



MALAD KANDIVALI EDUCATION SOCIETY'S
NAGINDAS KHANDWALA COLLEGE OF COMMERCE,
ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS KHANDWALA
COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
(AUTONOMOUS)

(Reaccredited 'A' Grade by NAAC)
(AFFILIATED TO UNIVERSITY OF MUMBAI)
(ISO 9001:2015)

CERTIFICATE

TALAVIYA HEETA

Name: Mr./Ms. _____

Roll No: __92__ Programme: BSc IT Semester: II

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **IT platforms, Tools and Practices** (Course Code: **2026UISTP**) for the partial fulfillment of Second Semester of BSc IT/CS during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

External Examiner

Subject-In-Charge
(Ms.Sweety Garg)

Date of Examination: (College Stamp)

Sr. No.	DATE	TITLE	SIGN
1.	02/02/21	INTRODUCTION and CONTRIBUTING TO WIKIPEDIA a) What is Wikipedia? b) Steps to Create Account on Wikipedia c) Creating Page on Wikipedia d) Edit your page	
2.	09/02/21	Creating account, repository on GitHub and Cloning repository in GitHub Page	
3.	16/02/21	BASIC UNDERSTANDING ON FREE AND OPEN-SOURCE SOFTWARE a) Describe Open-Source Software with Example. b) Describe Free Software with Example c) Difference between Free and Open-Source Software.	
4.	23/02/21	WRITING EMAIL	
5.	25/02/21	Using practical examples, describe green computing. List and explain the steps that you take to contribute to green computing	
6.	02/03/21	WRITING BLOGS	
7.	09/03/21	Implementing coding practices in Python using PEP8.	
8.	15/03/21	PRESENTATION: _____PEP8_____	

PRATICAL1:

a) Description about Wikipedia and its features

Wikipedia is a free multilingual open collaborative online encyclopedia created and maintained by community of volunteer editors using a Wiki-based editing system .

It is one of the 15 most popular website as ranked by Alexa as of January 2021 and the economist newspaper placed it as the 13-most-visited place on the web.

Wikipedia was launched on January 15,2001,by Jimmy Wales and Larry Sanger

Others collaborative encyclopedia were attempted before wikipedia but none were as successful.

Wikipedia began as complementary project for Nupedia,a free online English-language encyclopedia .

It was founded on March 9,2000 under ownership of Bomis,a web portal company

b) Creating Account on Wikipedia

Steps for creating an account on Wikipedia:

- 1) First go to website of Wikipedia or just simply write "CREATE ACCOUNT ON WIKIPEDIA" in Google
- 2) Then click on Create Account
- 3) Then write your details as per asked by them
- 4) Click on Create Account . Your account is created

Step 1

Special page

Create account

Username (help me choose)

Password

Confirm password

Email address (optional)

To protect the wiki against automated account creation, we kindly ask you to enter the words that appear below in the box (more info):

CAPTCHA Security check

Wikipedia
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

Upload file
Special pages

Wikiped

wikiHow to Contribute to Wikipedia

c) Creating your page on Wikipedia

Steps for creating page on Wikipedia:

- 1) It is very important to have Account in Wikipedia
- 2) Click on the Red link
- 3) When you click red link you will be transported to a blank pages
- 4) On that page you can enter any text you want
- 5) Then click on the "PUBLISH CHANCES" button.

Your page is created on Wikipedia.



d) Editing your page on Wikipedia

Steps for editing page on Wikipedia:

1) Go to any editable page/your won wiki page

2) Click on "EDIT" tab at the top of Wikipedia page

This will take you to the new page containing editable contents

3) When you have finished editing you should write a short edit summary below edit box

4) To see how the page looks with your edits press "**SHOW PREVIEW**" button

5) After verifying your contents and edits click on "PUBLISH CHANGES" button.

tab and filling in the blanks. You can also Re-use citations that are already in the article.

