2. Modules, Scripts, and I/O

Topics:

- Script Mode
- Modules
- The `print` and `input` statements
- Formatting
- First look at importing stuff from other modules

The Windchill Calculation

Let's compute the windchill temperature given that the air temperature is T = 32F and the wind is W = 20mph.

Here is the formula courtesy of the National Weather Service:

$$W_{chill} = (35.74 + 0.6215 * T) + (-35.75 + 0.4275 * T) * W^{.16}$$

The formula only applies if T <= 50F and W>=3mph.

Use Python in Interactive Mode

```
 <<<Temp = 32
 <<<Wind = 20
 <<<A = 35.74
 <<<B = .6215
 <<<C = -35.75
 <<<D = .4275
 <<<e = .16
 <<<WC = (A+B*Temp)+(C+D*Temp)*Wind**e
 <<<print WC
19.9855841878
```

The `print` statement is used for displaying values in variables.

Quick Note on `print`

The line

```
   >>> print WC
```

results in the display of the value currently housed in the variable **WC**

More on the **print** statement later.

Motivating "Script Mode"

What is the new windchill if the wind is increased from 20mph to 30mph?

Looks like we have to type in the same sequence of statements. Tedious.

Wouldn't it be nice if we could store the sequence of statements in a file and then have Python "run the file" after we changed `Wind = 20` to `Wind = 30` ?

Script Mode

Instead of running Python in interactive mode, we run Python in script mode.

The code to be run (called a script) is entered into a file (called a module).

We then ask Python to "run the script".

What is a Module?

A module is a `.py` file that contains Python code.

In CS 1110, these are created using Komodo Edit.

The Module WindChill.py

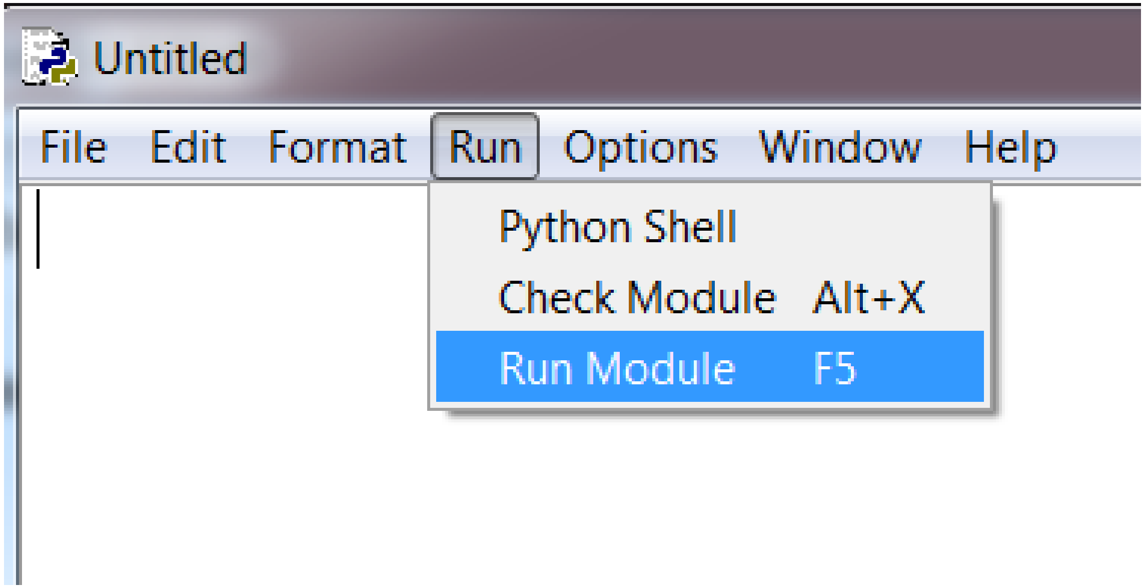
WindChill.py

```
Temp = 32
Wind = 20
A = 35.74
B = .6215
C = -35.74
D = .4275
e = .16
WC =  (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

Produced using Komodo Edit. This is our first draft.

Running the Module

1-



2- Press F5

Multiple Statements on a Line

Can put multiple statements on a line. Separate the statements with semicolons.

WindChill.py

```
Temp = 32
Wind = 20
A=35.74;B=.6215;C=-35.74;D=.4275;e=.16
WC = (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

For lecture slides we will sometimes do this to save space.
But in general, it makes for ``dense reading'' and should be avoided.

