Chapter 1

Python Beginnings

Cong Pu (Ph.D., Instructor)
cong.pu@ttu.edu

Suggestions of Programming

1: **Think** before you program.

2: A program is a **human-readable essay** on problem solving that also happens to execute on a computer.

3: The best way to improve your programming and problem skills is to **practice**.

Say “Hello” to “World”

- Creating and Running programs:
  - Go into IDLE if you are not already.
  - In the menu at the top, select *File* then *New File*.
  - In the next window that appears, type the following:
    - print (“Hello, World!”)
  - Now save the program:
    - Select *File* from the menu, then *Save*. Save it as “ch01_00.**py**” (you can save in any folder that you want.)
    - Now that it is saved it can be run.
  - Next run the program by going to *Run* then *Run Module*.
  - This will output Hello, World. on the Python Shell window.

Say “Hello” to “World”

- print()
  - Output function
- String
  - “Hello World!”
  - They can be enclosed in single quotes (‘…’) or double quotes (“…”) with the same result.
  - The string is enclosed in double quotes if the string contains a single quote and no double quotes, otherwise it is enclosed in single quotes or double quotes.

Program

- Task: calculate the **circumference** and **area** of a circle with **radius 2**.
- Relevant mathematical formulas:
  - Circumference = 2 * PI * radius
  - Area = PI * radius^2
- To create the program, we need to do a couple of things:
  1. We need to **have radius value**.
  2. We need to **apply the mathematical formulas** listed previously using the acquired radius to find the circumference and area.
  3. We need to **print out the results**.

Program

- Task: calculate the circumference and area of a circle with radius 2. (ch01_01.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: have a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_int = 2
10
11 circumference = 2 * math.pi * radius_int
12 area = math.pi * (radius_int ** 2)
13
14 print ("The cirumference is:",circumference,  \
15        ", and the area is:",area)
```

Examine ch01_01.py

- Lines 1-5:
  - Anything that follows a **pound sign** (#) is a ***comment*** for the human reader.
  - The Python interpreter **ignores** it.

```
1 # calculate the area and circumference
2 #   of a circle from its radius
3 # Step 1: have a radius
4 # Step 2: apply the area formula
5 # Step 3: Print out the results
```

Program

- Task: calculate the circumference and area of a circle with radius 2. (ch01_01.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: have a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_int = 2
10
11 circumference = 2 * math.pi * radius_int
12 area = math.pi * (radius_int ** 2)
13
14 print ("The cirumference is:",circumference,  \
15        ", and the area is:",area)
```

Examine ch01_01.py

- Line 7:
  - This line imports special Python code from the math ***module***.
  - A *module* is a Python file containing programs to solve particular problems.
  - We are interested in **PI** value provided by *math* module.
  - https://docs.python.org/3.0/library/math.html

```
7  import math
```

```
12 area = math.pi * (radius_int ** 2)
```

Program

- Task: calculate the circumference and area of a circle with radius 2. (ch01_01.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: have a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_int = 2
10
11 circumference = 2 * math.pi * radius_int
12 area = math.pi * (radius_int ** 2)
13
14 print ("The cirumference is:",circumference,  \
15        ", and the area is:",area)
```

Examine ch01_01.py

- Line 9: the Python code to the right of the = sign and the Python code on the left.
  - Variable, a name that is associated with a value.

```
9  radius_int = 2
```

Program

- Task: calculate the circumference and area of a circle with radius 2. (ch01_01.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: have a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_int = 2
10
11 circumference = 2 * math.pi * radius_int
12 area = math.pi * (radius_int ** 2)
13
14 print ("The cirumference is:",circumference,  \
15        ", and the area is:",area)
```

Examine ch01_01.py

- Lines 11-12:
  - Implement the formulas

```
11  circumference = 2 * math.pi * radius_int
12  area = math.pi * (radius_int ** 2)
```

Program

- Task: calculate the circumference and area of a circle with radius 2. (ch01_01.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: have a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_int = 2
10
11 circumference = 2 * math.pi * radius_int
12 area = math.pi * (radius_int ** 2)
13
14 print ("The cirumference is:",circumference,  \
15        ", and the area is:",area)
```

Examine ch01_01.py

- Lines 14-15:
  - ***print*** function can print **strings** bracketed by quotes and a **value associated with a variable** to the user console.
    - If the elements being printed is quoted, it is printed exactly as it appears in the quotes.
    - If the element is a variable, then the value associated with the variable is printed.
    - Each object that is to be printed is separated from other objects by commas.

```
14 print ("The cirumference is:",circumference, \
15         ", and the area is:",area)
```

Indicates that the statement continues onto the next line

Program

- Task: calculate the **circumference** and **area** of a circle **given its radius** from user.
- Relevant mathematical formulas:
  - Circumference = 2 * PI * radius
  - Area = PI * radius^2
- To create the program, we need to do a couple of things:
  1. We need to **prompt the user for a radius**.
  2. We need to **apply the mathematical formulas** listed previously using the acquired radius to find the circumference and area.
  3. We need to **print out the results**.

Program

- Task: calculate the circumference and area of a circle given its radius. (ch01_02.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: prompt for a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_str = input("Enter the radius of your circle: ")
10 radius_int = int(radius_str)
11
12 circumference = 2 * math.pi * radius_int
13 area = math.pi * (radius_int ** 2)
14
15 print ("The cirumference is:",circumference,  \
16        ", and the area is:",area)
```

