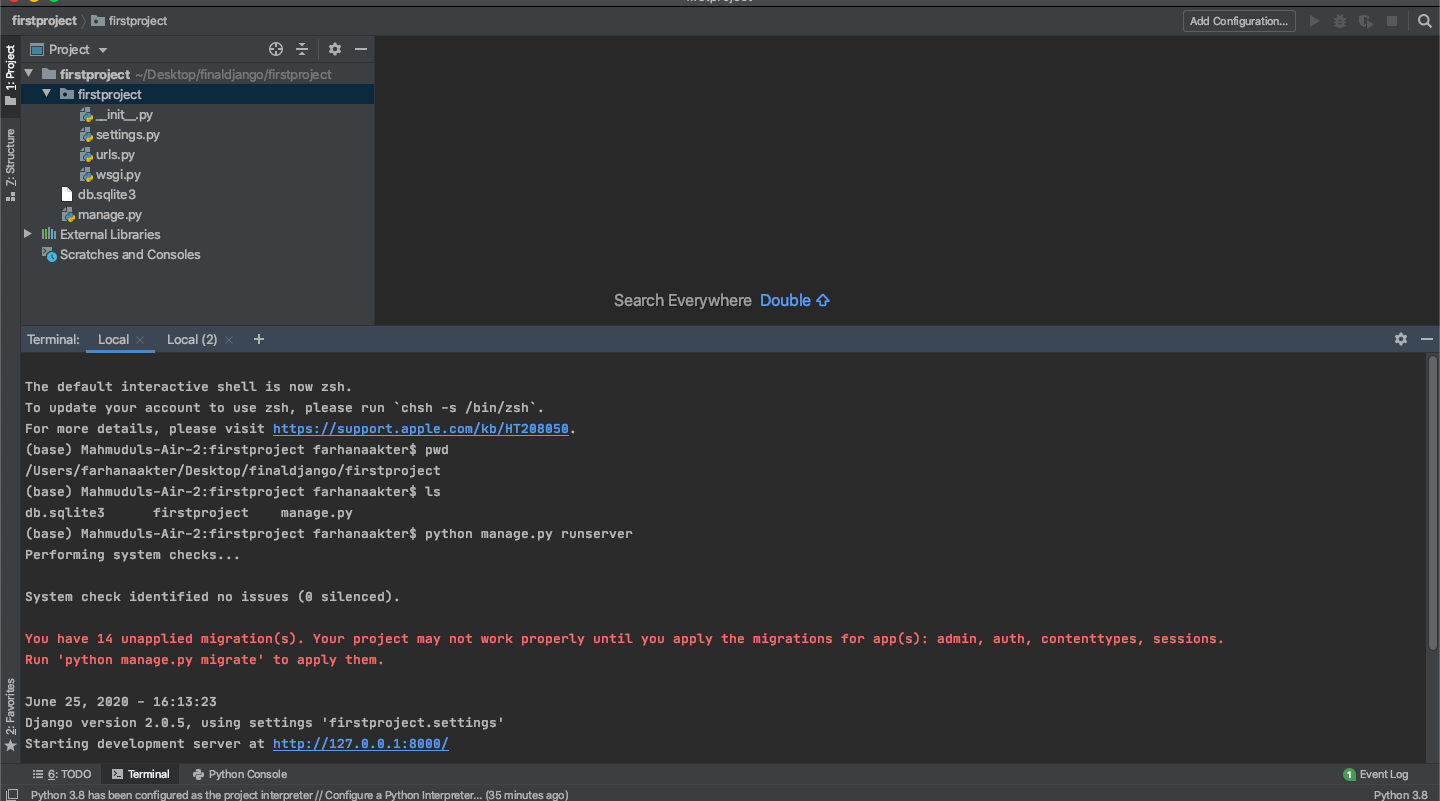
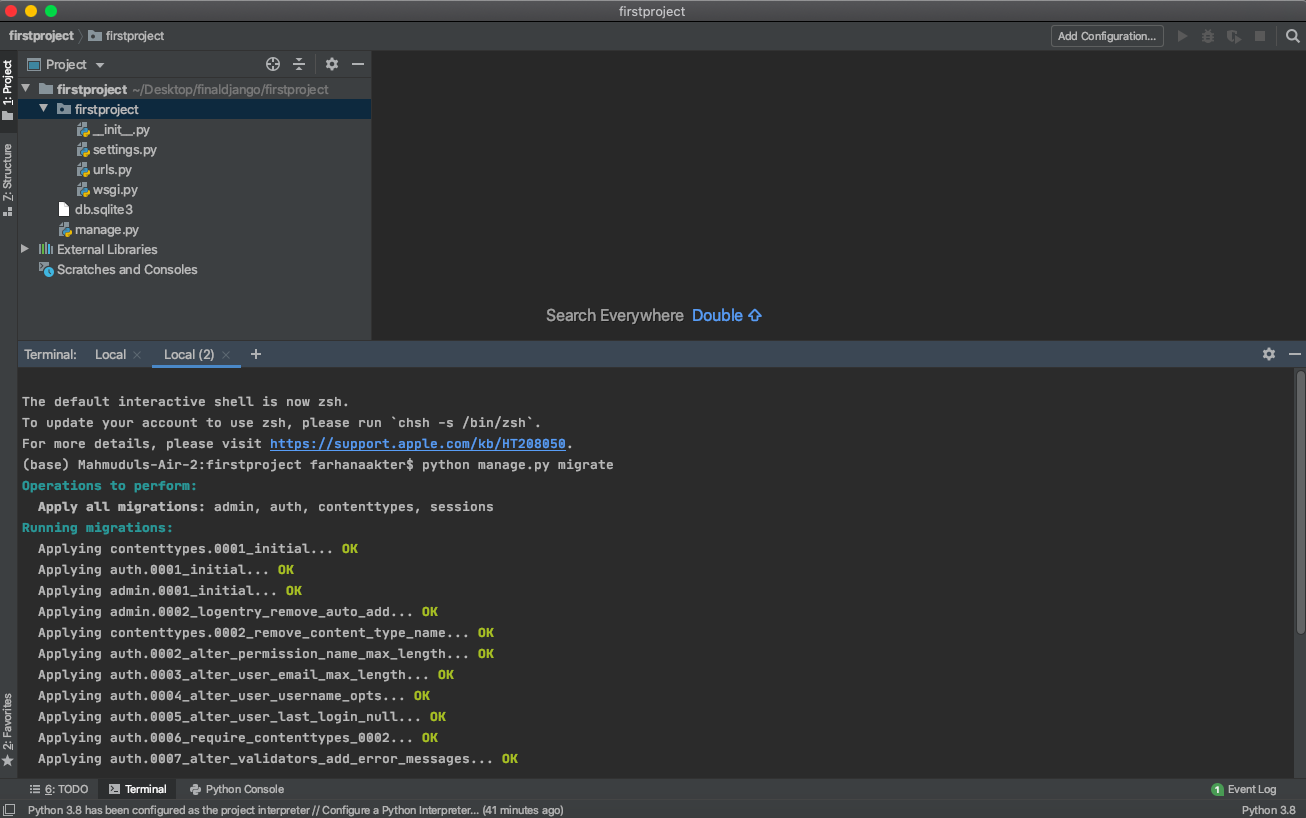
