Introduction to Computer Programming

Condition Controlled Loops In Practice

CSCI-UA.0002-005
While Loop Basics

Programmers commonly find that they need to write code that performs the same task over and over again.

One solution to this kind of problem is to use "while" loop, which involves the following:

- Write the code for the task one time
- Place the code into a while loop which causes Python to repeat it over and over
- Writing a condition that can be used to stop the loop once the desired # of repetitions has been achieved
- Then double-check your work by asking: which tasks happen before the loop? Which tasks happen during the loop? Which tasks happen after the loop?
While Loop Basics

1. Evaluate a Boolean expression.

3. If it is False, skip the block of statements associated with the while loop and condition the program as normal

5. If it is True
   - Execute a series of statements.
   - At the end of the statement block re-evaluate the condition
   - If it is True, repeat the block of statements
   - If it is False, skip the block of statements associated with the while loop and continue the program as normal
When to use While

We can use while loops when we need to repeat a task multiple times in order to solve a particular problem.

For example:
- Print the phrase "Hello, World" 100 times
- Ask the user to enter 10 price values and add them to an accumulator variable

A while loop works well for tasks that require an unknown number of iterations.

For example:
- Ask the user to enter in a positive number. If the user enters a negative number, re-prompt them until they supply a positive number.
When to use While

Write a program that asks the user for two numbers

If the first number is greater than the second number, add the numbers and display the total

If the second number is greater than the first number, multiply the numbers and display the product

Enter 1st number: 5
Enter 2nd number: 10
Product: 50
When to use While

Write a program that asks the user for two numbers. Only accept positive values.

If the first number is greater than the second number, add the numbers and display the total.

If the second number is greater than the first number, multiply the numbers and display the product.

Enter 1st number: 5
Enter 2nd number: -10
Sorry, try again
Enter 2nd number: 10
Product: 50
When to use While

Ask the user to enter in 5 price values

Add these values to a total variable

Print out the total at the end of the program plus 7% sales tax

Enter price: 1.00
Enter price: 2.00
Enter price: 3.00
Enter price: 3.00
Enter price: 1.00

Your total: 10.00
Tax: 0.70
Grand total: 10.07
When to use While

Ask the user to enter in a potentially *unlimited* number of price values

Add these values to a total variable

Print out the total at the end of the program plus 7% sales tax

Enter price, 0 to end: 1.00
Enter price, 0 to end: 2.00
Enter price, 0 to end: 3.00
Enter price, 0 to end: 3.00
Enter price, 0 to end: 1.00
Enter price, 0 to end: 0

Your total: 10.00
Tax: 0.70
Grand total: 10.07
Challenge

Write a program that asks the user for three numbers

Test those numbers against three “secret” numbers that represent the combination to a virtual padlock

If the user gets the numbers right you should let them know that they have gained access to your program

If not, allow them to continue to enter combinations until they guess correctly
Challenge

Assume you have a jar that contains 5 marbles. The jar can hold 10 marbles total.

Continually ask the user if they want to add or remove a marble

If they add a marble you should increase the total # of marbles in the jar. If the jar is full tell the user and end the program.

If they remove a marble you should decrease the # of marbles in the jar. If the jar is empty you should tell the user and end the program.
Start the user with $100

Ask them to guess a number between 1 and 12

Next, spin a virtual wheel and tell the user which number came out

Each spin costs $10

If their number is spun then they win $100

Stop when the user loses all of their money, or when they choose to retire from the game
next steps:

begin “Self Paced Learning Module # 5”

work on Assignment #4: ‘Repetition Structures’