# Essential Concepts in Python

- lamda
- map, filter
- import module
- regular expression
- numpy + pandas
- json

```
##create a new function in python
def hello():
  return "hello world"
```

```
hello()
```

```
'hello world'
```

```
def add_two_nums(a,b):
  return a+b
```

```
add_two_nums(5,2)
```

```
7
```

```
## lamda is annonymas function that add function + input + expression
test = lambda x: print(x)
```

```
test("may")
```

```
may
```

```
add_two = lambda num: print(num+2)
```

```
add_two(8)
```

```
10
```

```
test2 = lambda a,b:a+b
```

```
test2(5,6)
```

```
11
```

```
test3 = lambda name, fav_food: f"{name} love {fav_food}."
```

```
test3("john","somtam")
```

```
'john love somtam.'
```

```
##map / filter avoid for loop
```

```
##map / filter avoid for loop
##iterable (adj)
scores = [82,78,81,66,50]
```

```
for score in scores :
  if score >= 80:
    print("passed")
  else:
    print("failed")
```

```
passed
failed
passed
failed
failed
```

```
def grading(score):
  if score >= 80 :
    return "P"
  else:
    return "Failed"
```

```
##map
list(map(grading, scores))
```

```
['P', 'Failed', 'P', 'Failed', 'Failed']
```

```
##add five to scores
new_scores = []
for score in scores:
  new_scores.append(score+5)
print(new_scores)
```

```
[87, 83, 86, 71, 55]
```

```
list(map(lambda x: x+5,scores))
```

```
[87, 83, 86, 71, 55]
```

```
##filter
scores
```

```
[82, 78, 81, 66, 50]
```

```
students_score_over_80 = []
for score in scores:
  if score>=80:
    students_score_over_80.append(score)
print(students_score_over_80)
```

```
[82, 81]
```

```
list(filter(lambda x:x>=80,scores))
```

```
[82, 81]
```

```
scores = [("may",80), ("jane",78), ("lisa",96)]
```

```
##filter score >= 80
list(filter(lambda x: x[1]>= 80, scores))
```

```
[('may', 80), ('lisa', 96)]
```

```
##module/library
##homework: class ATM
class ATM:
  pass
```

```
##standard python modules
import random
```

```
fruits = ["orange","banana","pineapple"]
```

```
random.choice(fruits)
```

```
'banana'
```

```
##fix random
for i in range(5):
#  random.seed(42)
  print(random.choice(fruits))
```

```
orange
orange
pineapple
banana
orange
```

```
#random digit
random.randint(10000,99999)
```

```
39256
```

```
##python refer tab for control scope / class of function
import random
class ATM:

  def __init__(self,name,balance):
    self.name = name
    self.balance = balance

  def __str__(self):
    return f"Account Name:{self.name}"

  def get_otp(self):
    otp = random.randint(10000,99999)
    return f"Your Requested OTP: {otp}"
```

```
my_account = ATM("may",500)
```

```
print(my_account)
```

```
Account Name:may
```

```
##ATM method
my_account.get_otp()
```

```
'Your Requested OTP: 23434'
```

```
## import regular expression
import re
```

```
text = "a duck costs $80 usd per piece, duck! 20"
```

```
#name module (re) + name function (search) +.
re.search("duck",text)
```

```
<re.Match object; span=(2, 6), match='duck'>
```

```
text[2:6]
```

```
'duck'
```

```
re.findall("duck",text)
```

```
['duck', 'duck']
```

```
re.findall("[0-9]{2}", text)
```

```
['80', '20']
```

```
#run terminal in collab
!pwd
```

```
/content
```

```
!rm hello.py
```

```
rm: cannot remove 'hello.py': No such file or directory
```

```
!ls
```

```
sample_data
```

```
##install new library/module
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from 
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from pyth
```

```
!pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (2.0.2)
```

```
#see all library
!pip list
```

```
torchdata                          0.11.0
torchsummary                       1.5.1
torchtune                          0.6.1
torchvision                        0.24.0+cpu
tornado                            6.5.1
tqdm                               4.67.1
traitlets                          5.7.1
traittypes                         0.2.3
transformers                       4.57.3
treescope                          0.1.10
tsfresh                            0.21.1
tweepy                             4.16.0
typeguard                          4.4.4
