Notes for this week

Outline from the doc

- Lab
  - Tic-tac-toe 2D with list of lists (move to prev week)
  - More advanced NumPy, slicing, selecting, dtypes, vectorization
  - Basic plotting
- Class
  - quiz, book chapters 1 and 2
  - Debugging and timing code in IPython
  - Introduction to NumPy
  - Vectorized computation vs. Python loops
  - Slicing NumPy arrays
  - (2 and 3 dimensional arrays and NumPy broadcasting)
  - Practice basic NumPy manipulations and contrasting arrays with lists

In [2]:
```
import numpy as np
```

COMP 3122 - Artificial Intelligence with Python

Week 2

github.com/kamrik/ML1

The plan for today

- Using Jupyter
- Creating, slicing and reshaping numpy arrays
- Basic data plotting

Extra material

- Making data mean more through storytelling by Ben Wellington at TEDxBroadway (15 min video)
- All About Jupyter by Brian Granger - PyData NYC 2015 conference talk (40 min video)

Book section numbering

- The book table of contents has chapter numbers but no section numbers
- But you can see section numbers in the URL of each section (hover over the link)
- You can also see the number in the listing of files in notebooks folder on GitHub
- Example: **02.02-The-Basics-Of-NumPy-Arrays.ipynb**
- Sections X.00 (under the large heading of the chapter)

Book sections for this week

TBD

- Chapter 2. Introduction to NumPy, sections 2.1 - 2.5
- Chapter 4. Visualization with Matplotlib

Jupyter - quick recap

- A document editable in the browser
- Consists of 2 types of cells: Code cells and Text cells
- Saved as `*.ipynb` files, but can be exported/converted to many static formats like HTML, PDF etc

Two ways of reading the book

- Static at https://jakevdp.github.io/PythonDataScienceHandbook/
- Download from github and run as notebook, then you can modify and run the code samples
- Downloading the repo: either `git clone` it, or download as ZIP (screenshot)

Downloading and running a single notebook locally

- Download the [exercise notebook]
  - Either clone the repo or
  - Click Raw (screenshot) and then File->Save as)
- Make sure to place it under the user's home directory
- Run Jupyter (note, it can only see files below the dir where it runs, by defaul the user's home dir)
- Edit the notebook and complete tasks there

Array arithmetic

- Book section 2.3 "Computation on NumPy Arrays: Universal Functions"
- This part is rather different from Java or C# arrays - loops are extermely rare

Operations between an array and a single number

In [3]:
```
x = np.arange(7)
print("x     =", x)
print("x + 5 =", x + 5)
print("x - 5 =", x - 5)
print("x * 2 =", x * 2)
print("x / 2 =", x / 2)
print("x % 3 =", x % 3)  # floor division
```

```
x     = [0 1 2 3 4 5 6]
x + 5 = [ 5  6  7  8  9 10 11]
x - 5 = [-5 -4 -3 -2 -1  0  1]
x * 2 = [ 0  2  4  6  8 10 12]
x / 2 = [0.  0.5 1.  1.5 2.  2.5 3. ]
x % 3 = [0 1 2 0 1 2 0]
```

Operations between two arrays

In [4]:
```
y = np.arange(7, 0, -1)
print(x)
print(y)
```

```
[0 1 2 3 4 5 6]
[7 6 5 4 3 2 1]
```

In [5]:
```
# Element-wise operations
x + y
```

Out[5]:
```
array([7, 7, 7, 7, 7, 7, 7])
```

Lab exercise

- Download and run exercises/numpy_basics.ipynb notebook from GitHub
- Read the first section with examples
- Complete the tasks below the "Exercises" heading as you read book section 2.2 (it's pretty short)

