Python Programming: An Introduction to Computer Science

Chapter 2
Example Program: Temperature Converter

- Analysis – the temperature is given in Celsius, user wants it expressed in degrees Fahrenheit.

- Specification
  - Input – temperature in Celsius
  - Output – temperature in Fahrenheit
  - Output = \( \frac{9}{5}(\text{input}) + 32 \)
Example Program: Temperature Converter

- Design
  - Input, Process, Output (IPO)
  - Prompt the user for input (Celsius temperature)
  - Process it to convert it to Fahrenheit using \( F = \frac{9}{5}(C) + 32 \)
  - Output the result by displaying it on the screen
Example Program: Temperature Converter

- Before we start coding, let’s write a rough draft of the program in *pseudocode*.
- Pseudocode is precise English that describes what a program does, step by step.
- Using pseudocode, we can concentrate on the algorithm rather than the programming language.
Example Program: Temperature Converter

- Pseudocode:
  - Input the temperature in degrees Celsius (call it celsius)
  - Calculate fahrenheit as \((9/5)\times celsius + 32\)
  - Output fahrenheit

- Now we need to convert this to Python!
Example Program: Temperature Converter

#convert.py
# A program to convert Celsius temps to Fahrenheit
# by: Susan Computewell

def main():
    celsius = eval(input("What is the Celsius temperature? ")
    fahrenheit = (9/5) * celsius + 32
    print("The temperature is ",fahrenheit," degrees Fahrenheit.")

main()
Example Program: Temperature Converter

- Once we write a program, we should test it!

```python
>>> What is the Celsius temperature? 0
The temperature is  32.0  degrees Fahrenheit.
>>> main()
What is the Celsius temperature? 100
The temperature is  212.0  degrees Fahrenheit.
>>> main()
What is the Celsius temperature? -40
The temperature is  -40.0  degrees Fahrenheit.
>>> 
```
Elements of Programs

- Names
  - Names are given to variables (celsius, fahrenheit), modules (main, convert), etc.
  - These names are called *identifiers*
  - Every identifier must begin with a letter or underscore ("_"), followed by any sequence of letters, digits, or underscores.
  - Identifiers are case sensitive.
Elements of Programs

- These are all different, valid names
  - X
  - Celsius
  - Spam
  - spam
  - spAm
  - Spam_and_Eggs
  - Spam_And_Eggs
Some identifiers are part of Python itself. These identifiers are known as reserved words. This means they are not available for you to use as a name for a variable, etc. in your program.

and, del, for, is, raise, assert, elif, in, print, etc.

For a complete list, see table 2.1
Elements of Programs

- Expressions
  - The fragments of code that produce or calculate new data values are called *expressions*.
  - *Literals* are used to represent a specific value, e.g. 3.9, 1, 1.0
  - Simple identifiers can also be expressions.
Elements of Programs

- Output Statements
  - A print statement can print any number of expressions.
  - Successive print statements will display on separate lines.
  - A bare print will print a blank line.
Assignment Statements

- Simple Assignment
- `<variable> = <expr>`
  variable is an identifier, expr is an expression
- The expression on the RHS is evaluated to produce a value which is then associated with the variable named on the LHS.
Assignment Statements

- Variables are like a box we can put values in.
- When a variable changes, the old value is erased and a new one is written in.

Before \[ x = 10 \]  

After \[ x = x + 1 \]  

\[ x = 11 \]
Assignment Statements

- Technically, this model of assignment is simplistic for Python.
- Python doesn't overwrite these memory locations (boxes).
- Assigning a variable is more like putting a “sticky note” on a value and saying, “this is x”.

\[ x = x + 1 \]
Definite Loops

for <var> in <sequence>:
    <body>

    The variable after the *for* is called the *loop index*. It takes on each successive value in *sequence*. 
In chaos.py, what did `range(10)` do?

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- `range` is a built-in Python function that generates a sequence of numbers, starting with 0.
- `list` is a built-in Python function that turns the sequence into an explicit list.
- The body of the loop executes 10 times.
Example Program: Future Value

- Analysis
  - Money deposited in a bank account earns interest.
  - How much will the account be worth 10 years from now?
  - Inputs: principal, interest rate
  - Output: value of the investment in 10 years