Operators in Python

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Membership Operators

```
# Arithmetic Operators
print(5+6)

print(5-6)

print(5*6)

print(5/2)

print(5//2)

print(5%2)

print(5**2)
```

```
11
-1
30
2.5
2
1
25
```

```
# Relational Operators
print(4>5)

print(4<5)

print(4>=4)

print(4<=4)

print(4==4)

print(4!=4)
```

```
False
True
True
True
```

```
True
False

# Logical Operators
print(1 and 0)

print(1 or 0)

print(not 1)

0
1
False

# Bitwise Operators

# bitwise and
print(2 & 3)

# bitwise or
print(2 | 3)

# bitwise xor
print(2 ^ 3)

print(~3)

print(4 >> 2)

print(5 << 2)

2
3
1
-4
1
20

# Assignment Operators

# =
# a = 2

a = 2

# a = a % 2
a %= 2

# a++ ++a

print(a)
```

```
4
```

```
# Membership Operators

# in/not in

print('D' not in 'Delhi')

print(1 in [2,3,4,5,6])
```

```
False
False
```

```
# Program - Find the sum of a 3 digit number entered by the user

number = int(input('Enter a 3 digit number'))

# 345%10 -> 5
a = number%10

number = number//10

# 34%10 -> 4
b = number % 10

number = number//10
# 3 % 10 -> 3
c = number % 10

print(a + b + c)
```

```
Enter a 3 digit number345
12
```

If-else in Python

```
# login program and indentation
# email -> nitish.campusx@gmail.com
# password -> 1234

email = input('enter email')
password = input('enter password')

if email == 'nitish.campusx@gmail.com' and password == '1234':
  print('Welcome')
elif email == 'nitish.campusx@gmail.com' and password != '1234':
  # tell the user
  print('Incorrect password')
  password = input('enter password again')
  if password == '1234':
    print('Welcome,finally!')
```

```
  else:
    print('beta tumse na ho paayega!')
else:
  print('Not correct')
```

```
enter emailsrhreh
enter passworderhetjh
Not correct
```

```
# if-else examples
# 1. Find the min of 3 given numbers
# 2. Menu Driven Program

# min of 3 number

a = int(input('first num'))
b = int(input('second num'))
c = int(input('third num'))

if a<b and a<c:
  print('smallest is',a)
elif b<c:
  print('smallest is',b)
else:
  print('smallest is',c)
```

```
first num4
second num1
third num10
smallest is 1
```

```
# menu driven calculator
menu = input("""
Hi! how can I help you.
1. Enter 1 for pin change
2. Enter 2 for balance check
3. Enter 3 for withdrawl
4. Enter 4 for exit
""")

if menu == '1':
  print('pin change')
elif menu == '2':
  print('balance')
else:
  print('exit')
```

```
Hi! how can I help you.
1. Enter 1 for pin change
2. Enter 2 for balance check
```

```
3. Enter 3 for withdrawl
4. Enter 4 for exit
2
balance
```

