

1MD3 Tutorial 2

January 22, 2006

1 Introduction to IDLE

1. **Demo:** Open file, Edit program.
2. **Demo:** Using interpreter, loading a module, calling functions

```
>>> import test
>>> test.dec2bin(23)
'10111'
```

3. Demo: The debug feature in IDLE

- (a) MainWindow → Debug → Debugger
- (b) Turn on all the options, Stack, Locals, Source and Globals
 - *Source* displays the source code in another window.
 - *Locals* shows all local variables in a function.
 - *Globals* shows all module attributes, functions and variables defined in the module.
 - *Stack* shows function call stack.
 - **Demo:** Demonstrate the above debug features by calling function `test.fact(4)` in the interpreter window, then step through it, notice the changes in “Stack”, “Locals” and source window.

2 Function in Python

- **Syntax:** `def funName(argu1, argu2, ..., argun):`
`Statement1`
`...`
`Statementn`
- In Python a function always returns an object. Explicitly returning an object uses the keyword *return*. If there is no *return* keyword, a NULL object is returned.
- **Recursion:** A function is directly or indirectly defined by itself.
Example:

```
def dec2bin2(n):
    if n == 0:
        return ""
    else :
        if n % 2 == 0:
            a = "0"
        else:
            a = "1"
        return dec2bin2(n / 2) + a
```

- **High order function:** A function takes or returns a function.
- **Lambda Expression:**

$$\lambda x. fx$$

In Python, `lambda x: f(x)`

Informally speaking, lambda expression is a function without a name, in Python is quite useful when one wants to return a function in a function.

Exercise: Write a function which takes a function with two parameters returns a function which is exactly the same as the input function except it flips the position of the two parameters. Such that, `flip(f)(x,y) = f(y,x)` for all input x and y .