Examine ch01_02.py

- Line 9: the Python code to the right of the = sign and the Python code on the left.

```
9 radius_str = input("Enter the radius of your circle: ")
```

Examine ch01_02.py

- Line 9: On the right of the = sign
  - ***input*** is a small Python program called a ***function***.
  - ***input*** function prints the characters in quotes to the Python shell and waits for the user to type a response.
  - Whatever the user types in the shell before pressing the Enter key is provided as input to a program.

```
input("Enter the radius of your circle: ")
```

```
9 radius_str = input("Enter the radius of your circle: ")
```

Examine ch01_02.py

- Line 9: On the left of the = sign
  - Variable, a name that is associated with a value.
  - The value returned from ***input*** will be associated with the name radius_str.
  - =, linking the value on the right side with the variable on the left.
  - = is called an assignment operator.

radius_str

```
9 radius_str = input("Enter the radius of your circle: ")
```

Program

- Task: calculate the circumference and area of a circle given its radius. (ch01_02.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: prompt for a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_str = input("Enter the radius of your circle: ")
10 radius_int = int(radius_str)
11
12 circumference = 2 * math.pi * radius_int
13 area = math.pi * (radius_int ** 2)
14
15 print ("The cirumference is:",circumference,  \
16        ", and the area is:",area)
```

Examine ch01_02.py

- Line 10:
  - The user's response returned by input is stored as a **sequence of characters**.
  - Python differentiates a **sequence of characters** from **numbers** on which we can perform operations such as addition, subtraction, …
  - ***int*** function converts a string of characters to numbers.

```
10  radius_int = int(radius_str)
```

Program

- Task: calculate the circumference and area of a circle given its radius. (ch01_02.py)

```
1  # calculate the area and circumference
2  #   of a circle from its radius
3  # Step 1: prompt for a radius
4  # Step 2: apply the area formula
5  # Step 3: Print out the results
6
7  import math
8
9  radius_str = input("Enter the radius of your circle: ")
10 radius_int = int(radius_str)
11
12 circumference = 2 * math.pi * radius_int
13 area = math.pi * (radius_int ** 2)
14
15 print ("The cirumference is:",circumference,  \
16        ", and the area is:",area)
```

An Interactive Session

- An important feature of Python is that it is an *interpreted* language.
- Python has an *interpreter:*
  - Takes each line of Python code, one line at a time, and executes that code.
  - This feature allows us to **try out lines of code one at a time by typing into the Python shell**.
- Consider circumference program and type each line of codes into Python shell.

Part of A Program - Modules

- A *module* contains a set of Python commands.
- A *module* can be stored as a file and *imported* into the Python shell.
- Usage:

import module *# load the module*

- Hundreds of modules come with the standard Python distribution. https://docs.python.org/3.4/py-modindex.html
- You can even write your own modules and use them as tools in your own programming work.

Part of A Program - Statement

- A statement **does not return** a value, but does **perform some task**.
- Statements perform a wide variety of tasks:
  - Control the flow of the program
  - Ask for resources
- As a result of their operations, a statement may have a ***side effect***.
  - A ***side effect*** is some change that results from executing the statement
- Example:

radius_int = 2

Part of A Program - Expression

- An expression is a combination of values and operations that creates a new value.
  - This new value is called return value.
- Enter an expression into the Python shell, a value will be returned and displayed.
- Example:

```
radius_int = 5
radius_int + 5
```

Common Pitfall – Statement & Expression

- Knowing that an expression has a value but a statement does not is useful.
- Print value generated by an expression:

```
print (x + 5)
```

- Print a statement:

```
print (y = x + 5)
```

  – Python generates an error

Parts of A Program - Whitespace

- When we type, we usually separate words with what is typically called ***whitespace***.
- Python counts as ***whitespace*** the following characters:
  - Space, tab, return, linefeed, formfeed and vertical tab.
- Python has the following ***rules*** about how *whitespace* is used:
  - Whitespace is **ignored** within both expression and statement.
  - Leading whitespace, whitespace at the beginning of a line, defines ***indentation***.
    - ***Indentation*** plays a special role in Python.
  - Blank lines are also considered to be whitespace, and the rule for blank lines is trivial: blank lines are allowed anywhere and are ignored.

Parts of A Program - Indentation

- ***Indentation*** is used by all programmers to make code more readable.
  - the indented code is grouped together, meaning those statements have some common purpose.
- ***Indentation*** in Python:
  - Python ***requires*** ***indentation*** for ***grouping***.
  - When a set of statement or expression needs to be grouped together, Python does so by a ***consistent indentation***.
- Python *requires* ***consistency*** *in whitespace indentation*.
  - If previous statements use an indentation of ***four spaces*** to group elements, then that must be done consistently throughout the program.

Python Code Layout

- Reading material: ***code_layout.pdf***

Part of A Program - Continuation

- Long lines of code can make reading code difficult.
- Python provide ways to make long lines more readable by splitting them.
  - Such splitting is called a ***continuation***.
- Indicate a continuation by placing a **backslash character** (\) at the end of a line.
  - A single line of code will continue onto another line.

Part of A Program - Comments

- Comment begins with a ***pound sign*** (#)
- Comments contribute nothing to the running of the program because Python ***ignores*** them.
- But comments are one important way to improve *readability*.