

Getting started with Python

Summary of elementary commands/constructs

Mathematical operations and 'print'

The `%` command return the remainder upon division.

```
print 8 + 2
print 8/2
print 8 * 2
print 8 - 2
print 8**2
print 11 % 3
print "I love maths!"
```

```
10
4
16
6
64
2
I love maths!
```

Variables and integers/decimals

You might find the second value printed in this example surprising.

You need to tell Python to treat a variable as something which doesn't need to be a whole number. You can do this by including a decimal point when you define it.

```
apples = 4
pears = 2
bananas = apples/pears
print bananas
oranges = 5
print oranges/pears
grapes = 5.0
print grapes/pears
```

```
2
2
2.5
```

'for' loops

One of the first things to learn about Python is the way that it uses colons and indentation to structure code. This is shown for a 'for' loop on the right. This 'for' loop runs some code for a specified range of values of a variable increasing the variable by 1 each time that code is looped through.

```
for i in range (1,11):
    print i
```

```
1
2
3
4
5
6
7
8
9
10
```

MEI FPT Enrichment Materials

'if/else' 'and', 'or', 'not'

You don't need to have an 'else' part with an 'if' statement. You can join conditions together and negate them using 'and', 'or' and 'not'. Notice the use of colon, indentation and "==" here.

```
for i in range (1,11):  
    if i % 2 == 0 and i % 3 == 0:  
        print i  
    else:  
        print "not a multiple of 6"
```

```
>  
not a multiple of 6  
not a multiple of 6  
not a multiple of 6  
not a multiple of 6  
not a multiple of 6  
6  
not a multiple of 6  
not a multiple of 6  
not a multiple of 6  
not a multiple of 6
```

'while' loops

'while' allows looping until a certain condition is not satisfied.

```
count = 1  
  
while count < 100:  
    print count  
    count = count*2
```

```
>  
1  
2  
4  
8  
16  
32  
64  
>
```

A function with no input or output

Use 'def' to define a function. () means this function won't expect any input.

```
def sayhello():  
    print "hello"  
  
sayhello()  
sayhello()
```

```
>  
hello  
hello
```

A function with inputs but no output

Notice you can define functions with more than one input.

```
def sum(m,n):  
    print m + n  
  
sum(3,5)  
sum(10,20)
```

```
>  
8  
30
```

A function with inputs and an output (returned value)

Use 'return' if you want your function to send something back.

```
def sum(m,n):  
    return m + n  
  
print sum(3,5)  
print sum(10,sum(20,30))
```

```
>  
8  
60
```