

582332: Programming in Python, Fall 2009

Exercise sheet 1
7-11.9.2009

1. First program in Python. Write the following program into a file named `hello.py` using your favourite text editor.

```
print "Hello, world!"
```

Run the program with command `python hello.py` in the command shell or terminal program.

2. Use Python interactively. Start the interpreter with command `python`. Now type the statement `print "Hello world!"` at the *prompt* (`>>>`). Python executes the statement, shows the result, and gives the prompt again where you can type additional *statements* or *expressions* (these are defined in the lectures).

Use up-arrow button to retrieve your previous statement and edit your statement a bit: make it print `Hello again world!`.

Use `quit()` to stop the interpreter. Alternatively, you can do this by pressing control-d (Unix) or control-z (Windows).

3. Running your program directly. Have the file `hello.py`, which you used earlier, ready. Windows should automatically associate Python with the extension `.py`, so typing `hello.py` at the command prompt should run your script.

On Unix, you can associate your script with Python by adding line `#!/usr/bin/env python` as the first line of the program. In addition, you must give execution rights to the file by the Unix command `chmod u+x hello.py`. Now you should be able to run the script directly from a Unix shell with command `./hello.py`.

4. Use Python interpreter as a calculator. Start the interpreter with command `python` and try some trivial calculations, like `6/2+1`. What is the result of `1/3`? Why?

Most mathematical functions aren't directly available, for example `cos(3.14)` won't work. To get it to work, the function needs to be loaded from the *standard library* of Python. Use command `from math import cos, sin` first, and then play with the trigonometric functions *sine* and *cosine*.

5. Python has a builtin documentation system, which can be accessed with the `help(name)` command. To get help with the cosine function, try `help(cos)` (press 'q' to exit the help system).

If you encounter a function (or other name) unknown to you, use the help system. For example, run first the command `dir()`, peruse the result of the command, and then check what the help system can tell about the command. It may be a bit cryptic at first if you don't know the Python terminology. Can you see any connections between the result of the command and the earlier `from math import` commands you used?

Try to get help for the `print` statement. Note that now you need to use different syntax because `print` is one of the 30 *keywords* of Python, which have special meaning.

6. Playing with variables. You can assign a value to a *variable* named `a` with the following statement: `a=5`. Check the value of the variable by typing just `a`. Change the value of the variable by assigning it the expression `(10+2)/3`, and check the value again. Print the current scope using function `dir()`. Delete the name (and the associated value) with `del a` statement. Verify with `dir()` that the variable has disappeared.

7. What does the following two-line piece of code do? Try it out in the interpreter.

```
for i in [0,1,2,3,4,5,6,7,8,9]:  
    print "Square is", i*i
```

Note that the second line begins with four space characters. This is called *indentation*. Python's interactive interpreter and some text editors automatically produce a sequence of spaces when you press the tabulator key.

8. On the Linux machines of the department of Computer Science two alternative interfaces for Python are installed. *IDLE* is a graphical interface to Python, and *IPython* is a textual interface with many extra features compared to the usual Python interpreter.

Try at least to start and exit these programs, just to know they exist. The respective commands are `idle` and `ipython`. If you are interested and you have time, check out what features they offer.