Python 2 – Conditionals and loops
CS111

Yesterday’s Lecture

Input and Output
user_input_as_str = input('Prompt: ')
print('Hello. You entered: ' + user_input_as_str)

Comments
# everything to the right of the # is ignored
print('hi') # this is ignored

Data types (and conversion functions)
Strings, integers and floats
str(), int(), and float()

Assigning values to variables and
(Overloaded) Operators -- *, %, +, /, //

# get input from user
feet = int(input('Enter feet: '))

# convert to metres
feet_to_metres = 0.3048
metres = feet * feet_to_metres

# print output
print(str(feet) + ' ft. equals ' + str(metres) + ' m.')

Printing multiple things out

Concatenate
print(str(feet) + ' feet is equal to ' + str(metres) + ' metres.')

Or use commas
print(feet, 'feet is equal to ', metres, ' metres.

Today’s lecture

conditional statements
conditional execution. “If something is the case, then run these statements, otherwise, run those statements.”

loops
repeated execution. “Run this sequence of statements again and again until XXX.”

Reminder: Please get a pencil / pen and paper (or a laptop) out, so you can work through some exercises.
Conditional Statements

**If...**

**IF statement**

Conditional activity (i.e. 'if this then do that') is an important part of many programs.

The if statement lets you introduce conditional activity into your program.

Statements that are executed when if is true must be tabbed underneath the if statement.

In Python, white space is important! Incorrect whitespace can cause syntax errors.

```
if [logical condition]:
    [lines of code here]
else:
    [lines of code here]
```

**Comparison operators**

The following will evaluate to true or false:

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Operator</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than</td>
<td><code>&lt;</code></td>
<td><code>a &lt; b</code></td>
</tr>
<tr>
<td>Less than or equal to</td>
<td><code>&lt;=</code></td>
<td><code>a &lt;= b</code></td>
</tr>
<tr>
<td>Greater than</td>
<td><code>&gt;</code></td>
<td><code>a &gt; b</code></td>
</tr>
<tr>
<td>Greater than or equal to</td>
<td><code>&gt;=</code></td>
<td><code>a &gt;= b</code></td>
</tr>
<tr>
<td>Equal to</td>
<td><code>==</code></td>
<td><code>a == b</code></td>
</tr>
<tr>
<td>Not equal to</td>
<td><code>!=</code></td>
<td><code>a != b</code></td>
</tr>
</tbody>
</table>

**Logical operators**

You can combine comparisons using the following operators:

- **and**
  - If the test in `a and b` evaluate to true, then the logical condition will be true, e.g:
    ```python
    if x > 1 and y < 2:
    ```

- **or**
  - If either `a or b` evaluate to true, then the overall logical condition will be true, e.g:
    ```python
    if x > 1 or y < 2:
    ```
Logical operators

**not**
Inverts True into False and False into True. eg:

```python
if not(x):
```

Exercise: Logical and Boolean Operators

<table>
<thead>
<tr>
<th>Program A</th>
<th>Program E</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>print( 5 &gt; 4 )</code></td>
<td><code>x = 5</code></td>
</tr>
<tr>
<td></td>
<td><code>print( x != 3 )</code></td>
</tr>
<tr>
<td>Program B</td>
<td>Program F</td>
</tr>
<tr>
<td><code>print( 5 == 4 )</code></td>
<td><code>x = 4</code></td>
</tr>
<tr>
<td>Program C</td>
<td></td>
</tr>
<tr>
<td><code>print( 5 &gt; 4 and False )</code></td>
<td><code>print( x==0 or x==1 )</code></td>
</tr>
<tr>
<td></td>
<td><code>print( True or False )</code></td>
</tr>
<tr>
<td>Program D</td>
<td>Program G</td>
</tr>
<tr>
<td><code>x = 0</code></td>
<td></td>
</tr>
<tr>
<td><code>print( x &lt;= 5 and x &gt;= 0 )</code></td>
<td><code>print( not(x) or x )</code></td>
</tr>
<tr>
<td></td>
<td><code>print( not(x) and x )</code></td>
</tr>
</tbody>
</table>

**IF statement example**

Write a program that asks the user to enter a number between 1 and 10 (inclusive). The program will print out "Correct" if the number is in the range and "Incorrect" if the number is outside the range.

Example (user input is displayed in **bold**):

```
Enter a number (1-10): 34
Incorrect
Enter a number (1-10): 6
Correct
```

Answer

```python
# get input from user, and put it in the variable called "number"
users_number = int( input("Enter a number (1-10): ") )

# print output depending on input
if users_number >= 1 and number <= 10:
    print("Correct")
else:
    print("Incorrect")
```
Exercise

Write a program that asks the user to enter a number. The program should determine if the number is odd or even, and print out an appropriate message.