Modules in Python

- math
- keywords
- random
- datetime

```
# math
import math

math.sqrt(196)
```

```
14.0
```

```
# keyword
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await',
'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try',
'while', 'with', 'yield']
```

```
# random
import random
print(random.randint(1,100))
```

```
88
```

```
# datetime
import datetime
print(datetime.datetime.now())
```

```
2022-11-08 15:50:21.228643
```

```
help('modules')
```

```
Please wait a moment while I gather a list of all available modules...


/usr/local/lib/python3.7/dist-packages/caffe2/proto/__init__.py:17:
UserWarning: Caffe2 support is not enabled in this PyTorch build.
Please enable Caffe2 by building PyTorch from source with
`BUILD_CAFFE2=1` flag.
/usr/local/lib/python3.7/dist-packages/caffe2/proto/__init__.py:17:
```

```
UserWarning: Caffe2 support is not enabled in this PyTorch build.
Please enable Caffe2 by building PyTorch from source with
`BUILD_CAFFE2=1` flag.
/usr/local/lib/python3.7/dist-packages/caffe2/python/__init__.py:9:
UserWarning: Caffe2 support is not enabled in this PyTorch build.
Please enable Caffe2 by building PyTorch from source with
`BUILD_CAFFE2=1` flag.

Cython              collections         kaggle
requests_oauthlib
IPython             colorcet            kanren              resampy
OpenGL              colorlover          kapre               resource
PIL                 colorsys            keras
rlcompleter
ScreenResolution    community           keras_preprocessing rmagic
__future__          compileall          keyword             rpy2
_abc                concurrent          kiwisolver          rsa
_ast                confection          korean_lunar_calendar runpy
_asyncio            configparser        langcodes           samples
_bisect             cons                lib2to3             sched
_blake2             contextlib          libfuturize         scipy
_bootlocale         contextlib2         libpasteurize       scs
_bz2                contextvars         librosa             seaborn
_cffi_backend       convertdate         lightgbm            secrets
_codecs             copy                linecache           select
_codecs_cn          copyreg             llvmlite            selectors
_codecs_hk          crashtest           lmdb                send2trash
_codecs_iso2022     crcmod              locale              setuptools
_codecs_jp          crypt               locket
setuptools_git
_codecs_kr          csimdjson           logging             shapely
_codecs_tw          csv                 lsb_release         shelve
_collections        ctypes              lunarcalendar       shlex
_collections_abc    cufflinks           lxml                shutil
_compat_pickle      curses              lzma                signal
_compression        cv2                 macpath             simdjson
_contextvars        cvxopt              mailbox             site
_crypt              cvxpy               mailcap
sitecustomize
_csv                cycler              markdown            six
_ctypes             cymem               markupsafe          skimage
_ctypes_test        cython              marshal             sklearn
_curses             cythonmagic         marshmallow
sklearn_pandas
_curses_panel       daft                math                slugify
_cvxcore            dask                matplotlib          smart_open
_datetime           dataclasses         matplotlib_venn     smtpd
_dbm                datascience         mimetypes           smtplib
_decimal            datetime            missingno           sndhdr
_distutils_hack     dateutil            mistune
```

```
snowballstemmer
_dlib_pybind11      dbm                 mizani              socket
_dummy_thread       dbus                mlxtend
socketserver
_ecos               debugpy             mmap                socks
_elementtree        decimal             modulefinder
sockshandler
_functools          decorator           more_itertools
softwareproperties
_hashlib            defusedxml          moviepy
sortedcontainers
_heapq              descartes           mpmath              soundfile
_imp                difflib             msgpack             spacy
_io                 dill                multidict
spacy_legacy
_json               dis                 multipledispatch
spacy_loggers
_locale             distributed         multiprocessing     sphinx
_lsprof             distutils           multitasking        spwd
_lzma               dlib                murmurhash          sql
_markupbase         dns                 music21             sqlalchemy
_md5                docs                natsort             sqlite3
_multibytecodec     doctest             nbconvert           sqlparse
_multiprocessing    docutils            nbformat
sre_compile
_opcode             dopamine            netCDF4
sre_constants
_operator           dot_parser          netrc               sre_parse
_osx_support        dummy_threading     networkx            srsly
_pickle             easydict            nibabel             ssl
_plotly_future_     ecos                nis                 stan
_plotly_utils       editdistance        nisext              stat
_posixsubprocess    ee                  nltk                statistics
_py_abc             email               nntplib
statsmodels
_pydecimal          en_core_web_sm      notebook            storemagic
_pyio               encodings           ntpath              string
_pyrsistent_version entrypoints         nturl2path          stringprep
_pytest             enum                numba               struct
_queue              ephem               numbergen           subprocess
_random             erfa                numbers             sunau
_remote_module_non_scriptable errno             numexpr
symbol
_rinterface_cffi_abi et_xmlfile          numpy               sympy
_rinterface_cffi_api etils               oauth2client
sympyprinting
_scs_direct         etuples             oauthlib            symtable
_scs_indirect       fa2                 ogr                 sys
_sha1               fastai              okgrade             sysconfig
```

```
_sha256                fastcore               opcode                 syslog
_sha3                  fastdownload           openpyxl               tables
_sha512                fastdtw                operator               tabnanny
_signal                fastjsonschema         opt_einsum             tabulate
_sitebuiltins          fastprogress           optparse               tarfile
_socket                fastrlock              os                     tblib
_soundfile             faulthandler           osgeo                  telnetlib
_sqlite3               fcntl                  osqp                   tempfile
_sre                   feather                osqppurepy             tenacity
_ssl                   filecmp                osr
tensorboard
_stat                  fileinput              ossaudiodev
tensorboard_data_server
_string                filelock               packaging
tensorboard_plugin_wit
_strptime              firebase_admin         palettable             tensorflow
