# Tutorial 1

1. virtualenv
   - Relevant links - documentation
   - Installation
     ```
     pip3 install virtualenv
     ```
   - To create a virtualenv
     ```
     virtualenv test_env
     ```
   - To start working on the new environment
     ```
     source test_env/bin/activate
     ```
   - To exit environment
     ```
     deactivate
     ```
   - Generate `requirements.txt` from a `virtualenv`
     ```
     pip3 freeze > requirements.txt
     ```
2. virtualenvwrapper
   - Relevant links - Medium@gitudaniel, documentation
   - Run the following on your terminal
     ```
     pip3 install virtualenvwrapper
     export WORKON_HOME=~/Envs
     mkdir -p $WORKON_HOME
     ```
   - Add this to your .bashrc
     ```
     source /usr/local/bin/virtualenvwrapper.sh
     ```
   - You may run into an error regarding python not being found. In case of that, add this to your .bashrc
     ```
     export VIRTUALENVWRAPPER_PYTHON=$(which python3)
     ```
   - To create a virtualenv
     ```
     mkvirtualenv test_env
     ```
   - To start working on the new environment
     ```
     workon test_env
     ```
   - To list packages on environment
     ```
     lssitepackages
     ```
   - To exit environment
     ```
     deactivate
     ```
   - All environments stored in `~/Envs`
   - To list all available environments,
     ```
     ls $WORKON_HOME
     ```
3. python lists
   - Relevant links - documentation
   - Examples
     ```
     list1 = ['physics', 'chemistry', 1997, 2000]
     list2 = [1, 2, 3, 4, 5 ]
     list3 = ["a", "b", "c", "d"]
     list2[1:5]:  [2, 3, 4, 5]
     list[2] = 2001
     del list[2]
     ```

```
len(list1)
list1[1:] => ['chemistry', 1997, 2000]
list1.append(5)
list1.sort([func])
list1.reverse()
```

4. python dictionaries
   - Relevant links - documentation
   - Examples

```
marks = {'maths': 45, 'english': 47, 'chemistry': 45}
marks['english'] => 47
marks => {'maths': 45, 'english': 47, 'chemistry': 45}
del marks['english'] => {'maths': 45, 'chemistry': 45}

Using the dict() constructor
marks = dict([('english', 45), ('maths', 44)])

Looping through a dictionary
for k,v in marks.items():
  print(k, v)
```

5. Exceptions (links - documentation
   - Exceptions on python3

```
ZeroDivisionError, NameError, TypeError, ValueError, KeyboardInterrupt

try:
  '2' + 2
except TypeError:
  print("Found Type Error")

try:
  1/0
except ZeroDivisionError:
  print("Found Zero Division Error")

try:
    xyz*5
except NameError:
    print("Found Type Error")

while True:
   try:
      x = int(input("Please enter a number: "))
      break
   except (ValueError, KeyboardInterrupt):
      print("Not a valid number. Please try again")
```