Lecture I
Intro

- What is Python?
- Dynamic Typing
- Why Python?
- The Interpreter
- IDEs
- Examples

python

What is Python?

- A very high level, multi-paradigm, dynamic programming language.
- Invented in 1989 by Guido van Rossum.
- Still in active development.
- Emphasizes simple, explicit and elegant code.
- Used for scripting and as a general purpose programming language.
- Eschews speed for ease of use.

python

Dynamic Typing

- Variables are not defined but come into existence when first assigned to.
- The type of the object to which a variable refers may change.
- Type checking is performed during runtime rather than compile time.
- Easier to use than static typing, but sometimes harder to debug.

python

Why Python?

- Very High Level
  - A programming language is low level when its programs require attention to the irrelevant.
- Powerful
  - Type less code, get more done.
- Huge Standard Library
  - File management, GUI, Graphics, Network, Web, Email, Compression, Cryptography, Threading, Multimedia, Internationalization, Testing, and much more.

python

Why Python?

- Portable
  - Run the same code on Windows, UNIX, Linux, Mac OS and even Symbian OS.
- FOSS
  - You'll never have to pay for a license.
  - If you find a bug in the language, you have the source code to fix it.
- Interoperability
  - It's easy to embed Python into C++, Java or .NET applications.

The Interpreter

- Python can compile and run source files, but it can also run as a command interpreter console.
  - Type in commands and have them execute immediately.
  - Type in an expression to see its value.
  - Test your code while writing it.
  - Look up documentation using the help() function and try out code without separate compile steps and without leaving your IDE.

python

IDEs

- Usually consists of an editor, a command interpreter window and a debugger.
- Sometimes includes library/object browsers.
- Many different free IDEs exist for each of the supported operating systems.
- Examples will be shown using PythonWin on Windows, but any other IDE/OS combination will work just as well.

Example: Hello World

```
print 'Hello World!'
```

python

Example: Sum

```
s = 0
for i in range(100):
    s += i
print s
```

```
print sum(range(100))
```

Example: Triangle

```
levels = 10

for i in range(1, levels):
    print ' ' * (levels - i), "*" * 2 * i
```

Example: Input

```
name = raw_input('What is your name?\n')
print('Hi, ' + name + '!')
```

Example: Fibonacci

```
def fib_recursive(n):
    if n <= 1:
        return n
    else:
        return fib_recursive(n-1) + fib_recursive(n-2)
print fib_recursive(5)
```

```
def fib_iterative(n):
    i = 0
    j = 1
    for dummy in range(n):
        temp = i
        i = j + i
        j = temp
    return i
print fib_iterative(5)
```

python

Example: Lists

```
pi = 3.14159
fibonacci_sequence = [0, 1, 1, 2, 3, 5, 8, 13]
stuff_i_like = [1, pi, 'apples', fibonacci_sequence]

for item in stuff_i_like:
    print 'I like:', item
    print 'Two of that make:', item * 2
    print
```

python

Example: Strings

```
sentence = raw_input('Please enter a sentence: ')

words = sentence.split()

print 'There are %d words in your sentence.' % len(words)

message = 'The word "%s" starts with "%s".'
message2 = 'The word "%s" contains the letters: %s.'

for word in words:
    print message % (word, word[0].upper())

    letters = set(word)
    letters_formatted = ', '.join(letters)
    print message2 % (word, letters_formatted)
```