_struct                fix_yahoo_finance      pandas
tensorflow_datasets
_symtable              flask                  pandas_datareader
tensorflow_estimator
_sysconfigdata_m_linux_x86_64-linux-gnu flatbuffers            pandas_gbq
tensorflow_gcs_config
_sysconfigdata_m_x86_64-linux-gnu fnmatch                pandas_profiling
tensorflow_hub
_testbuffer            folium                 pandocfilters
tensorflow_io_gcs_filesystem
_testcapi              formatter              panel
tensorflow_metadata
_testimportmultiple    fractions              param
tensorflow_probability
_testmultiphase        frozenlist             parser                 termcolor
_thread                fsspec                 parso                  terminado
_threading_local       ftplib                 partd                  termios
_tkinter               functools              past                   test
_tracemalloc           future                 pasta                  testpath
_warnings              gast                   pastel                 tests
_weakref               gc                     pathlib
text_unidecode
_weakrefset            gdal                   pathy                  textblob
_xxtestfuzz            gdalconst              patsy                  textwrap
_yaml                  gdalnumeric            pdb                    thinc
abc                    gdown                  pep517                 this
absl                   genericpath            pexpect                threading
aeppl                  gensim                 pickle
threadpoolctl
aesara                 geographiclib          pickleshare            tifffile
aifc                   geopy                  pickletools            time
aiohttp                getopt                 pip                    timeit
aiosignal              getpass                pipes                  tkinter
```

```
alabaster            gettext              piptools             tlz
albumentations       gi                   pkg_resources        token
altair               gin                  pkgutil              tokenize
antigravity          glob                 platform             toml
apiclient            glob2                plistlib             tomli
appdirs              gnm                  plotly               toolz
apt                  google_auth_httplib2 plotlywidget          torch
apt_inst             google_auth_oauthlib plotnine
torchaudio
apt_pkg              google_drive_downloader pluggy
torchgen
aptsources           googleapiclient      pooch
torchsummary
argparse             googlesearch         poplib               torchtext
array                graphviz             portpicker
torchvision
arviz                greenlet             posix                tornado
ast                  gridfs               posixpath            tqdm
astor                grp                  pprint               trace
astropy              grpc                 prefetch_generator   traceback
astunparse           gspread              preshed
tracemalloc
async_timeout        gspread_dataframe    prettytable          traitlets
asynchat             gym                  profile              tree
asyncio              gym_notices          progressbar          tty
asyncore             gzip                 promise              turtle
asynctest            h5py                 prompt_toolkit       tweepy
atari_py             hashlib              prophet              typeguard
atexit               heapdict             pstats               typer
atomicwrites         heapq                psutil               types
attr                 hijri_converter      psycopg2             typing
attrs                hmac                 pty
typing_extensions
audioop              holidays             ptyprocess           tzlocal
audioread            holoviews            pvectorc
unicodedata
autograd             html                 pwd
unification
autoreload           html5lib             py                   unittest
babel                http                 py_compile
uritemplate
backcall             httpimport           pyarrow              urllib
base64               httplib2             pyasn1               urllib3
bdb                  httplib2shim         pyasn1_modules       uu
bin                  httpstan             pyclbr               uuid
binascii             humanize             pycocotools
vega_datasets
binhex               hyperopt             pycparser            venv
bisect               idna                 pyct                 vis
```

```
bleach              imageio             pydantic            warnings
blis                imagesize           pydata_google_auth  wasabi
bokeh               imaplib             pydoc               wave
boost               imblearn            pydoc_data          wcwidth
branca              imgaug              pydot               weakref
bs4                 imghdr              pydot_ng            webargs
bson                imp                 pydotplus           webbrowser
builtins            importlib           pydrive
webencodings
bz2                 importlib_metadata  pyemd               werkzeug
cProfile            importlib_resources pyexpat             wheel
cachecontrol        imutils             pygments
widgetsnbextension
cached_property     inflect             pygtkcompat         wordcloud
cachetools          inspect             pylab               wrapt
caffe2              intervaltree        pylev               wsgiref
calendar            io                  pymc                xarray
catalogue           ipaddress           pymeeus
xarray_einstats
certifi             ipykernel           pymongo             xdrlib
cffi                ipykernel_launcher  pymystem3           xgboost
cftime              ipython_genutils    pyparsing           xkit
cgi                 ipywidgets          pyrsistent          xlrd
cgitb               isympy              pysndfile           xlwt
chardet             itertools           pytest              xml
charset_normalizer  itsdangerous        python_utils        xmlrpc
chunk               jax                 pytz                xxlimited
clang               jaxlib              pyviz_comms         xxsubtype
click               jieba               pywt                yaml
client              jinja2              pyximport           yarl
clikit              joblib              qdldl
yellowbrick
cloudpickle         jpeg4py             qudida              zict
cmake               json                queue               zipapp
cmath               jsonschema          quopri              zipfile
cmd                 jupyter             random              zipimport
cmdstanpy           jupyter_client      re                  zipp
code                jupyter_console     readline            zlib
codecs              jupyter_core        regex               zmq
codeop              jupyterlab_plotly   reprlib
colab               jupyterlab_widgets  requests

Enter any module name to get more help.  Or, type "modules spam" to
search
for modules whose name or summary contain the string "spam".
```