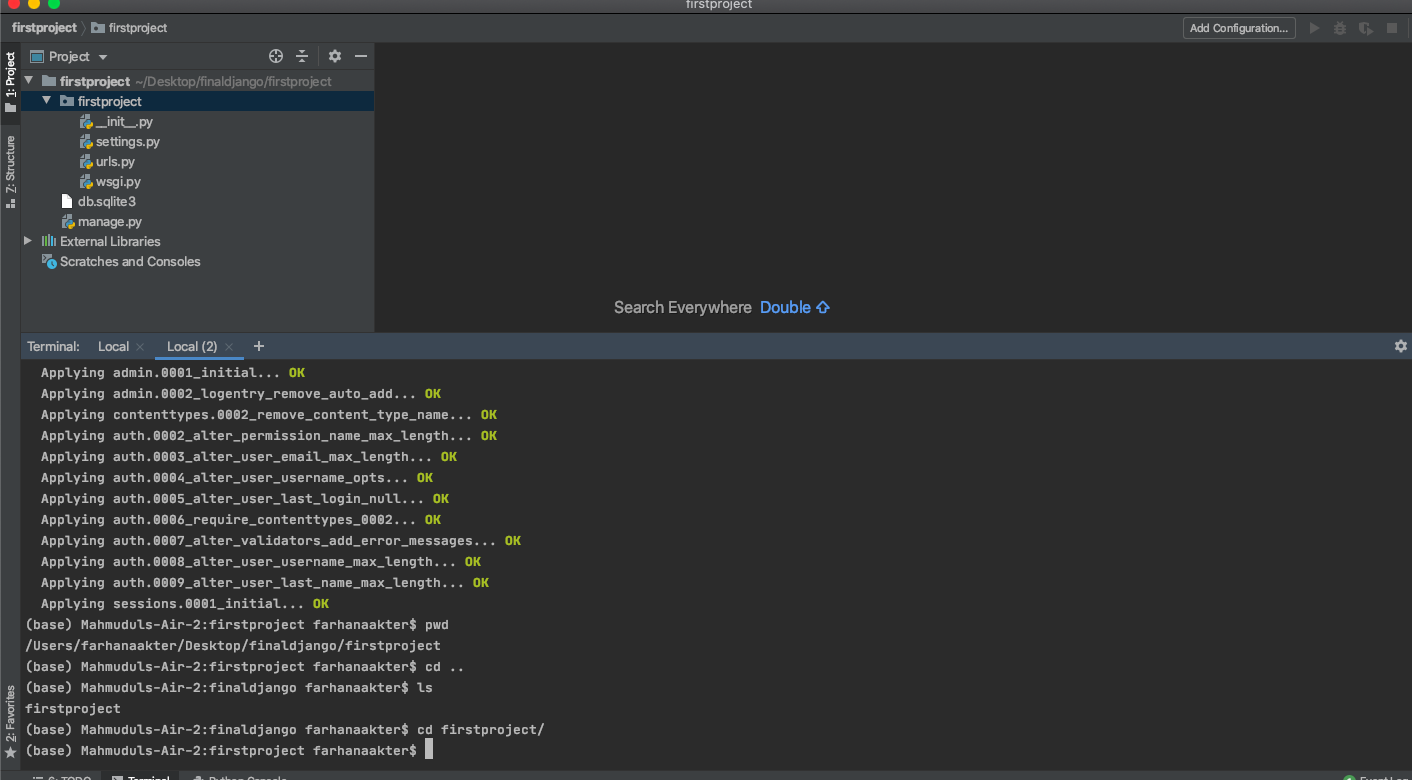
**Django Practice**