By age 16, she was working as a [governess](#).^[5]^[6]

here she studied

Girton College. E

d the choral soc

In 1880, [Ayrton](#)

Cambridge gave

idon, which awar

neering wo

ey by teaching a

skills to practical use – she taught at [Notting Hill](#) and [Falling High School](#) and was

Cancel

Add a citation

Automatic

Manual

Re-use

 Website

 Book

 News

 Journal

 Basic form

Then press the big blue Publish button at the top of the page

When the box appears, tell people what you added and press the blue information, then click Cite to add a source

Resume editing

Save your changes

Publish changes

Edit summary (Briefly describe your changes)

Added information about early life

☐ This is a minor edit

☒ Watch this page

966

By publishing changes: you agree to the [Terms of Use](#), and you irrevocably agree to



 Zeromonk

Article Talk Read Edit Edit source View his

Paragraph    Cite   Insert 

PRATICAL2:

Creating account, repository on Github and Cloning repository in Github

a) Creating Account

Steps for creating Account:

- 1) Go To “**github.com**” in a web browser
 - 2) Enter your all personal details
 - 3) Click on the “ CREATE AN ACCOUNT” button below the form
 - 4) complete the puzzle given there
 - 5) Click on “VERIFY EMAIL ADDRESS” button below
 - 6) Click on Continue and then click on Submit
- your account has been created

Step 1

Special page

Create account

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
Upload file
Special pages

Username (help me choose)
Enter your username

Password
Enter a password

Confirm password
Enter password again

Email address (optional)
Enter your email address

To protect the wiki against automated account creation, we kindly ask you to enter the words that appear below in the box (more info):

CAPTCHA Security check

wiki How to Contribute to Wikipedia

Step 3

wikihowsherman@gmail.com ✓

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password *

.....

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Verify account

✓

By clicking "Create an account" below, you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account-related emails.

Create an account

b) Creating Repository

Steps to create Repository:

- 1) In upper right corner of any page use +drop-down menu, and select "NEW REPOSITORY"
- 2) Type a short memorable name for your repository
- 3) Add description of your repository .

It is optional you can skip this step

- 4) Choose a repository visibility

For more information see " About repository visibility"

- 5) Select "**Initialize this repository with a README**"


- 6) Click on "Create Repository"

your repository has been created

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



 heeta25 ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about **probable-meme?**



Description (optional)

-
- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.
-

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
-

-
- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.
-

Initialize this repository with:

Skip this step if you're importing an existing repository.

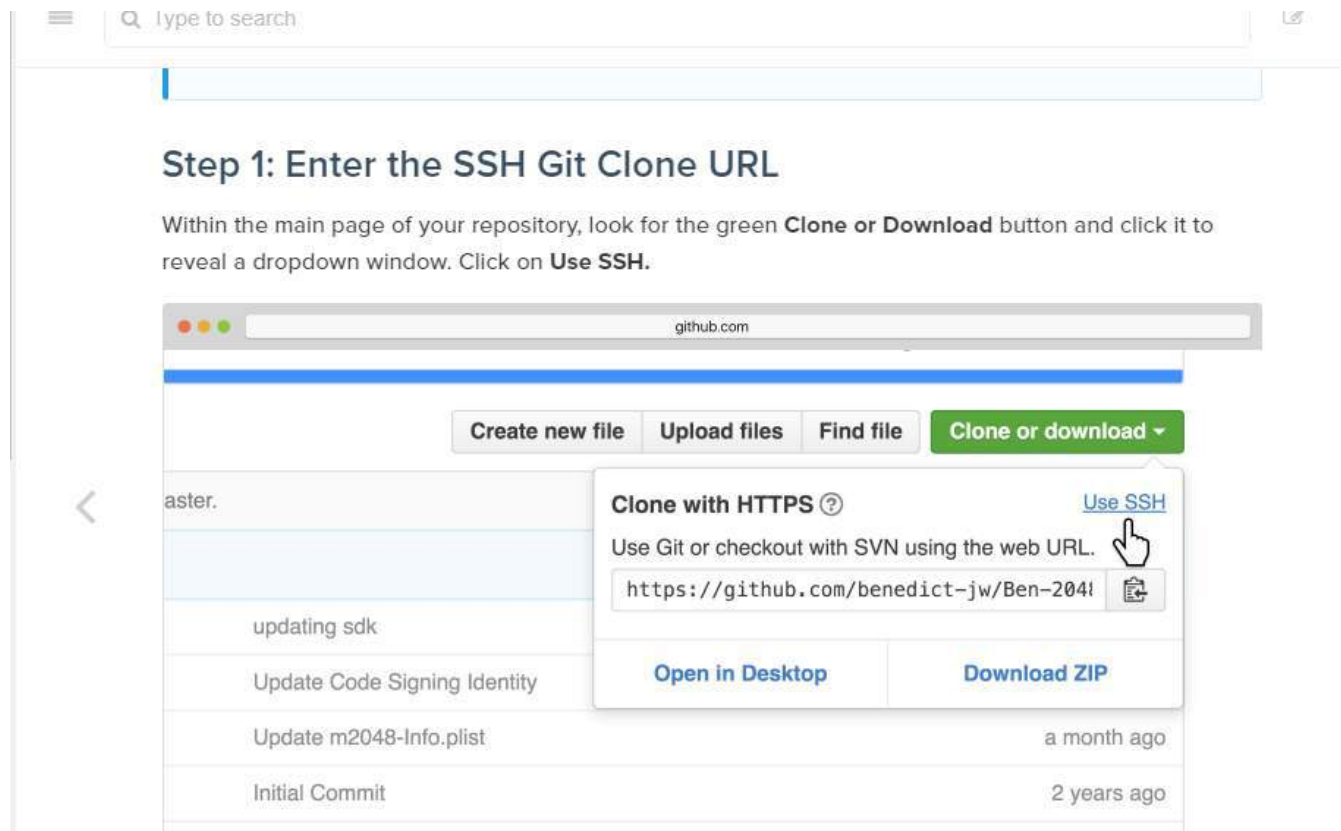
- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)
-

