

---

## Using python with Lout

---

This document show how python and its libraries can be used in the context of Lout to ease plotting of functions.

### Defining a @Python symbol

Put at the beginning of your document the follong definition

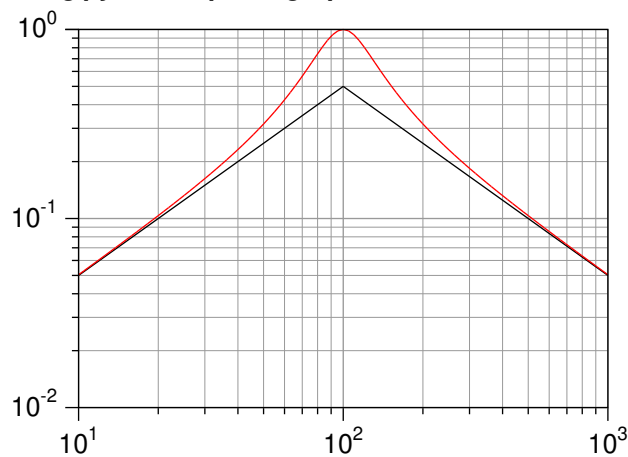
```
import @BasicSetup
def @Python
    right x
{
    def @Filter {python < @FilterIn > @FilterOut}
    x
}
```

### Calling @Python

```
@I @Python{
# my python code
print("Hello World!")
for i in range(0,5):
    print(i)
}
```

which produces: *Hello World! 0 1 2 3 4*

### Using python to plot a graph



```
@Graph
width{7c} height{5c}
style{frame} xextra{0c} yextra{0c}
grid{yes} label{yes}
xlog{10} ylog{10}
{
    @Data pairs{solid} color{black}{
        10 0.05 100 0.5 1000 0.05
    }
    @Data pairs{solid} color{red} {
        @Python {
            from numpy import *
            j=complex(0,1)
            # define the frequency vector
            f=10**linspace(1,3,51)
            m=0.25;f0=100
            # Calculate the magnitude of the
            # transfer function
            T=abs(1/(1+j/(2*m)*(f/f0-f0/f)))
            # print the points to stdout
            for i in range(f.size):
                print("%g %g " % (f[i],T[i]))
            }
        }
    }
}
```

Python executes the specified code and returns the result on its standard output (the calculated points) back to Lout.