Algorithms for Graphical Models (AGM)

Python: Basics

$Date: 2008/10/15 15:25:51 $

AGM-02

Learn Python in an afternoon?

- The key first step is to find the documentation that comes with a Python distribution.

- They are in this (standard) location:
  `/usr/doc/python-2.5.2/html/Python-Docs-2.5.2/index.html`

- A good way to get to grips with Python is plod through the Tutorial there.

Once this lecture is over

- These slides are just prompts for demos.
- Afterwards go to
  `http://www-users.cs.york.ac.uk/~jc/teaching/ipp/`
- Lecture slides, two online books, and example programs.

Writing and executing Python code

- Using a suitable text editor (preferably one equipped with a Python mode) save your code to a file `somefile.py`

- To run it do `python somefile.py`

Python as scripting language

- Python programs are scripts. There is no 'main'. Execution starts from the top line of the code and just works down.

- Of course, the first step might be to define a function which only gets called later.

Python as intepreter

- Typing `python` at the command prompt brings up the Python interpreter.

- Use this to try stuff out, and to get to the builtin `help` function.

Built in data types

- Float, complex, int
- List (mutable), tuple (immutable), string (immutable), Unicode strings (immutable)
- Dictionaries (mutable), set (mutable), frozenset (immutable)
- Explore them with the interpreter. Lots of builtin *methods* for these.

Typing

- The type of an object is determined by assignment
- `x=2` An integer object with the name `x`
- `x='foo'` An entirely different string object with the same name
- Can't use `x` to get to the '2' any longer.

Python syntax

- Uses *indentation* to make blocks, (and the occasional colon)
- Even when using the interpreter
- You get used to it!

Conditionals, loops

- All pretty standard: `if`, `elif`, `else`
- `for` loops are not like C ones . . .
- Have `while`, `break`, `continue` as per normal

Defining functions

- Indentation as always . . .

- A function definition `def f(x): ...` assigns a function object to the name `f`

- Objects created in a function are local to that function. Are allowed to access non-local objects.

- Since the program is a script, can't call a function before it is defined.

Exception handling

- Use `try: ... except SomeException: ...` to handle exceptions

- `raise` for raising them.

I/O

- Done using file objects (which are builtin).
- Builtin function `open` creates them.