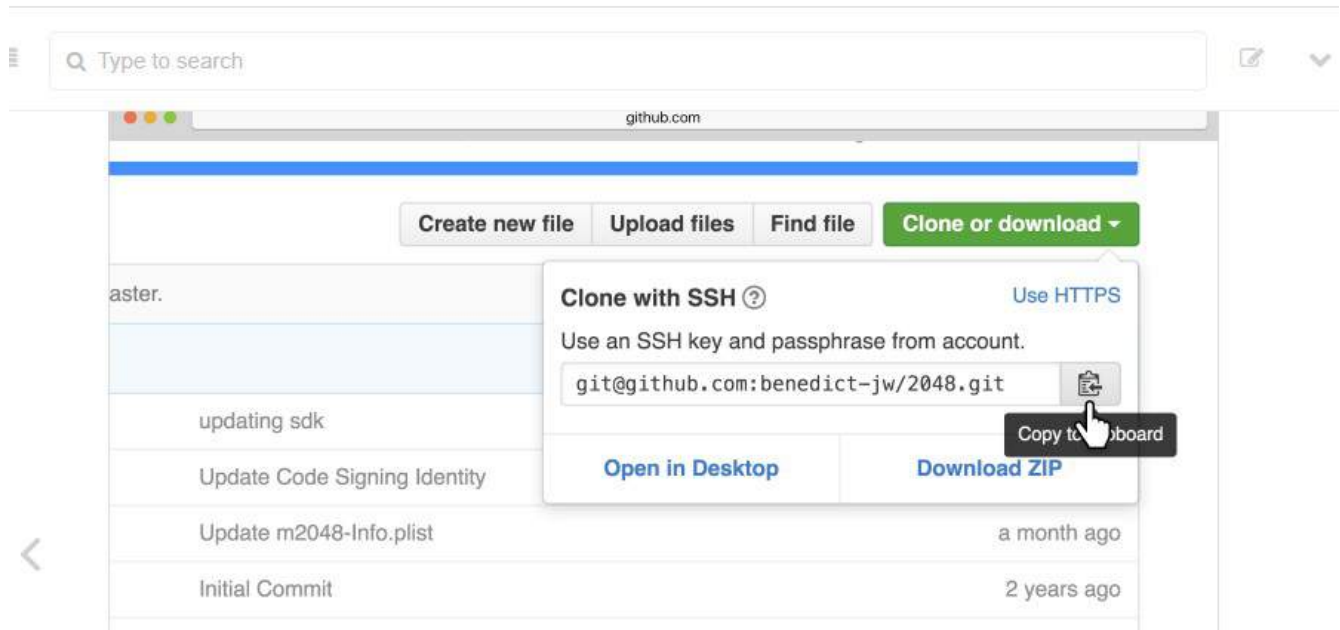
Create repository

c) Cloning repository

Steps for Cloning Repository:

- 1) On your right side of the screen there will be a green button "CLONE /DOWNLOAD"
- 2) In window that appears select "CLIPBOARD" icon
- 3) Copy that link and paste it on URL





Head over to dashboard, visit [Select source](#) and choose **SSH**.