****

****

****

**The default interactive shell is now zsh.**

**To update your account to use zsh, please run `chsh -s /bin/zsh`.**

**For more details, please visit https://support.apple.com/kb/HT208050.**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py migrate**

**Operations to perform:**

**Apply all migrations: admin, auth, contenttypes, sessions**

**Running migrations:**

**Applying contenttypes.0001\_initial... OK**

**Applying auth.0001\_initial... OK**

**Applying admin.0001\_initial... OK**

**Applying admin.0002\_logentry\_remove\_auto\_add... OK**

**Applying contenttypes.0002\_remove\_content\_type\_name... OK**

**Applying auth.0002\_alter\_permission\_name\_max\_length... OK**

**Applying auth.0003\_alter\_user\_email\_max\_length... OK**

**Applying auth.0004\_alter\_user\_username\_opts... OK**

**Applying auth.0005\_alter\_user\_last\_login\_null... OK**

**Applying auth.0006\_require\_contenttypes\_0002... OK**

**Applying auth.0007\_alter\_validators\_add\_error\_messages... OK**

**Applying auth.0008\_alter\_user\_username\_max\_length... OK**

**Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK**

**Applying sessions.0001\_initial... OK**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ pwd**

**/Users/farhanaakter/Desktop/finaldjango/firstproject**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd ..**

**(base) Mahmuduls-Air-2:finaldjango farhanaakter$ ls**

**firstproject**

**(base) Mahmuduls-Air-2:finaldjango farhanaakter$ cd firstproject/**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ pip freeze**