typer                              0.20.0
typer-slim                         0.20.0
types-pytz                         2025.2.0.20251108
types-setuptools                   80.9.0.20250822
typing_extensions                  4.15.0
typing-inspection                  0.4.2
tzdata                             2025.3
tzlocal                            5.3.1
uc-micro-py                        1.0.3
umap-learn                         0.5.9.post2
umf                                1.0.2
uri-template                       1.3.0
uritemplate                        4.2.0
urllib3                            2.5.0
uuid_utils                         0.12.0
uvicorn                            0.38.0
vega-datasets                      0.9.0
wadllib                            1.3.6
wandb                              0.23.1
wasabi                             1.1.3
watchdog                           6.0.0
wcwidth                            0.2.14
weasel                             0.4.3
webcolors                          25.10.0
webencodings                       0.5.1
websocket-client                   1.9.0
websockets                         15.0.1
Werkzeug                           3.1.4
wheel                              0.45.1
widgetsnbextension                 3.6.10
wordcloud                          1.9.4
wrapt                              2.0.1
wurlitzer                          3.1.1
xarray                             2025.12.0
xarray-einstats                    0.9.1
xgboost                            3.1.2
xlrd                               2.0.2
xxhash                             3.6.0
xyzservices                        2025.11.0
yarl                               1.22.0
ydf                                0.13.0
yellowbrick                        1.5
yfinance                           0.2.66
zipp                               3.23.0
zstandard                          0.25.0
```

```
##open file but server cannt connect to internet
##cannot install pandas
import csv
```

```
#context manager
with open("filename.csv","r") as file:
  pass
```

```
#import Jason
#working with Jason (Javascript opject notation)

profile = {
    "name":"may",
    "age": 39,
    "movies": ["The loard of the ring","Harry Potter"],
    "city":"Bangkok"


}
```

```
type(profile)
```

```
dict
```

```
##import json and save file
import json
##"w" for write json file
with open("profile.json","w") as file:
  json.dump(profile, file)
```

```
##read json file
with open("profile.json","r") as file:
  data = json.load(file) #load file to data

data
```

```
{'name': 'may',
 'age': 39,
 'movies': ['The loard of the ring', 'Harry Potter'],
 'city': 'Bangkok'}
```

```
data["movies"][1]
```

```
'Harry Potter'
```

```
##numpy
#statistical funcion in numpy (python)
##numerical python = matrix/computing
import random

nums =[]
for i in range(10) :
  nums.append(random.randint(100,999))
print(nums)
```

```
[792, 858, 658, 189, 704, 532, 132, 130, 195, 323]
```

```
sum(nums)
```

```
4513
```

```
#average in python
sum(nums)/len(nums)
```

```
451.3
```

```
min(nums)
```

```
130
```

```
max(nums)
```

```
858
```

```
import numpy as np
```

```
nums
```

```
[792, 858, 658, 189, 704, 532, 132, 130, 195, 323]
```

```
print(np.sum(nums), ##function sum in library numpy
      np.mean(nums),
      np.std(nums),
      np.median(nums),
      np.var(nums))
```

```
4513 451.3 273.99162395956563 427.5 75071.41
```

```
#change to numpy array
#array == vector in R
nums = np.array(nums)
```

```
type(nums) ##multiple dimension
```

```
numpy.ndarray
```

```
nums
```

```
array([792, 858, 658, 189, 704, 532, 132, 130, 195, 323])
```

```
#numpy method
nums.sum()
```

```
np.int64(4513)
```

```
print(nums.sum(),
      nums.std(),
      nums.mean(),
      nums.min(),
      nums.min(),
      nums.max()
      )
```

```
4513 273.99162395956563 451.3 130 130 858
```

```
##median no method but function instead meaning no [nums.median]
np.median(nums)
```

```
np.float64(427.5)
```

```
#import pandas+numpy
import pandas as pd
import numpy as np
```

```
df = pd.DataFrame({
    "name":["toy","top","tab"],
    "income": [50000,34000,48000]

})

df
```

