Print statement

Displays output in our notebook.

Print a single number

In [1]:
```
print 5
```

```
5
```

Print words, in quoted strings

In [2]:
```
print 'Hello World!'
```

```
Hello World!
```

You can also print variables, coming up next.

Variables

Like variables in math: you can store numbers

In [3]:
```
x = 12.3
print x
```

```
12.3
```

You can store words

In [4]:
```
word = 'petal width'
print word
```

```
petal width
```

Lists

A list hold a sequence of variables all at once. You make a list of values in square brackets separated by commas. You can print them if they are short enough.

In [5]:
```
my_lst = [7, 8, 9, 10, 42]
print my_lst
```

```
[7, 8, 9, 10, 42]
```

You can access individual variables directly with indexes starting at 0 and ending the length - 1.

In [6]:
```
print my_lst[1]
```

```
8
```

`for` loops

`for` loops iterate over every element of a list (and many other data structures) in turn.

This colon and indentation scheme is how Python keeps track of blocks of code that contain multiple lines, and is seen throughout the language.

In [23]:
```
for x in my_lst:
    print x * 2
```

```
14
16
18
20
84
```

`import` necessary software

Sometimes we need code we didn't write ourselves. So we import it from elsewhere. NumPy can generate random numbers for us, so first we need to import it.

In [7]:
```
from numpy import random
```

Then we can make a NumPy array (similar to the list we just discussed) with 5 random values.

In [8]:
```
random_numbers = random.rand(5)
print random_numbers
```

```
[ 0.98997924  0.8245293   0.25105814  0.85119907  0.72772861]
```

Matplotlib

- Is created to make data visualizations
- Creates histograms, scatter, line chart, and many more.

First we need to set it up to use in Jupyter

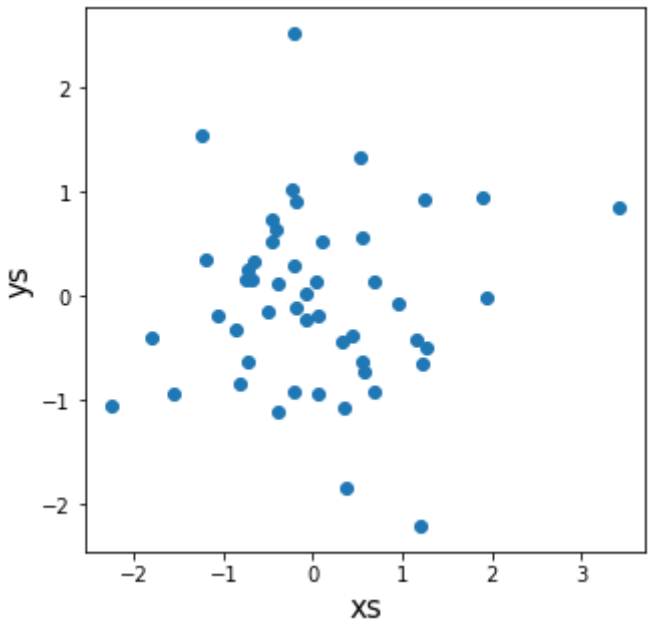
In [9]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

Simple scatter plot: the code

Let's make a scatter plot, which just prints x, y pairs. Both axes are normal distributions with means of μ=0, and standard deviation σ=1, with 50 numbers each.

In [24]:
```
xs = random.normal(0, 1, 50)
ys = random.normal(0, 1, 50)
fig = plt.figure(figsize=(5, 5))
ax = fig.add_subplot(1, 1, 1)
ax.scatter(x=xs, y=ys)
ax.set_xlabel('xs', size=15)
ax.set_ylabel('ys', size=15)
```

Out[24]:
```
<matplotlib.text.Text at 0x11a51f190>
```



Try out changing the standard deviation (second argument) and hit `Shift+Enter` a few times.

Summary of some Python

- `print`: Displays output in the notebooks.
- Variables: Like math variables they hold values. They can hold numbers or quoted words.
- Lists: hold multiple values in a collection.
- `import`: Grabs software for use in our notebooks.
- matplotlib: One tool to make data visualizations.

In [ ]:
```
```