**alabaster==0.7.12**

**anaconda-client==1.7.2**

**anaconda-navigator==1.9.12**

**anaconda-project==0.8.3**

**appdirs==1.4.4**

**applaunchservices==0.2.1**

**appnope==0.1.0**

**appscript==1.0.1**

**argh==0.26.2**

**asgiref==3.2.8**

**asn1crypto==1.3.0**

**astroid==2.3.3**

**astropy==4.0**

**async-generator==1.10**

**atomicwrites==1.3.0**

**attrs==19.3.0**

**autopep8==1.4.4**

**Babel==2.8.0**

**backcall==0.1.0**

**backports.functools-lru-cache==1.6.1**

**backports.shutil-get-terminal-size==1.0.0**

**backports.tempfile==1.0**

**backports.weakref==1.0.post1**

**beautifulsoup4==4.8.2**

**bitarray==1.2.1**

**bkcharts==0.2**

**bleach==3.1.0**

**bokeh==1.4.0**

**boto==2.49.0**

**Bottleneck==1.3.2**

**certifi==2019.11.28**

**cffi==1.14.0**

**chardet==3.0.4**

**Click==7.0**

**cloudpickle==1.3.0**

**clyent==1.2.2**

**colorama==0.4.3**

**conda==4.8.3**

**conda-build==3.18.11**

**conda-package-handling==1.6.0**

**conda-verify==3.4.2**

**contextlib2==0.6.0.post1**

**cryptography==2.8**

**cycler==0.10.0**

**Cython==0.29.15**

**cytoolz==0.10.1**

**dask==2.11.0**

**decorator==4.4.1**

**defusedxml==0.6.0**

**diff-match-patch==20181111**

**distlib==0.3.0**

**distributed==2.11.0**

**Django==2.0.5**

**djangorestframework==3.11.0**

**docutils==0.16**

**entrypoints==0.3**

**et-xmlfile==1.0.1**

**fastcache==1.1.0**

**filelock==3.0.12**

**flake8==3.7.9**

**Flask==1.1.1**

**fsspec==0.6.2**

**future==0.18.2**

**gevent==1.4.0**

**glob2==0.7**

**gmpy2==2.0.8**

**greenlet==0.4.15**

**h5py==2.10.0**

**HeapDict==1.0.1**

**html5lib==1.0.1**

**hypothesis==5.5.4**

**idna==2.8**

**imageio==2.6.1**

**imagesize==1.2.0**

**importlib-metadata==1.5.0**

**intervaltree==3.0.2**

**ipykernel==5.1.4**

**ipython==7.12.0**

**ipython-genutils==0.2.0**

**ipywidgets==7.5.1**

**isort==4.3.21**

**itsdangerous==1.1.0**

**jdcal==1.4.1**

**jedi==0.14.1**

**Jinja2==2.11.1**

**joblib==0.14.1**

**json5==0.9.1**

**jsonschema==3.2.0**

**jupyter==1.0.0**

**jupyter-client==5.3.4**

**jupyter-console==6.1.0**

**jupyter-core==4.6.1**

**jupyter-server==0.1.1**

**jupyterlab==1.2.6**

**jupyterlab-pygments==0.1.1**

**jupyterlab-server==1.0.6**

**keyring==21.1.0**

**kiwisolver==1.1.0**

**lazy-object-proxy==1.4.3**

**libarchive-c==2.8**

**lief==0.9.0**

**llvmlite==0.31.0**

**locket==0.2.0**

**lxml==4.5.0**

**MarkupSafe==1.1.1**

**matplotlib==3.1.3**

**mccabe==0.6.1**

**mistune==0.8.4**

**mkl-fft==1.0.15**

**mkl-random==1.1.0**

**mkl-service==2.3.0**

**mock==4.0.1**

**more-itertools==8.2.0**

**mpmath==1.1.0**

**msgpack==0.6.1**

**multipledispatch==0.6.0**

**navigator-updater==0.2.1**

**nbconvert==5.6.1**

**nbformat==5.0.4**

**networkx==2.4**

**nltk==3.4.5**

**nose==1.3.7**

**notebook==6.0.3**

**numba==0.48.0**

**numexpr==2.7.1**

**numpy==1.18.1**

**numpydoc==0.9.2**

**olefile==0.46**

**openpyxl==3.0.3**

**packaging==20.1**

**pandas==1.0.1**

**pandocfilters==1.4.2**

**parso==0.5.2**

**partd==1.1.0**

**path==13.1.0**

**pathlib2==2.3.5**

**pathtools==0.1.2**

**patsy==0.5.1**

**pep8==1.7.1**

**pexpect==4.8.0**

**pickleshare==0.7.5**

**Pillow==7.0.0**

**pkginfo==1.5.0.1**

**pluggy==0.13.1**

**ply==3.11**

**prometheus-client==0.7.1**

**prompt-toolkit==3.0.3**

**psutil==5.6.7**

**psycopg2-binary==2.8.5**

**ptyprocess==0.6.0**

**py==1.8.1**

**pycodestyle==2.5.0**

**pycosat==0.6.3**

**pycparser==2.19**

**pycrypto==2.6.1**

**pycurl==7.43.0.5**

**pydocstyle==4.0.1**

**pyflakes==2.1.1**

**Pygments==2.5.2**

**pylint==2.4.4**

**pyodbc===4.0.0-unsupported**

**pyOpenSSL==19.1.0**

**pyparsing==2.4.6**

**pyrsistent==0.15.7**

**PySocks==1.7.1**

**pytest==5.3.5**

**pytest-arraydiff==0.3**

**pytest-astropy==0.8.0**

**pytest-astropy-header==0.1.2**

**pytest-doctestplus==0.5.0**

**pytest-openfiles==0.4.0**

**pytest-remotedata==0.3.2**

**python-dateutil==2.8.1**

**python-jsonrpc-server==0.3.4**

**python-language-server==0.31.7**

**pytz==2019.3**

**PyWavelets==1.1.1**

**PyYAML==5.3**

**pyzmq==18.1.1**

**QDarkStyle==2.8**

**QtAwesome==0.6.1**

**qtconsole==4.6.0**

**QtPy==1.9.0**

**requests==2.22.0**

**rope==0.16.0**

**Rtree==0.9.3**

**ruamel-yaml==0.15.87**

**scikit-image==0.16.2**

**scikit-learn==0.22.1**

**scipy==1.4.1**

**seaborn==0.10.0**

**Send2Trash==1.5.0**

**simplegeneric==0.8.1**

**singledispatch==3.4.0.3**

**six==1.14.0**

**snowballstemmer==2.0.0**

**sortedcollections==1.1.2**

**sortedcontainers==2.1.0**

**soupsieve==1.9.5**

**Sphinx==2.4.0**

**sphinxcontrib-applehelp==1.0.1**

**sphinxcontrib-devhelp==1.0.1**

**sphinxcontrib-htmlhelp==1.0.2**

**sphinxcontrib-jsmath==1.0.1**

**sphinxcontrib-qthelp==1.0.2**

**sphinxcontrib-serializinghtml==1.1.3**

**sphinxcontrib-websupport==1.2.0**

**spyder==4.0.1**

**spyder-kernels==1.8.1**

**SQLAlchemy==1.3.13**

**sqlparse==0.3.1**

**statsmodels==0.11.0**

**sympy==1.5.1**

**tables==3.6.1**

**tblib==1.6.0**

**terminado==0.8.3**

**testpath==0.4.4**

**toolz==0.10.0**

**tornado==6.0.3**

**tqdm==4.42.1**

**traitlets==4.3.3**

**typed-ast==1.4.1**

**ujson==1.35**

**unicodecsv==0.14.1**

**urllib3==1.25.8**

**virtualenv==20.0.21**

**voila==0.1.21**

**watchdog==0.10.2**

**wcwidth==0.1.8**

**webencodings==0.5.1**

**Werkzeug==1.0.0**

**widgetsnbextension==3.5.1**

**wrapt==1.11.2**

**wurlitzer==2.0.0**

**xlrd==1.2.0**

**XlsxWriter==1.2.7**

**xlwings==0.17.1**

**xlwt==1.3.0**

**xmltodict==0.12.0**

**yapf==0.28.0**

**zict==1.0.0**

**zipp==2.2.0**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd firstproject/**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**\_\_init\_\_.py \_\_pycache\_\_ settings.py urls.py wsgi.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd..**