PRACTICAL 3:

BASIC UNDERSTANDING ON FREE AND OPEN-SOURCE SOFTWARE

a) Describe Open-Source Software with Example

Open-source software also called as OSS is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software to anyone and for any purpose.

Open-source software may be developed in a collaborative public manner. Open-source software is a prominent example of open collaboration

Open-source software is usually easier to obtain than proprietary software, often resulting in increased use.

Open-source code is usually stored in a public repository and shared publicly.

Anyone can access the repository to use the code independently or contribute improvements to design and functionality of overall project

These are the some reasons why open-source software is used:

- Security
- Affordability
- Transparency
- Perpetuity
- Interoperability
- Flexibility
- Localization

Here are some most popular examples of open-source software:

- 1) Firefox is a Web browser which is competing with Internet Explorer
- 2) OpenOffice a competitor to Microsoft Office
- 3) Gimp is a graphic tool with features found
- 4) VLC media player is one of the most popular open-source software examples which is commonly used now –a-days

b) Describe Free Software with Example

Free software is a software that can be freely used, modified and redistributed with only one restriction any redistributed version of software must be distributed with original terms of free use

The definition of software is stipulated as part of the GNU project and by the Free Software Foundation

According to the definition as presented by the Free Software Foundation, the word Free in “free software” implies the idea of freedom rather than not having a cost.

If software is available to be downloaded without being paid for, but the user is not able to modify the source then it is not free software

Free software is software you are free to modify and use for your own purposes

Free software does not mean non commercial

On the contrary, a free program must be available for commercial use, commercial development, and commercial distribution.

Example for Free Software are:

1. LINUX: is one of the most popular free software used by millions of the people
2. PostgreSQL: is an object-relational database. It is currently the most sophisticated free software database available.
3. Apache: is the most widely used web server in the world. More than 56% of the web servers on this planet use Apache

c) Difference between Free and Open-Source Software.

The term “free software” is sometimes misunderstood—it has nothing to do with price. It is about freedom.

When we say Open Source, source code of software is available publicly with Open-Source licenses like GNU (GPL) which allows you to edit source code and distribute it.

Free software is matter of liberty and not of price whereas open-source software does not just mean access to the source code

Open-source software license criteria focus on the availability of the source code and the ability to modify and share it, while free software focus on the user’s freedom to use the program, to modify it, and to share it

These terms are used to compare legal attributes of open-source and free software and other content publicly available licensing to proprietary licenses.

PRATICAL 4

WRITING AN EMAIL:

E-mail is defined as the transmission of messages on the Internet.

It is one of the most commonly used features over communications networks that may contain text, files, images, or other attachments.

Generally, it is information that is stored on a computer sent through a network to a specified individual or group of individuals.

Email messages include three components, which are as follows:

- **Message envelope:** It depicts the email's electronic format.
- **Message header:** It contains email subject line and sender/recipient information.
- **Message body:** It comprises images, text, and other file attachments.

SOME COMMON TYPES OF E MAIL ARE

1. NEWS LETTER EMAIL
2. LEAD NURTURING EMAIL
3. PROMOTIONAL EMAIL
4. TRANSACTIONAL EMAIL
5. PLAIN -TEXT EMAIL
6. WELCOME EMAIL

SOME COMMON SITES OF EMAIL ARE:

1. Aol
2. Zoho
3. Gmail
4. Protonmail
5. Com
6. Yahoo
7. Microsoft outlook

STRUCTURE OF EMAIL

B

I

U

- _____
- _____
- _____

1. _____
2. _____
3. _____

▼

Heading 1

5

→ R

From: hitakshitalaviya@gmail.com



To: Ashok Talaviya;



Cc & Bcc

Invitation of Farwell Party 2021

Respected Sir

Regards of the Day,

I, Heeta Talaviya member of ABC Union Club memberno-2032 is writing you this letter to draw your attention towards Farwell Party 2021 organized by our club in memorable completion of 25 years

As you being the member of this club would you like to invite you on this memorable day

Schedule :

Day : Sunday

Time : 7:00 pm

Date : March 8 21

Venue: City red hall, thane(w)

Contact no.000000000000

Your presence , our gesture

Looking forward towards your positive response

Encl:

Signature

PRATICAL 5

Q1) Explain Green computing with its advantages.

