

online help for programmers (NOTE: Employers are interested to know whether you're an active member of this site!).

'<http://www.draw.io>' - A very good, free online drawing tool that exports to many formats, including 'XML' and 'JPG'.

Aim & Objectives

- Re-cap and review difficult test questions
- Questions & Answers on previous sheet
- Work through lecture related exercises

Introduction

In this session we'll be looking at procedural programming in more detail, which includes the idea of looking at "sub-routines", also well known as functions.

Exercise - Simple example

The first exercise will involve creating a really pointless program that just highlights how functions work. Create a program that carries out the below task:

```
DISPLAY "Print before function"  
DISPLAY "Hello from function" -- Call this  
DISPLAY from another method  
DISPLAY "Print after function"
```

The following is a clue on how to do this:

```
def func():  
    # Do something here  
    # Remember the function returns after the  
    # indentation finishes!  
  
# Call function  
func()
```

Remember that indentation is extremely important here!

Exercise - Pass me that...

Now we're going to give our functions some value - literally in this case. The function you'll write will be simple and will add will find the long side of a right angle triangle! The equation is $\sqrt{a^2 + b^2} = c$ and in Python this looks like '`math.sqrt((a * a) + (b * b))`'. Don't forget to '`import math`' to use the math class.

Your code should have a function called '`missingSide`' and should take two parameters, '`a`' and '`b`'. Search on the internet "*python 3 parameters to function*" for help on how to add arguments to your function.

To test it works, find the value of '`a = 3`', '`b = 4`' - the value should be an integer and not have any numbers after the decimal point.

Exercise - Experiments & Research

So, you're getting the hang of these function things eh? What happens when we start mixing this up a little? Try out the following experiments:

- What happens if you're function calls itself?
- Find out what it's called when a function calls itself.
- Research how many times a function can call itself. Can you check this with a small bit of code?

Resources & Further Reading

'<http://homepages.herts.ac.uk/~db12aba/>' - All content from these sessions updated weekly.

'<http://code.org/>' - A good resource testing your programming skills.

'<http://stackoverflow.com/>' - Highly recommended