

- While Loops
- For Loops
- Recursion (Optional)
- Module Review
- ▶

Video: Loops Wrap Up

1 min
- ▶

Video: In Marga's Words: How I Got Into Programming

2 min
- ✔

Quiz: Module 3 Graded Assessment

10 questions
- 🗨

Discussion Prompt: Solving Problems with Loops

5 min



QUIZ • 50 MIN

✔

Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
90%

Module 3 Graded Assessment

LATEST SUBMISSION GRADE
90%

Submit your assignment

DUE Jun 21, 11:59 PM PDT

Try again

1. Fill in the blanks of this code to print out the numbers 1 through 7.

1 number = 1

2 while number <= 7:

3 print(number, end=" ")

4 number +=1

Grade
90%

Run

View Feedback

Reset

We keep your highest score

1 / 1 point

Correct

Nice job! You're really getting the hang of what goes into the while loops!

2. The show_letters function should print out each letter of a word on a separate line. Fill in the blanks to make that happen.

1 def show_letters(word):

2 for letter in word:

3 print(letter)

4

5 show_letters("Hello")

6 # Should print one line per letter

Run

Reset

1 / 1 point

Correct

Great job! You're working the "for" loops the way they're supposed to be done!

3. Complete the function digits(n) that returns how many digits the number has. For example: 25 has 2 digits and 144 has 3 digits. Tip: you can figure out the digits of a number by dividing it by 10 once per digit until there are no digits left.

1 def digits(n):

2 count = str(n)

3 return len(count)

4

5 print(digits(25)) # Should print 2

6 print(digits(144)) # Should print 3

7 print(digits(1000)) # Should print 4

8 print(digits(0)) # Should print 1

Run

Reset

1 / 1 point

Correct

Woohoo! You've cracked the code of writing code!

4. This function prints out a multiplication table (where each number is the result of multiplying the first number of its row by the number at the top of its column). Fill in the blanks so that calling multiplication_table(1, 3) will print out:

1 2 3

2 4 6

3 6 9

1 def multiplication_table(start, stop):

2 for x in range(1,4):

3 for y in range(1,4):

4 print(str(x*y), end=" ")

5 print()

6

7 multiplication_table(1, 3)

8 # Should print the multiplication table shown above

Run

Reset

1 / 1 point

Correct

Awesome! You've stepped up to the challenge of one of the more complex coding practices, nested loops!

5. The counter function counts down from start to stop when start is bigger than stop, and counts up from start to stop otherwise. Fill in the blanks to make this work correctly.

1 def counter(start, stop):

2 x = start

3 if x >stop:

4 return_string = "Counting down: "

5 while x >= stop:

6 return_string += str(x)

7 if x>stop:

8 return_string += ", "

9 x-=1

10 else:

11 return_string = "Counting up: "

12 while x <= stop:

13 return_string += str(x)

14 if x<stop:

15 return_string += ", "

16 x+=1

17 return return_string

18

19 print(counter(1, 10)) # Should be "Counting up: 1,2,3,4,5,6,7,8,9,10"

20 print(counter(2, 1)) # Should be "Counting down: 2,1"

21 print(counter(5, 5)) # Should be "Counting up: 5"

Run

Reset

1 / 1 point

Correct

You nailed it! You've figured out all of the situations that need to be considered!

6. The even_numbers function returns a space-separated string of all positive numbers that are divisible by 2, up to and including the maximum that's passed into the function. For example, even_numbers(6) returns "2 4 6". Fill in the blank to make this work.

1 def even_numbers(maximum):

2 return_string = ""

3 for x in range(maximum%2==0):

4 return_string += str(x) + " "

5 return return_string.strip()

6

7 print(even_numbers(6)) # Should be 2 4 6

8 print(even_numbers(10)) # Should be 2 4 6 8 10

9 print(even_numbers(1)) # No numbers displayed

10 print(even_numbers(3)) # Should be 2

11 print(even_numbers(0)) # No numbers displayed

Run

Reset

0 / 1 point

Incorrect

RuntimeErrorElement(RuntimeError,Error on line 7: print(even_numbers(6)) # Should be 2 4 6)
RuntimeErrorElement(RuntimeError,Error on line 3: for x in range():
TypeError: range expected 1 arguments, got 0
)

7. The following code raises an error when executed. What's the reason for the error?

1 def decade_counter():

2 while year < 50:

3 year += 10

4 return year

☐ Incrementing by 10 instead of 1

☒ Failure to initialize variables

☐ Nothing is happening inside the while loop