

Python While Loops

Textbook

Python While Loops

The `while` loop is used when you want code to run *while* something is true. You can think of a `while` loop as an `if` statement that repeats over and over again until a specific condition is met.



While loops run *while* a condition is true. For example, let's say we were writing a program for a roller coaster to run. We want the roller coaster to run *while* there are people in line. If there aren't any people in line, the roller coaster can stop. Here's what that code might look like:

```
1 people_in_line = 20
2
3 while people_in_line > 0:
4     print("There are " + str(people_in_line) + " people in line. Keep the coaster
    running!")
5     people_in_line = people_in_line - 1
```

Try it!

As you can see, this while loop will run the roller coaster while there are people in line.

Every time the loop runs, it subtracts 1 from the variable named `people_in_line`.

Every time the loop runs, it checks the line to see if it's empty. If the line is empty, the `people_in_line` condition is not met, which will end the while loop.

Caution!!! Infinite Loops!



You need to be very careful with using while loops. If you don't include a way for the condition to stop being met, the code will just run and run without an exit. **The programmer must include a way to quit the loop.** These are done with decrement and increment operators. If you don't include something that increments or decrements, your code will continue to run forever which is not good for your computer and leads to program crashes.

Decrement Operator

Notice this code in the example above:

```
people_in_line = people_in_line - 1
```

Every time the loop runs, the variable named `people_in_line` is subtracted by one. This means that the variable named `people_in_line` becomes a smaller and smaller integer until the condition is met (`people_in_line == 0`). Once the condition is met, the loop will quit. This stops an infinite loop from happening.

Here is another way of writing the decrement:

```
people_in_line -= 1
```

So the code all together would look like this:

```
1 people_in_line = 20
2
3 while people_in_line > 0:
4     print("There are " + str(people_in_line) + " people in line. Keep the coaster
    running!")
5     people_in_line -= 1
```

Increment Operator

An increment operator works the same way as a decrement operator. The increment simply adds to the variable instead of subtracting.

```
people_in_line = people_in_line + 1
```

```
people_in_line += 1
```

How to Stop an Infinite Loop



If an infinite loop happens, don't worry! You will need to quit your web browser and open a new window. Add the increment or decrement operator before running the loop again. Think about how to make sure the loop has a specified time and place to stop running. Remember that using increment and decrement

operators will tell your while loop to stop. If you don't tell it to stop in your program, it won't.

While Loop Example

Let's say you wanted to write a program where the user can input their age and the loop will print out all the numbers below their age. We can use a `while` loop to do this! Here's what that code would look like:

```
1 age = 17
2
3 while age >= 0:
4     print(age)
5     age -= 1
6
7
```

Try it!

This code will print all the numbers below their age, and stop after it prints 0.

Else

You can also add an else statement to run once the loop is finished.

```
1 age = 17
2
3 while age >= 0:
4     print(age)
5     age -= 1
6 else:
7     print("The loop is finished")
```

Try it!

Advanced Incrementing and Decrementing

There are other ways to increment and decrement by multiplying or dividing and then automatically reassigning the variable.

Multiplication Example

```
people_in_line = people_in_line * 2
```

```
people_in_line *= 1
```

Division Example

```
people_in_line = people_in_line / 2
```

```
people_in_line /= 1
```

Note: These are fairly advanced concepts. Mostly, just be aware that they exist. For this course we will mostly be using normal adding and subtracting in our increments/decrements.

Checkpoint

While Loops

Create a while loop that prints all the numbers from 1 - 50.

Requirements:

- Create a variable named number and assign it to 1
- Create a while loop where the loop will run until the variable named number reaches 50.
- Print out the numbers
- Using the += notation, increment the variable named number by 1 each time the loop runs.

Questions (10)

1. Which of the following are correct ways to include an increment operator? Select all that apply

SELECT MULTIPLE

Select all that apply:

- A. seeds = seeds + 1
- B. seeds += 1
- C. seed + s
- D. seeds = seeds += 1

2. True or False: While loops run when a certain condition is met.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

3. True or False: Infinite loops are created when there's no way to exit a loop

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

4. How many times will a while loop run?**Choose the correct answer:**

- A. 3 times
- B. Until the condition is false.
- C. Until it runs through a whole list.
- D. 1 time

5. Which is an example of an increment operator?**Choose the correct answer:**

- A. +=
- B. ^3
- C. -=
- D. +>

6. Which is an example of a decrement operator?**Choose the correct answer:**

- A. -=
- B. +=
- C. -^
- D. ->

7. How many times will this while loop run?

height = 14 while height > 10: print("height is " + str(height)) height = height - 1

Choose the correct answer:

- A. 1
- B. 4
- C. 14
- D. 10

8. How many times will this while loop run?

height = 8 while height > 13: print("height is " + height) height = height - 1

Choose the correct answer:

- A. 1
- B. 5
- C. 8
- D. 0

9. Edit the text box below to debug (fix) the code:

Code to Debug:

```
1 movies = 16
2
3 while movies > 12
4     print(movies)
5     movies -= 1
```

10. Edit the text box below to debug (fix) the code:

Code to Debug:

```
1 horses = 3
2
3 while horses <= 5:
4     print(I now have " + str(horses) + " horses.")
5     horses += 1
```

Challenges (6)

1. How many numbers?

1. Write a program that receives integers as inputs. The program will continue to ask the user to input an integer over and over again until the user enters the number "0".
2. When the user enters the number 0, the program will print out the total number of numbers that the user inputted.
3. Count how many numbers the user ended up inputting.

For example:

Inputs: 4 , 2 , 5 , 7 , 0

Output: 4

Another Example:

Inputs: 1 , 2 , 0

Output: 2

2. Sum of Numbers

Write a program that receives integers as inputs. The program will keep asking for inputs until the user enters the number 0 .

When the user enters the number 0 , the program will print out the total sum of all the numbers submitted.

For example:

Inputs: 4 , 2 , 5 , 7 , 0

Output: 18

Another example:

Inputs: 5 , 6 , 8 , 0

Output: 19

3. Largest Number

Write a program that receives integers as inputs. The program will continue asking the user for inputs **until the user enters the number "0"**.

When the user enters the number "0", the program will print the **largest number** that was submitted.

For example:

Inputs: 7 , 2 , 16 , 45 , 8 , 9 , 0

Output: 45

Another example:

Inputs: 5 , 2 , 13 , 99 , 12 , 0

Output: 99

Hint: Add an if statement inside the while loop. Pay CLOSE attention to indentation!

4. Find the Average

1. Write a program that receives integers as inputs.
2. The program will continue asking for inputs over and over again until the user enters the number "0".
3. When the user enters the number "0", the program will print out the average of all the numbers submitted.
4. Print the average, NOT including the "0". Your average should be a floating point number, which means it should have a decimal point.

For example:

Inputs: 1 , 10 , 5 , 2 , 9 , 0

Output: 5.4

Another example:

Inputs: 1 , 2 , 3 , 4 , 0

Output: 2.5

Hint: The average is the sum of all the numbers, divided by the number of numbers.

5. Running Trainer

1. This program will tell you how many days it should take you to prepare for a long distance race. Write a program that takes in two integers as input.
2. The first input is the number of miles you can run on day 1 (for this challenge, make sure this number is 10 or larger), and the second input is the number of miles you know you will be running on race day.
3. If you increase your mileage by **10%** every day, how many days will it take until you reach the point where you are running your race distance all in one day? Keep in mind that you will be taking 10% of the new distance run, which has already increased by 10 percent from the day before.
4. Your program should print out how many days it will take you to prepare for the race assuming you increase your mileage by 10% every day.
5. For example, if the inputs are 10 and 26 , your program should print 12 because if you increase your distance by 10% per day, it will take 12 days to hit 26 miles all in one day.

For example:

Inputs: 10 , 26

Output: 12

Another example:

Inputs: 10 , 30

Output: 13

Hint: Don't forget to include your first day in your total count of days it takes to get to your race distance.

6. Pirate Ship

1. Arrrrgh! You are the captain of a pirate ship. Create a program that repeatedly asks the captain of a pirate ship which way they want to go! (North, South, East, West).
2. Then, the program moves the ship to a new coordinate (x,y).
3. This should be done with a loop! The while loop will keep running until the ship moves outside the grid.

You are on a grid that goes 5 spaces in all 4 directions.

4. Keep that loop running the whole time starting with asking the captain which way he wants to go.
5. Then figure out which input means which direction on the compass rose, and change course that direction! If the user enters `Exit`, the loop will stop running and will print out the following.

`Program exited. You left the grid at (0,0)`

6. Be sure to print the final coordinates where you exit the grid in a final print statement with the words `You left the grid at (?,?)`

For example:

Inputs: `n`, `n`, `e`, `n`, `n`, `n`

Output: `You left the grid at (1,5)`

Another example:

Inputs: `n`, `n`, `n`, `n`, `n`

`Output: You left the grid at (0,5)`

Hint: Don't let the program crash if something other than n,s,e,w or Exit is entered! Let the captain know they used a "bad Input".

Make sure your capitalization is consistent!

Hint: Remember your data types!

