An overview of the Anaconda Python distribution

Cheat sheet: https://conda.io/docs/_downloads/conda-cheatsheet.pdf Help pages: https://conda.io/docs/user-guide/index.html

Create virtual python environment

Allows you to have numerous python 'versions' on one machine without breaking each other!

```
conda create -n yourenvname_27 python=2.7 numpy
conda create -n yourenvname_35 python=3.5 numpy
activate yourenvname_35 # windows
#-or-
source activate yourenvname_35 # unix
```

NB/ Watch out for 32-bit and 64-bit installs e.g. when using arcpy and ArGIS <= v10.3, you need to have python 2.7

List existing Anaconda environments

You can list your environments using:

```
conda env list
```

Install a package

Let's install pandas to the currently activate version of Anaconda python that you are using:

```
conda install pandas
```

Running this will result in Anaconda checking the cuirrent environemnt and then offering to install/upgrade/downgrade other packages that the package you are trying to install requires - if you don't want certain upgrades/downgradesto be made then it is best that you create a new Anaconda environemtn to isntall the packge into. This prevents you breaking anything and a major benefit of using Anconda in the first place!

If you already have a package, you can update it using:

```
conda update pandas
```

iPython - an interactive command line python interface is also a package that needs to be installed for each anaocnda environment that you have:

```
conda install ipython
```

Jupyter notebooks must also be installed for each conda environment:

```
conda install jupyter
```

To learn more about working with jupyter notebooks, look here:
https://jupyter.readthedocs.io/en/latest/running.html#running

Installing a specific build of a package

Find what versions of numpy are available to Anaconda:

```
conda search numpy # << will give a long list....
```

Now install the version you want - note that this adds it to the currently activate version of Anaconda python that you are using:

```
# syntax: conda install <package_name>=<version>=<build_string>
conda install numpy=1.11.3=py36hb60be0b_3 # << notice that this is for python 3.7
conda install numpy=1.11.3=py27hab9e983_3 # << notice that this is for python 2.7
```

Export and recreate an environment

Particularly useful when switching between machines etc.

```
conda env export > environment.yml
conda env create -f environment.yml # name of the build is taken from the top of the yml file
(so will match whatever the yml was created from)
```

Remove an environment

```
conda remove --name arc_conda --all
```

Can I still use pip to install things?

Yes! Just make sure you are in the enviornment that you want to install the programme to i.e. `activate yourenvname`