Module Readability: Comments

Comments begin with a “#”

WindChill.py

```
Temp = 32
Wind = 20
# Model Parameters
A=35.74;B=.6215;C=-35.74;D=.4275;e=.16
# Compute and display the windchill
WC = (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

Comments: Guidelines

Comments can also appear on the same line as a statement:

```
Wind = 20 # wind speed in miles-per-hour
```

Everything to the right of the “#” is part of the comment and not part of the program.

Module Readability: docstrings

A special comment at the top of the module.

WindChill.py

```
“““Computes windchill as a function of
   wind(mph)and temp (Fahrenheit).”””
Temp = 32
Wind = 20
# Model Parameters
A=35.74;B=.6215;C=-35.74;D=.4275;e=.16
# Compute and display the windchill
WC = (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

Trying Different Inputs

WindChill.py

```
“““Computes windchill as a function of
   wind(mph)and temp (Fahrenheit).”””
Temp = 32
Wind = 20
# Model Parameters
A=35.74;B=.6215;C=-35.74;D=.4275;e=.16
# Compute and display the windchill
WC = (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

Can we be more flexible here?

Temp and Wind via input

WindChill.py

```
“““Computes windchill as a function of
   wind(mph)and temp (Fahrenheit).”””

Temp = input(‘Enter temp (Fahrenheit):’)
Wind = input(‘Enter wind speed (mph):’)

# Model Parameters
A=35.74;B=.6215;C=-35.74;D=.4275;e=.16
# Compute and display the windchill
WC = (A+B*Temp)+(C+D*Temp)*Wind**e
print WC
```

The `input` Statement

The input statement is used to solicit values via the keyboard:

`input(` *< string that serves as a prompt >* **`)`**

Later we will learn how to input data from a file.

A Sample Run

The prompt is displayed…

```
>>> Enter temp (Fahrenheit) :
```

And you respond…

```
>>> Enter temp (Fahrenheit) : 15
```

A Sample Run

The next prompt is displayed…

```
> Enter wind speed (mph) :
```

And you respond again…

```
> Enter wind speed (mph) : 50
```

A Sample Overall “Dialog”

```
> python WindChill.py Enter
temp (Fahrenheit) : 15 Enter
wind speed (mph)        : 50
-9.79781580448
```

The `print` Statement

The `print` statement tries to intelligently format the results that it is asked to display.

`print` with formatting puts you in control.

Later we will learn how to direct output to a file

print Formatting

Script:

```
x = 2./5.
print  x
x = 1./3.
print x
x = 1234.5678901234
print x
```

Output:

```
0.4
0.333333333333
1234.56789012
```

For float values, print (by itself) displays up to 12 significant digits

print Formatting

Script:

```
x = 1234
y = 12345678
print x,y
```

Output:

```
1234 12345678
```

To display more then one value on a line, separate the references with commas.
A single blank is placed in between the displayed values.

A Final Example

Write a script to input the area of a circle and prints out the radius.

Preliminary Solution

Radius.py

```
A = input(‘Enter the circle area: ‘)
r = sqrt(A/3.14)
print r
```

The Math: solve A = pi*r*r for r.

We Get an Error

```
A = input(‘Enter the circle area: ‘)
r = sqrt(A/3.14)
print r
```

```
r = sqrt(A/3.14)
NameError: name 'sqrt' is not defined
```

sqrt is NOT a built-in function

Final Solution

Radius.py

```
from math import sqrt
A = input(‘Enter the circle area: ‘)
r = sqrt(A/3.14)
print ‘The radius is’, r
```

We are importing the function sqrt from the math module.

The Math: solve A = pi*r*r for r.

The Idea Behind `import`

People write useful code and place it in modules that can be accessed by others.

The **`import`** statement makes this possible.

One thing in the math module is the square root function **`sqrt`**.

If you want to use it in your module just say

```
from math import sqrt
```

Better Final Solution

Radius.py

```
from math import sqrt,pi
A = input(‘Enter the circle area:  ‘)
r = sqrt(A/pi)
print ‘The radius is’, r
```

We are importing the function sqrt and the constant pi from the math module.

Can import more than one thing from a module. Much more on import later.

Sample Run

```
Enter the circle area: 10
The radius is  1.785
```

For more insights, check out the lecture script **`SurfaceArea.py`**.