Condition Controlled Loops

Introduction to Programming - Python
Repetition Structures
Why Computers?

- One of the great things about computers is that they can do tedious, repetitive tasks over and over again, without getting bored. You program it once, and it will do it as many times as needed to get the job done.
Remember These Computers?
The ENIAC (1945)
Range table for 3-inch field gun.

<table>
<thead>
<tr>
<th>Range (Yds.)</th>
<th>Angle of departure.</th>
<th>Angle of elevation.</th>
<th>One minute, in yards of range.</th>
<th>One mil, in yards of range.</th>
<th>( \Delta X ) for ( \pm ) 10 f.s. M.V.</th>
<th>( \Delta X ) for wind 10 miles per hour.</th>
<th>Deviation for 10 miles cross wind.</th>
<th>Angle of fall.</th>
<th>Slope of fall.</th>
<th>Time of flight.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>112.1</td>
<td>21.0</td>
<td>105.6</td>
<td>11.2</td>
<td>38</td>
<td>8.8</td>
<td>18.1</td>
<td>1.80</td>
<td>2.60</td>
<td>1.50</td>
</tr>
<tr>
<td>1,100</td>
<td>120.2</td>
<td>23.6</td>
<td>14.5</td>
<td>10.8</td>
<td>36</td>
<td>9.5</td>
<td>21.7</td>
<td>2.20</td>
<td>2.80</td>
<td>1.60</td>
</tr>
<tr>
<td>1,200</td>
<td>129.4</td>
<td>26.4</td>
<td>123.8</td>
<td>10.4</td>
<td>35</td>
<td>10.2</td>
<td>23.3</td>
<td>2.60</td>
<td>3.10</td>
<td>1.80</td>
</tr>
<tr>
<td>1,300</td>
<td>139.0</td>
<td>29.3</td>
<td>133.4</td>
<td>10.0</td>
<td>33</td>
<td>10.8</td>
<td>29.1</td>
<td>3.10</td>
<td>3.30</td>
<td>2.00</td>
</tr>
<tr>
<td>1,400</td>
<td>149.0</td>
<td>32.3</td>
<td>11.2</td>
<td>9.6</td>
<td>32</td>
<td>11.4</td>
<td>32.9</td>
<td>3.50</td>
<td>3.60</td>
<td>2.20</td>
</tr>
<tr>
<td>1,500</td>
<td>159.4</td>
<td>35.4</td>
<td>153.8</td>
<td>9.4</td>
<td>31</td>
<td>12.1</td>
<td>36.9</td>
<td>4.00</td>
<td>3.90</td>
<td>2.30</td>
</tr>
<tr>
<td>1,600</td>
<td>170.3</td>
<td>38.6</td>
<td>204.7</td>
<td>9.2</td>
<td>31</td>
<td>12.7</td>
<td>42.1</td>
<td>4.50</td>
<td>4.20</td>
<td>2.40</td>
</tr>
<tr>
<td>1,700</td>
<td>181.5</td>
<td>41.9</td>
<td>215.9</td>
<td>9.0</td>
<td>30</td>
<td>13.9</td>
<td>45.5</td>
<td>5.10</td>
<td>4.80</td>
<td>2.50</td>
</tr>
<tr>
<td>1,800</td>
<td>192.9</td>
<td>45.3</td>
<td>227.3</td>
<td>8.8</td>
<td>29</td>
<td>13.9</td>
<td>49.8</td>
<td>5.60</td>
<td>5.30</td>
<td>2.60</td>
</tr>
<tr>
<td>1,900</td>
<td>204.7</td>
<td>48.8</td>
<td>239.1</td>
<td>8.5</td>
<td>28</td>
<td>14.5</td>
<td>54.1</td>
<td>6.00</td>
<td>5.80</td>
<td>2.70</td>
</tr>
<tr>
<td>2,000</td>
<td>216.7</td>
<td>52.4</td>
<td>251.1</td>
<td>8.5</td>
<td>27</td>
<td>15.0</td>
<td>58.4</td>
<td>6.80</td>
<td>6.40</td>
<td>2.80</td>
</tr>
</tbody>
</table>

Range table for 3-inch field gun.

