Python Basic Syntax

Basic Syntax - First Program 1

- All python files will have extension **.py**
- put the following source code in a test.py file.
- `print "Hello, Python!";#hello world program`
- run this program as follows:
- $ python test.py
- This will produce the following result:
- Hello, Python!
- (or you can run it from eclipse IDE)

First Program 2

- All python files will have extension **.py**
- What error message will you get if you save the file as test.txt?
- `print "Hello, Python!";#hello world program`
- What if you mistype print?
- Or miss a “
- Do you need the “;”?
- (try to learn by **OBSERVING** your mistakes)

Python Identifiers 1

- Identify (name) a variable, function, class, module or other object.
- An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
- What error messages will you get if you do not follow this?

Python Identifiers 2

- Python does not allow punctuation characters such as @, $ and % within identifiers.
- Python is a case sensitive programming language.
- Thus, **Manpower** and **manpower** are two different identifiers in Python.
- (is this the same in Windows/Unix?)

Python Identifiers 3

- Class names start with an **uppercase** letter and all other identifiers with a lowercase letter.
- Starting an identifier with a single leading underscore indicates by convention that the identifier is **meant to be private**.
- Starting an identifier with two leading underscores indicates a **strongly private identifier**.
- If the identifier also ends with two trailing underscores, the identifier is a **language-defined special name**.

Reserved Words

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Lines and Indentation

- No braces to **indicate blocks of code** for class and function definitions or flow control.
- Blocks of code are denoted by **line indentation**, which is rigidly enforced.
- The number of spaces in the indentation is variable, but **all statements within the block must be indented the same amount**. Both blocks in this example are fine:

Correct Indentation

- `if True:`
- `        print "True"`
- `else:`
- `    print "False“`
- Or even better
- `if True:`
- `        print "True"`
- `else:`
- `        print "False“`

Incorrect Indentation

- `if True:`
- `        print "Answer"`
- `print "True"`
- `else:`
- `        print "Answer"`
- `    print "False”`
- (what will the error message be?)
- Try a few different example to understand.

What about this?

- `if True:`
- `    print "Answer"`
- `    print "True"`
- `else:`
- `    print "Answer"`
- `print "False"`

Multi-Line Statements

- `total = item_one + \`
- `            item_two + \`
- `            item_three`
- Statements contained within the [], {} or () brackets do not need to use the line continuation character.
- `days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']`

Quotation in Python

- Python accepts single ('), double (") and triple (''' or """) quotes to denote string literals as long as the same type of quote starts and ends the string.
- The triple quotes can be used to span the string across multiple lines. E.g.
- `word = 'word'`
- `sentence = "This is a sentence."`
- `paragraph = """This is a paragraph. It is`
- `made up of multiple lines and sentences. """`
- error message if you mixed quotes???

Comments in Python – single line

- All characters after the # and up to the physical line end are part of the comment and the Python interpreter ignores them.
- `# First comment`
- `print "Hello, Python!";  # second comment`

Commenting multiple lines

- All of the lines below are ignored by the interpreter
- *"""*
- *you can comment multiple lines*
- *like this*
- *""“*

Waiting for the User

- The following line of the program displays the prompt, “Press the enter key to exit” and waits for the user to press the Enter key:
- `raw_input("\n\nPress the enter key to exit.")`
- Here, "\n\n" are being used to create two new lines before displaying the actual line.
- Once the user presses the key, the program ends.
- This is a nice trick to keep a console window open until the user is done with an application.

Multiple Statements on a Single Line

- The semicolon (;) allows multiple statements on the single line given that neither statement starts a new code block. Here is a sample snip using the semicolon:
- `import sys; x = 'foo';`
- The same as
- `import sys;`
- `x = 'foo';`

Multiple Statement Groups as Suites

- A group of individual statements, which make a single code block are called **suites** in Python. **Compound or complex statements**, such as if, while, def, and class, are those which **require a header line and a suite**.
- Header lines begin the statement (with the keyword) and terminate with a colon (:) and are followed by one or more lines which make up the suite. For example:

For example:

```
if expression :
    suite
elif expression :
    suite
else :
    suite
```