Green Computing: Green Computing is the environmentally responsible and eco-friendly use of the computers and their resources.

In broader terms, it is also defining as the study of designing, manufacturing, using and disposing of computer devices in a way that reduces their environment impact.

ADVANTAGES OF GREEN COMPUTING:

- 1) Sustainable computing means reduced energy consumption that leads to reduced GHG emissions and fossil fuel usage
- 2) Green computing is cost effective due to less energy usage and cooling requirements
- 3) Green IT uses non-toxic components which do not pose any health hazard to end users.
- 4) Sustainable computing inspires people to reduce, reuse, and recycle
- 5) Green IT implementations' helps in improving public image of an individual or an organization

Steps to contribute green computing:

- 1) Power down when not in use Seems simple but many of us leave computers powered up for a long time when not in use a large sum of power is being wasted, so if you're not using the computer press the power button to shut it off until needed.
- 2) Use the power saving features All computers include power saving options. Using theses features you can command the computer to do various energy-saving tasks automatically, including shutting off unseared disks, powering off a monitor after a given time or even placing the computer into sleep mode when not in use.
- 3) Use a laptop instead of desktop Laptops are much better for the environment than desktop computers as they have components which require less power.
- 4) Recycle responsibly Computer hardware is filled with different material which can be hazardous to the environment so make sure you dispose of old components effectively. Don't just throw broken technology in the Bin, take the time to trace local recycling organizations

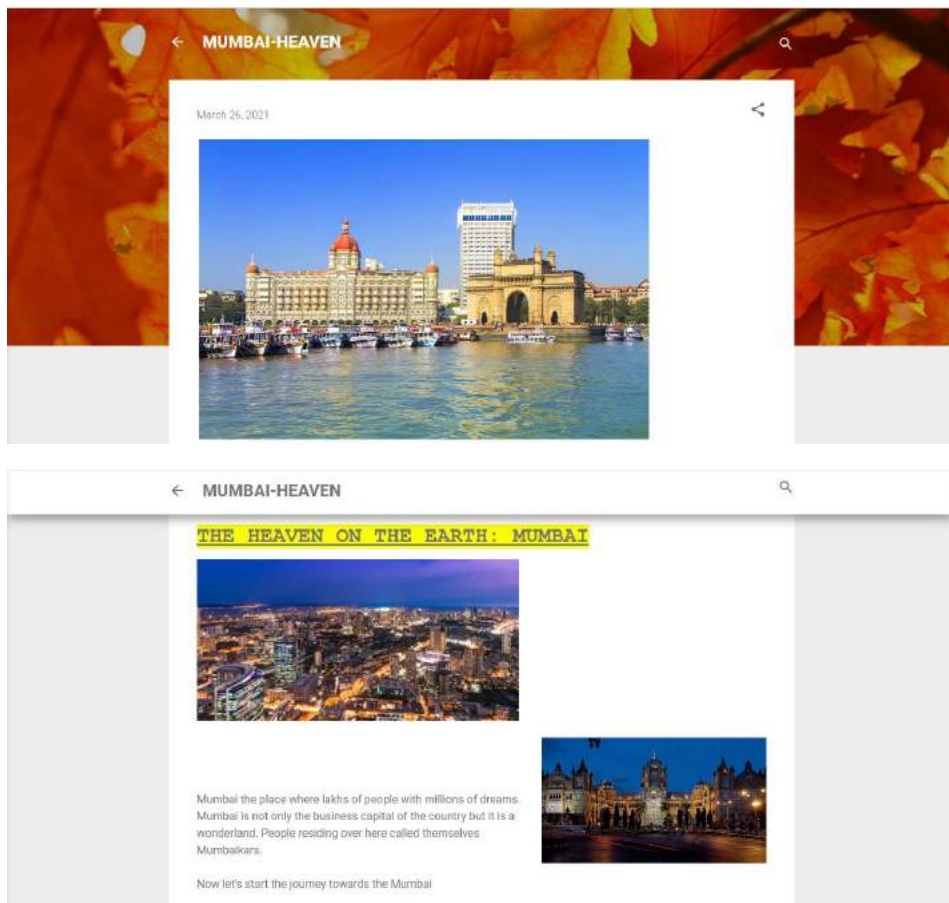
PRATICAL 6

BLOG PRESENTATION

Website for Blog:

<https://92-it.blogspot.com/2021/03/the-heaven-on-earth-mumbai-mumbai-place.html>

Do comment



DO I NEED TO SAY ANYTHING?



