





Implement below:

**1 / 1 point**

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

def format\_name(first\_name, last\_name):

    # code goes here

    if len(first\_name)>0 and len(last\_name)>0:

        string = "Name: "+last\_name +", "+first\_name

    elif (len(first\_name)>0 and len(last\_name)==0):

        string ="Name: "+first\_name

    elif  ( len(first\_name)==0 and len(last\_name)>0):

        string ="Name: "+last\_name

    else:

        string =""

    return string

print(format\_name("Ernest", "Hemingway"))

# Should return the string "Name: Hemingway, Ernest"

print(format\_name("", "Madonna"))

# Should return the string "Name: Madonna"

print(format\_name("Voltaire", ""))

# Should return the string "Name: Voltaire"

print(format\_name("", ""))

# Should return an empty string





RunReset

**Correct**

Awesome! You're getting the hang of the multiple and

embedded "if" clauses!

7.Question 7

The longest\_word function is used to compare 3 words. It should return the word with the most number of characters (and the first in the list when they have the same length). Fill in the blank to make this happen.

**1 / 1 point**

1

2

3

4

5

6

7

8

9

10

11

12

def longest\_word(word1, word2, word3):

    if len(word1) >= len(word2) and len(word1) >= len(word3):

        word = word1

    elif len(word2) >= len(word1) and len(word2) >= len(word3):

        word = word2

    else:

        word = word3

    return(word)

print(longest\_word("chair", "couch", "table"))

print(longest\_word("bed", "bath", "beyond"))

print(longest\_word("laptop", "notebook", "desktop"))





RunReset

**Correct**

You got it! You've figured out how to use an elif clause,

well done!

8.Question 8

What’s the output of this code?

1

2

3

def sum(x, y):

        return(x+y)

print(sum(sum(1,2), sum(3,4)))





**1 / 1 point**

10

**Correct**

You nailed it! We’re calling the sum function 3 times: returning 3, then 7, then adding up 3 plus 7 for the total of 10.

9.Question 9

What's the value of this Python expression?

((10 >= 5\*2) and (10 <= 5\*2))

**1 / 1 point**



True



False



10



5\*2

**Correct**

Right on! When using the "and" operator, a statement is True if both parts of the conditional are True.

10.Question 10

The fractional\_part function divides the numerator by the denominator, and returns just the fractional part (a number between 0 and 1). Complete the body of the function so that it returns the right number. Note: Since division by 0 produces an error, if the denominator is 0, the function should return 0 instead of attempting the division.

**1 / 1 point**

1

2

3

4

5

6

7

8

9

10

11

12

13

def fractional\_part(numerator, denominator):

    # Operate with numerator and denominator to

    if denominator==0:

        return 0

    return (numerator/denominator)-(numerator//denominator)

# keep just the fractional part of the quotient

print(fractional\_part(5, 5)) # Should be 0

print(fractional\_part(5, 4)) # Should be 0.25

print(fractional\_part(5, 3)) # Should be 0.66...

print(fractional\_part(5, 2)) # Should be 0.5

print(fractional\_part(5, 0)) # Should be 0

print(fractional\_part(0, 5)) # Should be 0





RunReset

**Correct**

Well done! You're handling the math operations, as well as

division by 0, perfectly!