<table>
<thead>
<tr>
<th>Range (Yds.)</th>
<th>Angle of departure.</th>
<th>Angle of elevation.</th>
<th>One minute, in yards of range.</th>
<th>One mil, in yards of range.</th>
<th>( \Delta X ) for ( \pm ) 10 f.s. M.V.</th>
<th>( \Delta X ) for wind 10 miles per hour.</th>
<th>Deviation for 10 miles cross wind.</th>
<th>Angle of fall.</th>
<th>Slope of fall.</th>
<th>Time of flight.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,100</td>
<td>526.6</td>
<td>96.8</td>
<td>20.0</td>
<td>6.7</td>
<td>22</td>
<td>19.5</td>
<td>107.1</td>
<td>13.80</td>
<td>9.80</td>
<td>70.0</td>
</tr>
<tr>
<td>3,200</td>
<td>541.6</td>
<td>101.3</td>
<td>20.0</td>
<td>6.6</td>
<td>21</td>
<td>19.8</td>
<td>111.3</td>
<td>14.40</td>
<td>9.80</td>
<td>70.0</td>
</tr>
<tr>
<td>3,300</td>
<td>556.9</td>
<td>105.9</td>
<td>19.5</td>
<td>6.5</td>
<td>20</td>
<td>20.1</td>
<td>115.5</td>
<td>15.00</td>
<td>9.80</td>
<td>70.0</td>
</tr>
<tr>
<td>3,400</td>
<td>572.4</td>
<td>110.5</td>
<td>20.0</td>
<td>6.8</td>
<td>19</td>
<td>20.4</td>
<td>119.7</td>
<td>15.70</td>
<td>9.80</td>
<td>70.0</td>
</tr>
<tr>
<td>3,500</td>
<td>587.7</td>
<td>115.2</td>
<td>20.0</td>
<td>6.1</td>
<td>18</td>
<td>20.8</td>
<td>123.9</td>
<td>16.40</td>
<td>9.80</td>
<td>70.0</td>
</tr>
</tbody>
</table>
Repetition Structures

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

- Consider, for example, our Turtle program...

- Our code ended up being one long sequence structure which contained a lot of duplicate code.

- There are several disadvantages to this approach:
  - Your programs will tend to get very large.
  - Writing this kind of program can be extremely time consuming.
  - If part of the duplicated code needs to be corrected then the correction must be implemented many times.
Example: Commission calculator for a sales team

- Write a program that allows the user to calculate sales commission earned by each member of a sales team.

- Currently there are 3 people on the sales team, but there may be more in the future.

- Input
  - Gross sales (float)
  - Commission Rate (float)

- Process
  - Commission = gross sales * commission rate

- Output
  - Commission earned
One solution to this kind of problem is to use a repetition structure, which involves the following:

- Write the code for the operation one time
- Place the code into a special structure that causes Python to repeat it as many times as necessary

We call this a “repetition structure” or, more commonly, a “loop”

There are a variety of different repetition structures that can be used in Python
Condition Controlled Loops
A condition controlled loop is programming structure that causes a statement or set of statements to repeat as long as a condition evaluates to True
Condition Controlled Loops

Start

Boolean Expression

True → do something ...

False

Finish
The “While” Loop

- In Python we can implement a condition controlled loop by writing a “while” loop.

- “while” loops work as follows:
  - Evaluate a Boolean expression.
  - If it is False, skip the block of statements associated with the while loop and continue the program as normal.
  - If it is True
    - Execute the block of statements inside the loop.
    - Go back to the conditional at the top of the loop and start again.
The “While” Loop

while condition:
    statement
    statement
    statement
    statement

{ }

the statements that will be repeated

indentation indicates that the statements under the while loop should be repeated

standard Boolean condition that evaluates to True or False
Programming Challenge: Commission Calculator

- Write a program that allows the user to calculate sales commission earned by each member of a sales team.

- **Input**
  - Gross sales (float)
  - Commission Rate (float)

- **Process**
  - Commission = gross sales * commission rate

- **Output**
  - Commission earned
Some notes on “while” loops

- We refer to the process of going through a loop as an “iteration”

- If a loop cycles through 5 times then we say we have “iterated” through it 5 times

- The “while” loop is considered a “pre-test” loop, meaning that it only iterates upon the successful evaluation of a condition

- This means that you always need to “set up” your loop prior to Python being able to work with it (i.e. setting up a control variable)
Warning!

- When working with a “while” loop there is nothing to prevent you from writing a Boolean condition that will never evaluate to False.

- If this happens your loop will continue executing forever, or until you send an “interrupt” to IDLE using the CTRL-C key combination.

- We call this an “infinite loop” since it never stops executing.

- With the exception of a few special cases you want to try and avoid writing infinite loops.
Trace the Output

\[ a = 5 \]

\[ \text{while } a < 10: \]

\[ \text{print } ("A is less than 10!") \]
Infinite Loop

Start

Get Temp.

Temp > hi_thresh?

YES → Turn on A/C

NO → Temp < lo_thresh?

YES → Turn on Heat

NO → Turn off whatever is on.

NO
Divisibility Tester

- Write a program that lets the user test to see if a series of numbers are evenly divisible by another number (3). If they are, print out a status message telling the user.

- Extension: Start off by asking the user to enter in the number that should be used during the test (i.e. enter 5 if you want to test to see if a range of numbers is evenly divisible by 5)
Programming Challenge: Guess the Number

- Rewrite the “guess the number” game we wrote back in the selection statement unit to use a “while” loop.
- Allow the user to continually guess a number until they eventually guess the correct number.
Programming Challenge: Combo Lock

- 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