The lifeline of Mumbai -the local. The experience jam by the locals has no words. If a person is traveling in locals for the first time, the experience may be scary. You won the battle if you manage to have seat in local during office hours. Local makes the person covers the longest distance within minutes.

THE LIFE OF MUMBAIKARS

← MUMBAI-HEAVEN



THE LIFE OF MUMBAIKARS

It is said People of Mumbai are tireless. People residing in Mumbai are very approachable, they are hardworking, cultural, modern. Mumbaikars in every situation know how to remain cheerful. They know how to tackle the difficulties and enjoy their lives to full extended.

It is impossible to define Mumbaikars as North meets South. Mumbaikars are like a hand though different from each other but still come together and work with dedication.

For making the life of Mumbaikars more enjoyable Mumbai offers hundreds of places to visit.

KALA GHODA FESTIVAL:



This is one type of festival which Mumbai celebrates. It is the festival of art. The KALA Ghoda festival is an amazing festival where different arts, dance, street events take place. It is the most awaited festival each year for art lovers. This is the longest festival. It starts in November and ends in February. Kala Ghoda festival celebrates both modern as well as old art. It is informative as well as enjoyable.



HEART OF THE MUMBAIKARS:

The life of Mumbaiers without food is unimaginable. Mumbai has a variety of food to serve that suits every pocket. Mumbaiers more love street food compared to the restaurant.

1) BOMBAY BHILPURI:



← MUMBAI-HEAVEN



Bhilpuris is a blender of textures and flavors.

The taste of chutneys which makes this simple dish mouthwatering has no words.

Puri used has spoon, onions, and boiled potatoes makes this dish the yummiest ever one can have.

2) VADA PAV:



Mumbai's most famous street food Vada pav. Vada Pav is not less than the burger. It is a type of that snack that needs no time. It is a simple enough snack. Most Mumbaier's day starts with this street food and it with the same. No other taste can match the taste of Vada pav which comes with two chutneys and fried chilies.

3) BOMBAY SANDWICHES:



The name is so true of the dish. As sandwiches combine potatoes, onions, beetroots, tomatoes, two chutneys, cheese butter just stuffed between two pieces of bread tastes the excellent like Mumbaiers apart from religions and culture still works together.

4) PAV BHAJI:

1) PAV BHUJIA:



The most favorite and desired dish in Mumbai. The Indian masalas blended together and added to it has no words. It was invented by Gujarati's for late-night workers. The only dish which is healthy as it includes all most all the vegetables.

MARINE DRIVE:



Marine drive is the concrete road of 3KM located in the southern part of the city. It is the place where one can visit no matter what the time is. During Evening the glimpse of the sunset can be the most memorable one can have. Marine drive also known as Queen's Necklace has made the place a tourist destination.

Marine drive is the concrete road of 3KM located in the southern part of the city. It is the place where one can visit no matter what the time is. During Evening the glimpse of the sunset can be the most memorable one can have. Marine drive also known as Queen's Necklace has made the place a tourist destination.



Mumbai having hectic life offers peace and natural wonder places also.

DANJAY GANDHI NATIONAL PARK:

SANJAY GANDHI NATIONAL PARK



These places in Mumbai bring the peoples of Mumbai nearest to nature and make themselves relax and peaceful in nature's soothing environment

In SANJAY GANDHI NATIONAL PARK there are various places to visit which makes people feel with energy and enthusiasm.

MUMBAI-HEAVEN

WILDLIFE SPECIES



Sanjay Gandhi National park has 250 species including local as well as migratory birds. It has many wild species like lion and tiger also. There special lion and tiger safari build to have a glance at wild heritage species
There are above 160 butterflies which gives a beautiful effect on this park. These butterflies are kept at the Butterfly garden.

VAMBANI TRAIN-MINI TRAIN



This cute little train gives the most beautiful experiences. This train with four coaches for 20 minutes makes us feel like nature is pampering us and we are rotating around nature making the most beautiful experience

BOATING IN RIVERS.