**bash: cd..: command not found**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd ..**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py**

**Type 'manage.py help <subcommand>' for help on a specific subcommand.**

**Available subcommands:**

**[auth]**

**changepassword**

**createsuperuser**

**[contenttypes]**

**remove\_stale\_contenttypes**

**[django]**

**check**

**compilemessages**

**createcachetable**

**dbshell**

**diffsettings**

**dumpdata**

**flush**

**inspectdb**

**loaddata**

**makemessages**

**makemigrations**

**migrate**

**sendtestemail**

**shell**

**showmigrations**

**sqlflush**

**sqlmigrate**

**sqlsequencereset**

**squashmigrations**

**startapp**

**startproject**

**test**

**testserver**

**[sessions]**

**clearsessions**

**[staticfiles]**

**collectstatic**

**findstatic**

**runserver**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py runserver**

**Performing system checks...**

**System check identified no issues (0 silenced).**

**June 25, 2020 - 17:08:07**

**Django version 2.0.5, using settings 'firstproject.settings'**

**Starting development server at http://127.0.0.1:8000/**

**Quit the server with CONTROL-C.**

**Error: That port is already in use.**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd firstproject/**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**\_\_init\_\_.py \_\_pycache\_\_ settings.py urls.py wsgi.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd..**

**bash: cd..: command not found**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ cd ..**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py**

**Type 'manage.py help <subcommand>' for help on a specific subcommand.**

**Available subcommands:**

**[auth]**

**changepassword**

**createsuperuser**

**[contenttypes]**

**remove\_stale\_contenttypes**

**[django]**

**check**

**compilemessages**

**createcachetable**

**dbshell**

**diffsettings**

**dumpdata**

**flush**

**inspectdb**

**loaddata**

**makemessages**

**makemigrations**

**migrate**

**sendtestemail**

**shell**

**showmigrations**

**sqlflush**

**sqlmigrate**

**sqlsequencereset**

**squashmigrations**

**startapp**

**startproject**

**test**

**testserver**

**[sessions]**

**clearsessions**

**[staticfiles]**

**collectstatic**

**findstatic**

**runserver**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py runserver**

**Performing system checks...**

**System check identified no issues (0 silenced).**

**June 25, 2020 - 17:08:07**

**Django version 2.0.5, using settings 'firstproject.settings'**

**Starting development server at http://127.0.0.1:8000/**

**Quit the server with CONTROL-C.**

**Error: That port is already in use.**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py runserver**

**Performing system checks...**

**System check identified no issues (0 silenced).**

**June 25, 2020 - 17:20:47**

**Django version 2.0.5, using settings 'firstproject.settings'**

**Starting development server at http://127.0.0.1:8000/**

**Quit the server with CONTROL-C.**

**Error: That port is already in use.**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ conda info --envs**

**# conda environments:**

**#**

**base \* /Users/farhanaakter/opt/anaconda3**

**ML\_env /Users/farhanaakter/opt/anaconda3/envs/ML\_env**

**backend /Users/farhanaakter/opt/anaconda3/envs/backend**

**demo\_env /Users/farhanaakter/opt/anaconda3/envs/demo\_env**

**djangoproject /Users/farhanaakter/opt/anaconda3/envs/djangoproject**

**finaldjangosetup /Users/farhanaakter/opt/anaconda3/envs/finaldjangosetup**

**nameofenv /Users/farhanaakter/opt/anaconda3/envs/nameofenv**

**webapp /Users/farhanaakter/opt/anaconda3/envs/webapp**

**(base) Mahmuduls-Air-2:firstproject farhanaakter$ conda activate backend**

**(backend) Mahmuduls-Air-2:firstproject farhanaakter$ ls**

**db.sqlite3 firstproject manage.py**

**(backend) Mahmuduls-Air-2:firstproject farhanaakter$ django-admin startapp firstapp**

**(backend) Mahmuduls-Air-2:firstproject farhanaakter$ python manage.py runserver**

**Performing system checks...**

**System check identified no issues (0 silenced).**

**You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin.**

**Run 'python manage.py migrate' to apply them.**

**June 25, 2020 - 17:35:51**

**Django version 2.1.5, using settings 'firstproject.settings'**

**Starting development server at http://127.0.0.1:8000/**

**Quit the server with CONTROL-C.**

**Error: That port is already in use.**

**DAY -11-12 ESTIMATE  (2-3 hours)**

**TASK ONE: NUMBERS AND VARIABLES**

1. Create three variables in a single a line and assign different values to them and make sure their data types are different. Like one is int, another one is float and the last one is a string.