Example (user input is displayed in **bold**)

Enter a number: 56
You entered 56 which is even

Enter a number: 33
You entered 33 which is odd

Hint

How do you check if a number is even?
What is an even number?

- It is a number that is divisible by 2.
- When you divide it by 2, there is nothing extra left over. There is no "remainder".

Operators

<table>
<thead>
<tr>
<th>Operation</th>
<th>Symbol</th>
<th>Applied to integers</th>
<th>Applied to floating point numbers</th>
<th>Applied to strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponent</td>
<td>**</td>
<td>2 ** 3 = 8</td>
<td>2.0 ** 3.0 = 8.0</td>
<td>N/A (ERROR)</td>
</tr>
<tr>
<td>Multiply</td>
<td>*</td>
<td>2 * 2 = 4</td>
<td>2.0 * 2.0 = 4.0</td>
<td>&quot;a&quot; * 3 = &quot;aaa&quot;</td>
</tr>
<tr>
<td>Divide</td>
<td>/</td>
<td>10/3 = 3.333</td>
<td>10.0/3.0 = 3.333</td>
<td>N/A (ERROR)</td>
</tr>
<tr>
<td>Divide (integer)</td>
<td>//</td>
<td>10 // 3 = 3</td>
<td>10.0//3.0 = 3.0</td>
<td>N/A (ERROR)</td>
</tr>
<tr>
<td>Remainder</td>
<td>%</td>
<td>10 % 3 = 1</td>
<td>10.0 % 3.0 = 1.0</td>
<td>N/A (ERROR)</td>
</tr>
<tr>
<td>Add</td>
<td>+</td>
<td>8 + 9 = 17</td>
<td>8.0 + 9.0 = 17.0</td>
<td>&quot;a&quot; + &quot;b&quot; = &quot;ab&quot;</td>
</tr>
<tr>
<td>Subtract</td>
<td>-</td>
<td>9 - 7 = 2</td>
<td>9.0 - 7.0 = 2.0</td>
<td>N/A (ERROR)</td>
</tr>
</tbody>
</table>
## Answer

```python
# get input from user, and put it in the variable called "number"
users_number = int(input("Enter a number: "))

# print output, reporting on whether or not number is even
if (users_number % 2) == 0:
    print("You entered ", users_number, " which is even")
else:
    print("You entered ", users_number, " which is odd")
```

## Loops

**Loops**

Used to perform a sequence of statements repeatedly.

Statements that are executed when the "while's condition" is True must be tabbed underneath the while statement.

**Syntax:**

```python
while [logical condition]:
    [statements that will repeatedly run while the condition is true...]
    [statements that run once the condition is no longer true]
```

**While...**

- **Execute the sequence of statements in the block.**
- **Is the condition true?**
  - **Yes**
  - **No**
- **Statements after the loop**
- **Statements before the loop**
Example

Goal
Write a program to print the numbers from 1 to 5

Program

```python
current = 1
while current <= 5:
    print(current)
    current = current + 1

print("Finished!")
```

Output

```
1
2
3
4
5
Finished!
```

Exercise

Write a program that repeatedly asks the user to enter a number between 1 and 10 (inclusive). When they do so, the program should print "Well done" on screen and end (i.e. stop asking the user to enter a number).

Example (user input is displayed in bold):

```
Enter a number between 1 and 10: 99
Enter a number between 1 and 10: -1
Enter a number between 1 and 10: 10
Well done
```

Answer

```python
# get input from user, and put it in the variable called "number"
number = int(input("Enter a number between 1 and 10: "))

# keep asking for a number until it is in range
while number < 1 or number > 10:
    number = int(input("Enter a number between 1 and 10: "))
print("Well done")
```
Exercise

Write a program that repeatedly asks the user to enter a number. If the number is even, then “x is even” (where x is the number) should be printed on screen. If the number is odd, then “x is odd” should be printed on screen. The program must print “Thanks” and end when the user types ‘0’.

Example (user input is displayed in **bold**)

Please enter a number: 45
45 is odd

Please enter a number: 12
12 is even

Please enter a number: 0

Thanks

Answer

```python
user_input = int(input("Please enter a number: "))

# until the user input is 0...
while user_input != 0:
    # report on evenness of input
    if user_input % 2 == 0:
        print(user_input, "is even")
    else:
        print(user_input, "is odd")

    # the user's input was 0
    user_input = int(input("Please enter a number: "))

print("Thanks")
```

Summary

**if statements**
allow you to introduce conditional activities into your program

```python
x = 10
if x == 5:
    print("x is equal to five.")
if x % 2 == 0:
    print("x is even")
```

**while loops**
allow you to repeat certain statements for as long as the logical condition evaluates to true

```python
x = 0
while x < 10:
    print(x)
    x = x + 1
```