Answer Keys & Solutions

Checkpoint Solutions

While Loops

```
1 number = 1
2 while number <= 50:
3     print(number)
4     number += 1
```

Questions

1. Which of the following are correct ways to include an increment operator?
Select all that apply

SELECT MULTIPLE

Correct Answers:

- | | |
|-----------------------|-------------|
| A. seeds = seeds + 1 | ✓ Correct |
| B. seeds += 1 | ✓ Correct |
| C. seed + s | ✗ Incorrect |
| D. seeds = seeds += 1 | ✗ Incorrect |

Explanation:

There are 2 correct answers

2. True or False: While loops run when a certain condition is met.

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------|-------------|
| A. True | ✓ Correct |
| B. False | ✗ Incorrect |

Explanation:

While loops can run over and over again until a condition is met.

3. True or False: Infinite loops are created when there's no way to exit a loop

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

While loops can create an infinite loop if a condition isn't met yet.

4. How many times will a while loop run?

MULTIPLE CHOICE

Correct Answer:

A. 3 times

✗ Incorrect

B. Until the condition is false.

✓ Correct

C. Until it runs through a whole list.

✗ Incorrect

D. 1 time

✗ Incorrect

Explanation:

While loops can run over and over again until a condition is met.

5. Which is an example of an increment operator?

MULTIPLE CHOICE

Correct Answer:

A. +=

✓ Correct

B. ^3

✗ Incorrect

C. -=

✗ Incorrect

D. +>

✗ Incorrect

Explanation:

Incrementing is adding to the value and making the value equal to that sum

6. Which is an example of a decrement operator?

Correct Answer:

- | | |
|-------|-------------|
| A. -- | ✓ Correct |
| B. += | ✗ Incorrect |
| C. -^ | ✗ Incorrect |
| D. -> | ✗ Incorrect |

Explanation:

Decrementing is subtracting and making the new value equal to the answer

7. How many times will this while loop run?

Correct Answer:

- | | |
|-------|-------------|
| A. 1 | ✗ Incorrect |
| B. 4 | ✓ Correct |
| C. 14 | ✗ Incorrect |
| D. 10 | ✗ Incorrect |

Explanation:

height starts at 14 and subtracts 1 each time the loop runs.

8. How many times will this while loop run?

Correct Answer:

- | | |
|------|-------------|
| A. 1 | ✗ Incorrect |
| B. 5 | ✗ Incorrect |
| C. 8 | ✗ Incorrect |
| D. 0 | ✓ Correct |

Explanation:

This loop will run as long as height is greater than 13

9. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 movies = 16
2
3 while movies > 12
4     print(movies)
5     movies -= 1
```

Correct Solution:

```
1 movies = 16
2
3 while movies > 12:
4     print(movies)
5     movies -= 1
```

Explanation:

This is missing a colon.

10. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 horses = 3
2
3 while horses <= 5:
4     print(I now have " + str(horses) + " horses.")
5     horses += 1
```

Correct Solution:

```
1 horses = 3
2
3 while horses <= 5:
4     print("I now have " + str(horses) + " horses.")
5     horses += 1
```

Explanation:

Check for correct quotation marks.

1. How many numbers?

Solution:

```
1 total = 0
2 number = int(input())
3 while number > 0:
4     number = int(input())
5     total += 1
6 print(total)
```

2. Sum of Numbers

Solution:

```
1 my_number = int(input("Pick a number."))
2 total = my_number
3 while my_number != 0:
4     my_number = int(input("Pick another number."))
5     total = total + my_number
6
7 print(total)
```

3. Largest Number

Solution:

```
1 my_number = int(input("Pick a number."))
2 largest = my_number
3 while my_number != 0:
4     my_number = int(input("Pick another number."))
5     if my_number > largest:
6         largest = my_number
7
8
9 print(largest)
```

4. Find the Average

Solution:

```
1 my_number = int(input("Pick a number."))
2 total = my_number
3 number_count = 1
4 while my_number != 0:
5     my_number = int(input("Pick another number."))
6     total = total + my_number
7     number_count += 1
8     if my_number == 0:
9         number_count -= 1
```

```
10
11 average = total / number_count
12
13 print(average)
```

5. Running Trainer

Solution:

```
1 day1 = int(input("Enter your first distance on day 1 (10 or bigger)"))
2 day2 = int(input("Enter your race distance"))
3 total_days = 1
4
5 while day1 <= day2:
6     day1 = day1 + (day1 / 10)
7     total_days += 1
8
9 print(total_days)
```

6. Pirate Ship

Solution:

```
1 x=0
2 y=0
3 run=True
4 while x<5 and y<5 and x>=-5 and y>=-5 and run == True:
5     direction=input('What way would you like to go (N,S,E,W)? ')
6     if direction=='n':
7         y+=1
8     elif direction=='s':
9         y-=1
10    elif direction=='e':
11        x+=1
12    elif direction=='w':
13        x-=1
14    elif direction=='Exit':
15        print('Program exited. You left the grid at (' + str(x) + ',' + str(y) + ')')
16        run=False
17    else:
18        print('Bad input')
19 print('You left the grid at ' + '(' + str(x) + "," + str(y) + ')')
```