2. Create a variable of value type complex and swap it with another variable whose value is an integer.

3. Swap two numbers using the third variable as the result name and do the same task without using any third variable.

4. Write a program to print the value given by the user by using both Python 2.x and Python 3.x Version.

5. Write a program to complete the task given below:

* Ask the user to enter any 2 numbers in between 1-10 and add both of them to another variable call z.
* Use z for adding 30 into it and print the final result by using variable result.

6. Write a program to check the data type of the entered values. HINT: Printed output should say -  The input value data type is: int/float/string/etc

7. If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value. If Yes then Why?

**TASK TWO: OPERATORS AND DECISION MAKING STATEMENT**

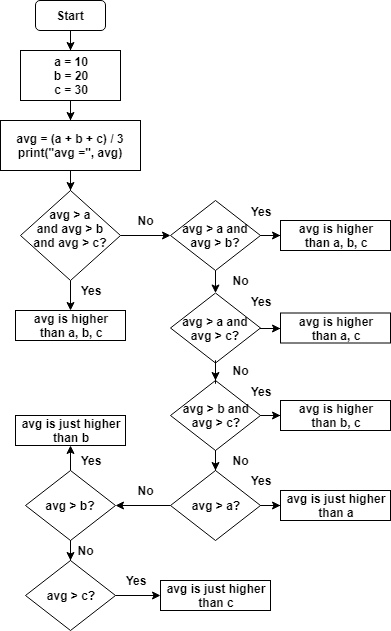
1. Write a program in Python to perform the following operation:

* If a number is divisible by 3 it should print “Consultadd” as a string
* If a number is divisible by 5 it should print “c” as a string
* If a number is divisible by both 3 and 5 it should print “Consultadd Python Training” as a string.

2. Write a program in Python to perform the following operator based task:

* Ask user to choose the following option first:
  + If User Enter 1 - Addition
  + If User Enter 2 - Subtraction
  + If User Enter 3 - Division
  + If USer Enter 4 - Multiplication
  + If User Enter 5 - Average
* Ask user to enter the 2 numbers in a variable for first and second for the first 4 options mentioned above.
* Ask user to enter two more numbers as first1 and second2 for calculating the average as soon as user choose an option 5.
* At the end if the answer of any operation is Negative print a statement saying “zsa”
* NOTE: At a time user can perform one action at a time.

3. Write a program in Python to implement the given flowchart:



4. Write a program in Python to break and continue if the following cases occurs:

* If user enters a negative number just break the loop and print “It’s Over”
* If user enters a positive number just continue in the loop and print “Good Going”

5.   Write a program in Python which will find all such numbers which are divisible  by 7 but are not a multiple of 5, between 2000 and 3200.

6. What is the output of the following code examples?

* x=123

      for i in x:

    print(i)

* i = 0

while i < 5:

    print(i)

    i += 1

    if i == 3:

        break

else:

    print(“error”)

* count = 0

while True:

    print(count)

    count += 1

    if count >= 5:

        Break

7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.

       Expected output: 0 1 2 4 5

Note: Use ‘continue’ statement

8.  Write a program that accepts a string as an input from user and calculate the number of digits and letters.

     Expected output: consul12

     Letters 6

     Digits 2

9. Read the two parts of the question below:

* Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.
* Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. ( The program continues as long as a user has not answered “no” and has not guessed the correct number)

10. Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as

            counter=1

While counter <= 5:

print(“Type in the”, counter, “number”

counter=counter+1

The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess it stops and prints “Game over!”.

11. In the previous question, insert “break” after the “Good guess!” print statement. “break” will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.

**TASK THREE: DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

2. Create a list of size 5 and execute the slicing structure

3.          Write a program to get the sum and multiply of all the items in a given list.

4.   Find the largest and smallest number from a given list.

5. Create a new list which contains the specified numbers after removing the even numbers from a predefined list.

6. Create a list of first and last 5 elements where the values are square of numbers between 1 and 30 (both included).

7. Write a program to replace the last element in a list with another list.

Sample data: [[1,3,5,7,9,10],[2,4,6,8]]

Expected output: [1,3,5,7,9,2,4,6,8]

8. Create a new dictionary by concatenating the following two dictionaries:

a={1:10,2:20}

b={3:30,4:40}

Expected Result: {1:10,2:20,3:30,4:40}

9. Create a dictionary that contains a number (between 1 and n) in the form(x,x\*x).

Sample data (n=5)

Expected Output: {1:1,2:4,3:9,4:16,5:25}

10. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program:

34,67,55,33,12,98

The output should be:

[‘34’,’67’,’55’,’33’,’12’,’98’]

(‘34’,’67’,’55’,’33’,’12’,’98’)

**WEEKEND ACTIVITY ON DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Types like int, string, complex and float.

2. Create a list of size 5 and execute the slicing structure

3. Create a list of given structure and run

**x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]**

* Access list [1, 2, 3, 4]
* Access list [600,  700]
* Access list [100, 300, 500, 600, 800]
* Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]
* Access list [10]
* Access list [ ]

4. Create a list of thousand number using range and xrange and see the difference between each other.

5. How Tuple is beneficial as compare to the list?

6. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

7. Write a program in Python to reverse a string and print only the vowel alphabet if exist in the string with their index.

8. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which has even length of word.

9. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

**x=[1,2,3,4,5,6,7,8,9,-1]**

10. Write a program in Python to complete the following task:

* Create two different list as in even\_list and odd\_list
* Ask user to enter the number in the range of 1,50 and make sure if the entered number is even append it to the even\_list and if the entered number is odd append it to the odd list.
* Keep that in mind you can only add 5 items in each list
* Make sure once you entered the total 5 element calculate the sum of the list and return the maximum out of the list.

11. Write a program to find out the occurrence of a specific word from an alphanumeric statement. **Example:** 12abcbacbaba344ab

**Output:** a=5 b=5 c=2 make sure you should avoid the numbers in you logic

12.          Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

**WEEKEND ACTIVITY ON FUNCTIONS**

1. Write a program to reverse a string.

Sample data: “1234abcd”

Expected Output: “dcba4321”

2. Write a function that accepts a string and calculate the number of uppercase letters and lowercase letters.

Expected Output:

No. of Upper case characters : 3

No. of Lower case Characters : 12

3.        Create a function that takes a list and returns a new list with unique elements of the first list.

4.         Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

5.         Write a program that accepts a sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Sample input:

Hello world

Practice makes perfect

Expected Output:

HELLO WORLD

PRACTICE MAKES PERFECT

6.          Define a function that can receive two integral numbers in string form and compute their sum and print it in console.

7.        Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print all strings line by line.

8.        Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20.

9.         Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

Example: If the limit is 3 , it should print:

0 EVEN

1 ODD

2 EVEN

3 ODD

10. Write a program which can filter() to make a list whose elements are even number between 1 and 20 ( both included)

11. Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]

Hints: Use map() to generate a list.

          Use filter() to filter elements of a list

            Use lambda to define anonymous functions

12. Write a function to compute 5/0 and use try/except to catch the exceptions

13. Flatten the list [[1,2,3].,[4,5],[6,7,8]] into [1,2,3,4,5,6,7,8] using reduce

Goal : Turn [1,2,3,4,5,6,7] to 1234567

 14.

(i) def foo():

    try:

        return 1

    finally:

        return 2

k = foo()

print(k)

**DAY -14 & 15 ESTIMATE  (2-3 hours)**

**DESCRIPTION (Django == 2.1.5)**

This assignment is based on web development using the Django framework of python. In this assignment you will be asked to create one project with one application followed by the proper structure of the application. Use Model, Views and Template to take the full advantage of Django. Make sure the final out of the application should display according to the following

constraint.

**CONSTRAINT: 01**

-- Give the name of the project as BootProject

-- Give the name of the application as Bootapp

-- Allow views.py to be get properly mapped with urls.py file.

-- Use a template to create a generic HTML file that can display data like- Hi! Welcome to the innovation with Bootcamp training.

-- While working with the template make sure you give name "TEMPLATE\_PATH" to actually connect to the os path.

**CONSTRAINT: 02**

-- Make use of the Static files concept to display jpg file on the web page.

-- Make use of some other static files like CSS or JavaScript to change the font or color of the text.

-- Create one models classes as - Name, ID, Contact, Address and migrate them into the database.

-- Initially migrate them to Django SQLLite

-- Make sure to create a super user & Get into the admin console

-- Connect with PostgreSQL

-- I need to install psycopg2-binary

ADDTIONAL :

-- Create a model with your choice of Database design

# Python Basic Operators

Operators are the constructs which can manipulate the value of operands.

Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.

## **Types of Operator**

Python language supports the following types of operators.

* Arithmetic Operators
* Comparison (Relational) Operators
* Assignment Operators
* Logical Operators
* Bitwise Operators
* Membership Operators
* Identity Operators

Let us have a look on all operators one by one.

## **Python Arithmetic Operators**

Assume variable a holds 10 and variable b holds 20, then −

[ [Show Example](https://www.tutorialspoint.com/python/arithmetic_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + Addition | Adds values on either side of the operator. | a + b = 30 |
| - Subtraction | Subtracts right hand operand from left hand operand. | a – b = -10 |
| \* Multiplication | Multiplies values on either side of the operator | a \* b = 200 |
| / Division | Divides left hand operand by right hand operand | b / a = 2 |
| % Modulus | Divides left hand operand by right hand operand and returns remainder | b % a = 0 |
| \*\* Exponent | Performs exponential (power) calculation on operators | a\*\*b =10 to the power 20 |
| // | Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) − | 9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0 |

## **Python Comparison Operators**

These operators compare the values on either sides of them and decide the relation among them. They are also called Relational operators.

