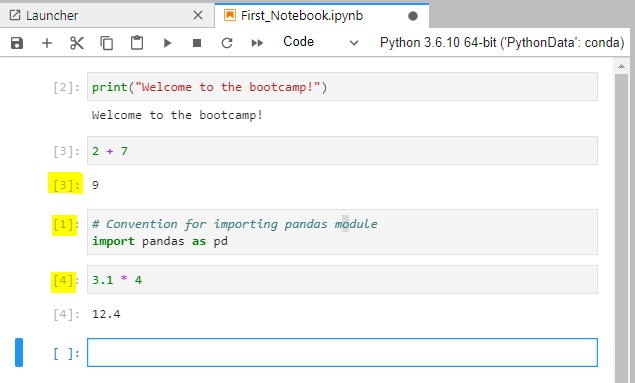
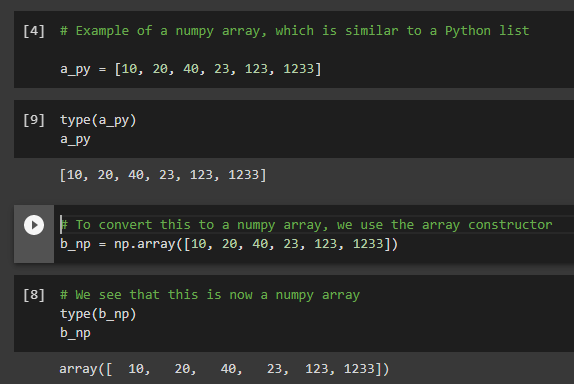
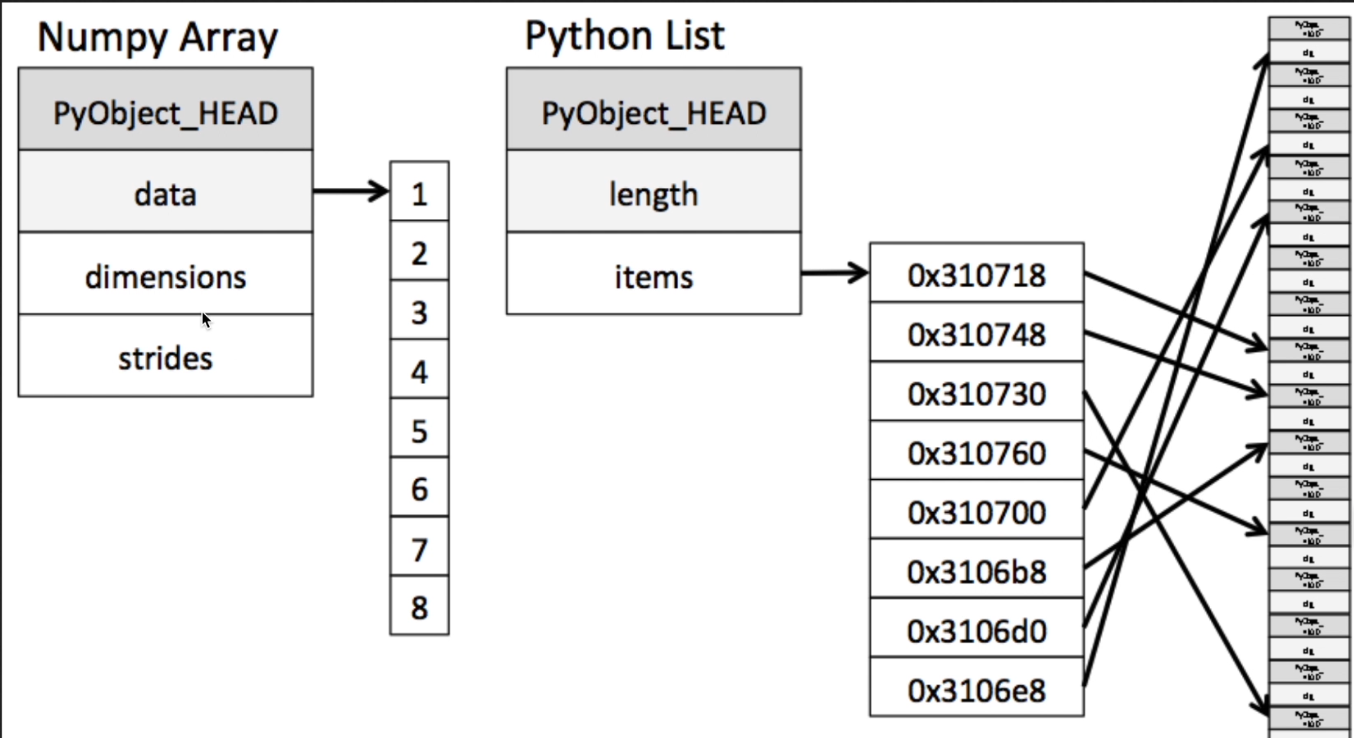
* Jupyter notebooks and JupyterLab
  + Convenient interface for writing and executing Python code in a web-based environment
  + Notebooks consist of individual cells in which Python code can be written. Cells are executed independently from one-another
  + Bracketed numbers to the left of cells indicate the order in which the code was executed in the notebook. This need not be the same as the visual ordering of the cells
    - Only cells containing Python code will be numbered



* + Cells can be used for Python code, raw text, or Markdown language
  + JupyterLab keyboard shortcuts: <https://nocomplexity.com/documents/jupyterlab/keyboardshortcuts.html>
* Cloud hosted data science environments
  + Almost all cloud service providers today offer some sort of cloud-hosted Jupyter notebooks
    - Amazon web services Sagemaker
    - Microsoft Azure
    - IBM Watson Studio
    - Google Colab – instructor’s personal favorite
  + These services allow us to write all sorts of Python code within our browsers without worrying about installation and dependencies
* **Numpy** (or numerical python) is a scientific computing package for Python
  + Pandas relies on Numpy behind the scenes for its data structures
  + Check out this Numpy array, compared to a Python list



* + Numpy arrays are similar to Python lists, but have a lot of distinct benefits
    - Applying transformations to numpy arrays is much quicker than on Python lists
    - Numpy gives you access to **universal functions** (ufuncs) that are applied almost instantaneously
    - Numpy arrays are computationally much more efficient than Python lists, and this has to do with how arrays are stored in memory
      * Basically, Numpy stores data in continuous blocks of memory, whereas Python lists store pointers to data that can be located in various places in memory



* + - However, this comes with a tradeoff – within a given Numpy array, you can only store one type of data. Remember that Python lists can store multiple types of data
      * If you attempt to store both integers and floats into a Numpy array, Numpy will force them to one data type
      * In the case of integers and floating point variables, it will force everything to a floating point variable, which is known as **implicit upcasting**