|   | name | income |
|---|---|---|
| **0** | toy | 50000 |
| **1** | top | 34000 |
| **2** | tab | 48000 |

Next steps: New interactive sheet

```
df = pd.DataFrame({
    "name":["toy","top","tab"],
    "income": [50000,34000,48000]

})

df["income"].sum()  #numpy function in pandas
```

```
np.int64(132000)
```

```
##simple web scraping
```

## web scraping

### for statics web

```
!pip install gazpacho
```

```
Collecting gazpacho
  Downloading gazpacho-1.1.tar.gz (7.9 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: gazpacho
  Building wheel for gazpacho (pyproject.toml) ... done
  Created wheel for gazpacho: filename=gazpacho-1.1-py3-none-any.whl size=7521 sha256=122acfe
  Stored in directory: /root/.cache/pip/wheels/2b/49/33/b889bdad7e58b8a514eb3a47869eadb9ef67b
Successfully built gazpacho
Installing collected packages: gazpacho
Successfully installed gazpacho-1.1
```

```
##import library (modules)
import requests
from gazpacho import Soup
```

```
url = "https://raw.githubusercontent.com/toyeiei/python-test-bc12/refs/heads/main/inde
```

```
##get request
requests.get(url) ## if 200 means status ok
```

```
<Response [200]>
```

```
response = requests.get(url)
```

```
response.status_code ##use . to see inside
```

```
200
```

```
response.text
```

```
'<!DOCTYPE html>\n<html lang="en">\n<head>\n    <meta charset="UTF-8">\n    <meta name="view
port" content="width=device-width, initial-scale=1.0">\n    <title>Simple Structure with Cla
sses</title>\n</head>\n<body>\n\n    <h1>Mastering Development Fundamentals</h1>\n\n    <h2
class="section-tech">Core Technologies</h2>\n    <ul>\n        <li>Python</li>\n        <li>
SQL</li>\n        <li>HTML/CSS</li>\n    </ul>\n\n    <h2 class="section-workflow">Data Scie
nce Workflow</h2>\n    <ol>\n        <li>Data Collection</li>\n        <li>Cleaning and Prep
rocessing</li>\n        <li>Model Training</li>\n    </ol>\n\n    <h2 class="section-mindse
t">Key Mindsets</h2>\n    <ul>\n        <li>Stoic discipline</li>\n        <li>Strategic foc
us</li>\n        <li>Continuous learning</li>\n    </ul>\n\n</body>\n</html>\n'
```

```
#build text from html
web = Soup(response.text)
```

```
web
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Structure with Classes</title>
</head>
<body>

    <h1>Mastering Development Fundamentals</h1>

    <h2 class="section-tech">Core Technologies</h2>
    <ul>
        <li>Python</li>
        <li>SQL</li>
        <li>HTML/CSS</li>
    </ul>

    <h2 class="section-workflow">Data Science Workflow</h2>
    <ol>
        <li>Data Collection</li>
        <li>Cleaning and Preprocessing</li>
        <li>Model Training</li>
    </ol>

    <h2 class="section-mindset">Key Mindsets</h2>
    <ul>
        <li>Stoic discipline</li>
        <li>Strategic focus</li>
        <li>Continuous learning</li>
    </ul>
```

```
</body>
</html>
```

```
web.find("h1")
```

```
<h1>Mastering Development Fundamentals</h1>
```

```
web.find("h2")
```

```
[<h2 class="section-tech">Core Technologies</h2>,
 <h2 class="section-workflow">Data Science Workflow</h2>,
 <h2 class="section-mindset">Key Mindsets</h2>]
```

```
web.find("li") ##soup method to find eliment list
```

```
[<li>Python</li>,
 <li>SQL</li>,
 <li>HTML/CSS</li>,
 <li>Data Collection</li>,
 <li>Cleaning and Preprocessing</li>,
 <li>Model Training</li>,
 <li>Stoic discipline</li>,
 <li>Strategic focus</li>,
 <li>Continuous learning</li>]
```

```
#delete html message but keep only text
for h2 in web.find("h2"):
  print(h2.strip())
```

```
Core Technologies
Data Science Workflow
Key Mindsets
```

```
for li in web.find("li"):
  print(li.strip())
```

```
Python
SQL
HTML/CSS
Data Collection
Cleaning and Preprocessing
Model Training
Stoic discipline
Strategic focus
Continuous learning
```