Assume variable a holds 10 and variable b holds 20, then −

[ [Show Example](https://www.tutorialspoint.com/python/comparison_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | If the values of two operands are equal, then the condition becomes true. | (a == b) is not true. |
| != | If values of two operands are not equal, then condition becomes true. | (a != b) is true. |
| <> | If values of two operands are not equal, then condition becomes true. | (a <> b) is true. This is similar to != operator. |
| > | If the value of left operand is greater than the value of right operand, then condition becomes true. | (a > b) is not true. |
| < | If the value of left operand is less than the value of right operand, then condition becomes true. | (a < b) is true. |
| >= | If the value of left operand is greater than or equal to the value of right operand, then condition becomes true. | (a >= b) is not true. |
| <= | If the value of left operand is less than or equal to the value of right operand, then condition becomes true. | (a <= b) is true. |

## **Python Assignment Operators**

Assume variable a holds 10 and variable b holds 20, then −

[ [Show Example](https://www.tutorialspoint.com/python/assignment_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Assigns values from right side operands to left side operand | c = a + b assigns value of a + b into c |
| += Add AND | It adds right operand to the left operand and assign the result to left operand | c += a is equivalent to c = c + a |
| -= Subtract AND | It subtracts right operand from the left operand and assign the result to left operand | c -= a is equivalent to c = c - a |
| \*= Multiply AND | It multiplies right operand with the left operand and assign the result to left operand | c \*= a is equivalent to c = c \* a |
| /= Divide AND | It divides left operand with the right operand and assign the result to left operand | c /= a is equivalent to c = c / a |
| %= Modulus AND | It takes modulus using two operands and assign the result to left operand | c %= a is equivalent to c = c % a |
| \*\*= Exponent AND | Performs exponential (power) calculation on operators and assign value to the left operand | c \*\*= a is equivalent to c = c \*\* a |
| //= Floor Division | It performs floor division on operators and assign value to the left operand | c //= a is equivalent to c = c // a |

## **Python Bitwise Operators**

Bitwise operator works on bits and performs bit by bit operation. Assume if a = 60; and b = 13; Now in the binary format their values will be 0011 1100 and 0000 1101 respectively. Following table lists out the bitwise operators supported by Python language with an example each in those, we use the above two variables (a and b) as operands −

a = 0011 1100

b = 0000 1101

-----------------

a&b = 0000 1100

a|b = 0011 1101

a^b = 0011 0001

~a  = 1100 0011

There are following Bitwise operators supported by Python language

[ [Show Example](https://www.tutorialspoint.com/python/bitwise_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & Binary AND | Operator copies a bit to the result if it exists in both operands | (a & b) (means 0000 1100) |
| | Binary OR | It copies a bit if it exists in either operand. | (a | b) = 61 (means 0011 1101) |
| ^ Binary XOR | It copies the bit if it is set in one operand but not both. | (a ^ b) = 49 (means 0011 0001) |
| ~ Binary Ones Complement | It is unary and has the effect of 'flipping' bits. | (~a ) = -61 (means 1100 0011 in 2's complement form due to a signed binary number. |
| << Binary Left Shift | The left operands value is moved left by the number of bits specified by the right operand. | a << 2 = 240 (means 1111 0000) |
| >> Binary Right Shift | The left operands value is moved right by the number of bits specified by the right operand. | a >> 2 = 15 (means 0000 1111) |

## **Python Logical Operators**

There are following logical operators supported by Python language. Assume variable a holds 10 and variable b holds 20 then

[ [Show Example](https://www.tutorialspoint.com/python/logical_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| and Logical AND | If both the operands are true then condition becomes true. | (a and b) is true. |
| or Logical OR | If any of the two operands are non-zero then condition becomes true. | (a or b) is true. |
| not Logical NOT | Used to reverse the logical state of its operand. | Not(a and b) is false. |

## **Python Membership Operators**

Python’s membership operators test for membership in a sequence, such as strings, lists, or tuples. There are two membership operators as explained below −

[ [Show Example](https://www.tutorialspoint.com/python/membership_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| in | Evaluates to true if it finds a variable in the specified sequence and false otherwise. | x in y, here in results in a 1 if x is a member of sequence y. |
| not in | Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. | x not in y, here not in results in a 1 if x is not a member of sequence y. |

## **Python Identity Operators**

Identity operators compare the memory locations of two objects. There are two Identity operators explained below −

[ [Show Example](https://www.tutorialspoint.com/python/identity_operators_example.htm) ]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| is | Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. | x is y, here **is** results in 1 if id(x) equals id(y). |
| is not | Evaluates to false if the variables on either side of the operator point to the same object and true otherwise. | x is not y, here **is not** results in 1 if id(x) is not equal to id(y). |

## **Python Operators Precedence**

The following table lists all operators from highest precedence to lowest.

[ [Show Example](https://www.tutorialspoint.com/python/operators_precedence_example.htm) ]

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **\*\***  Exponentiation (raise to the power) |
| 2 | **~ + -**  Complement, unary plus and minus (method names for the last two are +@ and -@) |
| 3 | **\* / % //**  Multiply, divide, modulo and floor division |
| 4 | **+ -**  Addition and subtraction |
| 5 | **>> <<**  Right and left bitwise shift |
| 6 | **&**  Bitwise 'AND' |
| 7 | **^ |**  Bitwise exclusive `OR' and regular `OR' |
| 8 | **<= < > >=**  Comparison operators |
| 9 | **<> == !=**  Equality operators |
| 10 | **= %= /= //= -= += \*= \*\*=**  Assignment operators |
| 11 | **is is not**  Identity operators |
| 12 | **in not in**  Membership operators |
| 13 | **not or and**  Logical operators |