In [5]:
```
w1, w2, w3 = 0.3, 0.2, 0.5
```

In [6]:
```
kanto_temp = 73
kanto_rainfall = 67
kanto_humidity = 43
```

In [7]:
```
kanto_yield_apples = kanto_temp * w1 + kanto_rainfall * w2 + kanto_humidity * w3
kanto_yield_apples
```

Out[7]:
```
56.8
```

In [8]:
```
kanto = [73, 67, 43]
johto = [91, 88, 64]
hoenn = [87, 134, 58]
sinnoh = [102, 43, 37]
unova = [69, 96, 70]
```

In [9]:
```
weights = [w1, w2, w3]
```

In [10]:
```
def crop_yield(region, weights):
    result = 0
    for x, w in zip(region, weights):
        result += x * w
    return result
```

In [11]:
```
crop_yield(kanto, weights)
```

Out[11]:
```
56.8
```

In [12]:
```
crop_yield(johto,weights)
```

Out[12]:
```
76.9
```

In [13]:
```
crop_yield(hoenn, weights)
```

Out[13]:
```
81.9
```

In [14]:
```
crop_yield(sinnoh, weights)
```

Out[14]:
```
57.699999999999996
```

In [15]:
```
crop_yield(unova, weights)
```

Out[15]:
```
74.9
```

In [16]:

```
!pip -v
```

```

Usage:   
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  cache                       Inspect and manage pip's wheel cache.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  debug                       Show information useful for debugging.
  help                        Show help for commands.

General Options:
  -h, --help                  Show help.
  --isolated                  Run pip in an isolated mode, ignoring
                              environment variables and user configuration.
  -v, --verbose               Give more output. Option is additive, and can be
                              used up to 3 times.
  -V, --version               Show version and exit.
  -q, --quiet                 Give less output. Option is additive, and can be
                              used up to 3 times (corresponding to WARNING,
                              ERROR, and CRITICAL logging levels).
  --log <path>                Path to a verbose appending log.
  --no-input                  Disable prompting for input.
  --proxy <proxy>             Specify a proxy in the form
                              [user:passwd@]proxy.server:port.
  --retries <retries>         Maximum number of retries each connection should
                              attempt (default 5 times).
  --timeout <sec>             Set the socket timeout (default 15 seconds).
  --exists-action <action>    Default action when a path already exists:
                              (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname>   Mark this host or host:port pair as trusted,
                              even though it does not have valid or any HTTPS.
  --cert <path>               Path to alternate CA bundle.
  --client-cert <path>        Path to SSL client certificate, a single file
                              containing the private key and the certificate
                              in PEM format.
  --cache-dir <dir>           Store the cache data in <dir>.
  --no-cache-dir              Disable the cache.
  --disable-pip-version-check
                              Don't periodically check PyPI to determine
                              whether a new version of pip is available for
                              download. Implied with --no-index.
  --no-color                  Suppress colored output.
  --no-python-version-warning
                              Silence deprecation warnings for upcoming
                              unsupported Pythons.
  --use-feature <feature>     Enable new functionality, that may be backward
                              incompatible.
  --use-deprecated <feature>  Enable deprecated functionality, that will be
                              removed in the future.
```

In [17]:
```
!pyhton -m pip install --upgrade pip
```

```
'pyhton' is not recognized as an internal or external command,
operable program or batch file.
```

In [18]:
```
!pip install numpy --upgrade --quiet --user
```

```
  WARNING: The script f2py.exe is installed in 'C:\Users\JAY NARAYAN\AppData\Roaming\Py
thon\Python38\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, us
e --no-warn-script-location.
```

In [19]:
```
import numpy as np
```

In [20]:
```
kanto = np.array([73, 67, 43])
```

In [21]:
```
kanto
```

Out[21]:
```
array([73, 67, 43])
```

In [26]:
```
weights = np.array([w1, w2, w3])
```

In [27]:
```
weights
```

Out[27]:
```
array([0.3, 0.2, 0.5])
```

In [28]:
```
weights[0]
```

Out[28]:
```
0.3
```

In [29]:
```
kanto[2]
```

Out[29]:
```
43
```

In [30]:
```
#operating on numpy arrays
np.dot(kanto, weights)
```

Out[30]:
```
56.8
```

In [31]:
```
type(kanto)
```

Out[31]:
```
numpy.ndarray
```

In [32]:
```
type(weights)
```

Out[32]:
```
numpy.ndarray
```

In [ ]:
```

```