Loops in Python

- Need for loops

- While Loop
- For Loop

```
# While loop example -> program to print the table
# Program -> Sum of all digits of a given number
# Program -> keep accepting numbers from users till he/she enters a 0 and then find the avg

number = int(input('enter the number'))

i = 1

while i<11:
  print(number,'*',i,'=',number * i)
  i += 1
```

```
enter the number12
12 * 1 = 12
12 * 2 = 24
12 * 3 = 36
12 * 4 = 48
12 * 5 = 60
12 * 6 = 72
12 * 7 = 84
12 * 8 = 96
12 * 9 = 108
12 * 10 = 120
```

```
# while loop with else

x = 1

while x < 3:
  print(x)
  x += 1

else:
  print('limit crossed')
```

```
1
2
limit crossed
```

```
# Guessing game

# generate a random integer between 1 and 100
import random
jackpot = random.randint(1,100)

guess = int(input('guess karo'))
counter = 1
while guess != jackpot:
```

```
  if guess < jackpot:
    print('galat!guess higher')
  else:
    print('galat!guess lower')

  guess = int(input('guess karo'))
  counter += 1

else:
  print('correct guess')
  print('attempts',counter)



guess karo7
galat!guess higher
guess karo50
galat!guess lower
guess karo30
galat!guess higher
guess karo40
galat!guess lower
guess karo35
galat!guess lower
guess karo32
galat!guess higher
guess karo33
correct guess
attempts 7

# For loop demo

for i in {1,2,3,4,5}:
  print(i)

1
2
3
4
5

# For loop examples
```

Program - The current population of a town is 10000. The population of the town is increasing at the rate of 10% per year. You have to write a program to find out the population at the end of each of the last 10 years.

```
# Code Used in the session
curr_pop = 10000

for i in range(10,0,-1):
  print(i,curr_pop)
  curr_pop = curr_pop - 0.1*curr_pop
```

```
10 10000
9 9090.90909090909
8 8264.462809917353
7 7513.148009015775
6 6830.134553650703
5 6209.213230591548
4 5644.739300537771
3 5131.5811823070635
2 4665.07380209733
1 4240.976183724845
```

```
# Correct Answer of above question:
curr_pop = 10000

for i in range(10,0,-1):
  print(i,curr_pop)
  curr_pop /= 1.1
```

Explanation :

To calculate the population for each year with a 10% increase, you can use a simpler equation based on the previous year's population. Let's assume the population of the previous year is represented by variable x.

The equation can be written as:

Current Year Population = x * 1.1

In this equation, the current year's population is equal to the previous year's population multiplied by 1.1, representing a 10% increase.

To find the population of the previous year (x), we can rearrange the equation as follows:

x = Current Year Population / 1.1

Using this simplified equation, if you have the current year's population (e.g., 10,000), you can divide it by 1.1 to calculate the population of the previous year.

This equation allows you to calculate the population for each year, assuming you know the population of the current year and want to find the population of the previous year.

Sequence sum

1/1! + 2/2! + 3/3! + ...

```
# code here

# For loop vs While loops (When to use what?)
```

Nested Loops

```
# Examples

# Program - Unique combination of 1,2,3,4
# Program - Pattern 1 and 2
```

Pattern 1

*** **** ***

```

```

Pattern 2

1 121 12321 1234321

```

```

Loop Control Statement

- Break
- Continue
- Pass

```
# Break demo

# Break example (Linear Search) -> Prime number in a given range

# Continue demo

# Continue Example (Ecommerce)

# Pass demo
```