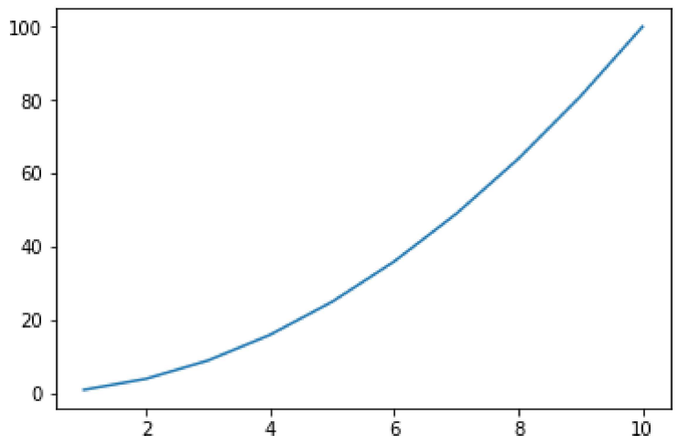
Plotting

In [6]:
```
import matplotlib.pyplot as plt
```

In [7]:
```
%matplotlib inline
```

In [10]:
```
x = np.arange(1,11)
plt.plot(x, x**2)
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x21ad157b630>]
```



In [ ]:
```
```

In [12]:
```
import seaborn as sns
mpg = sns.load_dataset('mpg')
```

In [13]:
```
mpg.head()
```

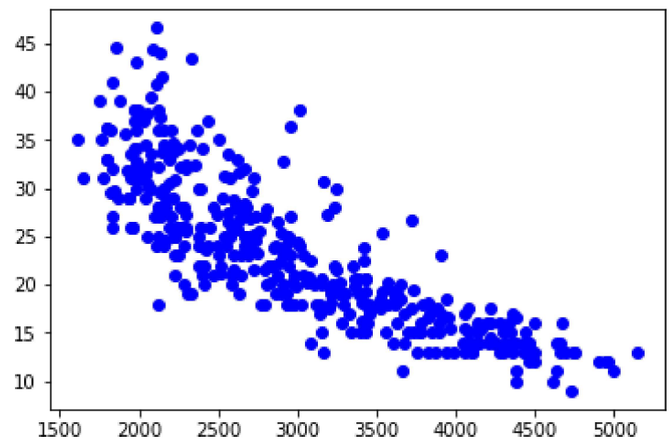
Out[13]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

In [18]:
```
year = mpg['model_year'].values
fuel = mpg['mpg'].values
weight = mpg['weight'].values
```

In [24]:
```
plt.plot(weight, fuel, 'bo')
```

Out[24]:
```
[<matplotlib.lines.Line2D at 0x21ad4447208>]
```



Format parameter

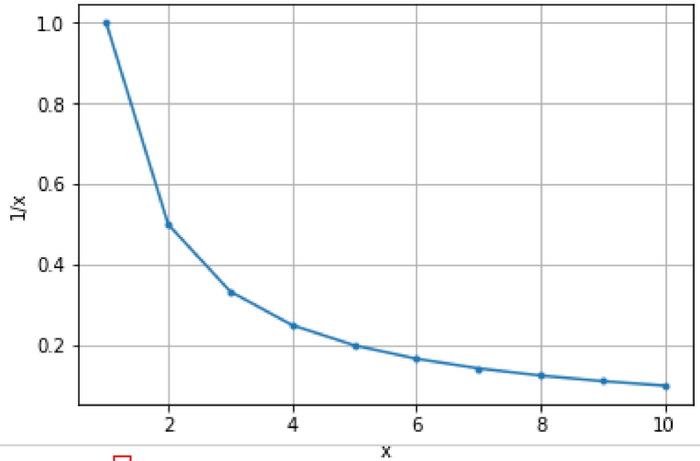
The optional parameter *fmt* is a convenient way for defining basic formatting like color, marker and linestyle.

- `plot(x, y)` # plot x and y using default line style and color
- `plot(x, y, 'bo')` # plot x and y using blue circle markers
- `plot(y)` # plot y using x as index array 0..N-1
- `plot(y, 'r+')` # ditto, but with red plusses

Full list under "Format strings" here

Labels and legends

In [29]:
```
plt.plot(x, 1/x, '.-')
plt.xlabel('x')
plt.ylabel('1/x')
plt.grid()
```



In [25]:
```
plt.plot?
```

(optional) Demo of a Jupyter notbook

TBD - maybe move to the lecture

- NYC Taxi data - https://anaconda.org/jbednar/nyc_taxi/notebook