# Formatted Strings

**Formatted Strings** have the form

*pattern* %(*values*)

The pattern is allowed to have placeholders:

- %d is a placeholder for an integer
- %s is a placeholder for a string
- %f is a placeholder for a float

The placeholders get their values from the list of values. For example if variable s is "Mom" and variable n is 5

```
"Send %s %d flowers"%(s,n)
```

is

```
"Send Mom 5 flowers"
```

The print statement in fancy.py in Lab 1 could have been written

```
print( 'Welcome back, %s "%s" %s!' %(first, nick, last))
```

Placeholders can even assign fieldwidths to their values. Placeholder %5d says to use 5 spaces for whatever value goes in for this placeholder, and pad with blanks if it needs less than 5. If you just say print( x, y, z) twice and the first time the values are 1, 2, 3 and the second time 100, 200, 300, the output looks like

```
        1 2 3
        100 200 300
```

If your print statement is

```
        print( "%5d  %5d  %5d"%(x, y, z))
```

your  output will be

```
    1    2    3
  100  200  300
```

Your output is coming out in columns!

The float placeholder %f can even specify how many decimal places to use:

```
%6.3f
```

says to use at least 6 spaces for the float, with 3 after the decimal point.

If we say print( "pi is %6.3f" % 3.1415926535 )
it will actually print

```
pi is  3.142
```