Python (version 3)

- general purpose
- high level language
- open source
- paradigms supported:
  - **procedural** <- CP104
  - object-oriented
  - functional (partial)
- uses dynamic typing

Python's language design philosophy is "there should be one -- and preferably only one -- obvious way to do it".
[T. Peters, PEP 20 – The Zen of Python , 2004]

Comments

Notes for programmers reading your code that explain how the program works. Comments are not executable code.

1. *comment line* begins with a # sign and goes to the end of the line, e.g.
   # calculate the tax using marginal rates

2. *end-line comments* follow a line of Python code, e.g.
   print (“Hello world!”) # always the first program

Comments *(not in text)*

3. *multi-line comments* use *document strings* which start and end with triple quotes. Single or double quotes can be used, e.g.

   """

   tax rates are based on a dollar figure per $1,000 dollars of assessed property value

   """

NOTE: Comments can be used to turn code on/off. Known as "commenting out" code.

Working with Text

- a sequence of characters (letters, numbers, symbols) that Python calls a *string*
- in your program, indicate that data is a string by using single or double quotes (*string literal*)
  - e.g. ″Introduction to Programming″
  - e.g. ′CP104′
  - e.g. ″519-884-0710″
- opening/closing quotes must match

Working with Text

- ‘how’s this going to work?’
- “ “what about this”, she said.”
- """ or
  this?"""

Output

Our programs will typically output results to the screen (the console).

The Python print function takes many forms

print (*argument1* [, *argument 2*, …])

if the argument is a string, it will print the string

e.g. print (“Hello World!”)

e.g. print (“Here’s the information”)

All of these examples are <u>unformatted output</u>.

Input

Programs are more general if you get the user to enter data via the keyboard.

The Python input function

variable = input(*prompt*)

1. prints the *prompt* (literal string)
2. reads data from keyboard when entered by user (data entry complete when user presses the *enter* key)
3. puts the data in *variable* for later use

Input

e.g. name = input (”Enter your name ”)

e.g. age = input (”Enter your age ”)

e.g. age = input () # NOT user friendly

Assignment Statements

Assignment statements have the form

variable = *expression*

- *expression* is evaluated (*expression* is a value or code that results in a value)
- the value is assigned to *variable*
  - if variable exists, the current value is overwritten by the new value
  - if variable does not exist, a new memory location is allocated and value is written to that memory location

Sample Python program

```
"""
-----------------------------------------------------
greetings.py
Prompt for name and greet
-----------------------------------------------------
Author:  Nora Znotinas
ID:
Email:   nznotinas@wlu.ca
Version: 2014-09-03
-----------------------------------------------------
"""
name = input ( "Please enter your name: " )
print ( "Pleased to meet you " )
print ( name )
```