This place makes the Mumbai experience sailing in the ocean and having fun with water riding. Sailing over water can be an inexpressible experience

FROM LATE 20TH CENTURY TO PRESENT

THIS IS NOT LIMITED TO DEFINING MUMBAI. MUMBAI IS THE WONDERLAND DEEPER FOR SO MYSTERY TOO FIND!!



PRATICAL 7

PEP 8 PROGRAM

```
def list_sum(my_list):
    sum = 0
    # INDENTATION

    for i in my_list:
        sum += i
    return sum

# CORRECT CLOSING BRACES
my_list = [1,2,3,
           4,5,6]      # MY LIST OF NUMBERS (INLINE COMMENT)
output = list_sum(my_list)
if output >=10:
    print("You have entered correct number")  # AVOIDING WHITESPACES
else:
    print("You have entered wrong number")
```

PRATICAL 8

ANNE VERONICA	1
PRIYA GUPTA	20
SHUBH PATEL	60
BHAVANA PRAJAPATI	66
POOJA PRAMANIK	67
HEETA TALAVIYA	92
GLORY LITHIYAL	95
NITESH GUPTA	111
AMAN UPADHYAY	132
DEEPAK KESHRI	134
SURYASEN VISHWAKARMA	139

PEP8

PEP8 is a style guide for python code.

- PEP stands for Python Enhancement Proposal, and they describe and document the way python language evolves.
- It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan.
- A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.
- They also provide a reference point (and a standard) for the pythonic way to write code

→ It also has a lot of programming recommendations and useful tips on various topics, which aim to improve readability and reliability of your code.

→ PEP8 features:-

1. Plugin architecture: Adding new checks is easy.
2. Parseable output: Jump to error location in your editor.
3. Small: Just one Python file, requires only stdlib. You can use just the pep8.py file for this purpose.

Naming Conventions

Naming Conventions:

1.Variable

2.Function

3.Class

4.Method

5.Constant

6.Module

7.Package

,

Variable: A variable is created the moment you first assign a value to it

```
#Wrong Way to Initialize or assigning a name to a variable  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
1variable=2 #Variable name started with a number (Wrong Way)  
print(1variable)
```

```
File "<ipython-input-1-d1860915d72c>", line 5  
    1variable=2  
    ^  
SyntaxError: invalid syntax
```

```
#Wrong Way to Initialize or assigning a name to a variable  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
x='Bhavana' #Variable name is too common and not intuitive (Not a Good Way)  
print(x)
```

Bhavana

```
#Wrong Way to Initialize or assigning a name to a variable  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
first_name='Bhavana' #Variable name is self-explanatory and has a readability, and it is seperated using underscores  
print(x)
```

Bhavana

Function: A function is a block of code which only runs when it is called.

```
#Wrong Way to Initialize or assigning a name to a function  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
def ^function(): #Function name should not be started with a Number or special characters  
    print("Not a correct way to represent a function name")  
  
^function()
```

```
File "<ipython-input-5-5f84f1733e34>", line 5  
    def ^function():  
        ^  
SyntaxError: invalid syntax
```

```
#Wrong Way to Initialize or assigning a name to a function  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
def x(): #Function name is too generic and it can create a confusion in enterprise programming  
    print("Function Name is too generic, you can use it but it is not recommended as it is not self-explanatory and intuitive")  
  
x()
```

Function Name is too generic, you can use it but it is not recommended as it is not self-explanatory and intuitive

```
#Wrong Way to Initialize or assigning a name to a function  
#Name Should not start with a number  
#Name should be intuitive and not too common.  
  
def display_function(): #Function name is self explanatory  
    print("Function Name is self explanatory, name can be more intuitive in case of proper functionality")  
  
display_function()
```

Function Name is self explanatory, name can be more intuitive in case of proper functionality

Class: class definitions begin with a class keyword.

```
#Wrong Way to Initialize or assigning a name to a class
#Name Should not start with a number
#Name should be intuitive and not too common.

1class x:
def display_function(): #Function name is self explanatory
    print("Function Name is self explanatory, name can be more intuitive in case of proper functionality")

display_function()
```

File "<ipython-input-9-0547726683a1>", line 5

```
1class x:
  ^
```

SyntaxError: invalid syntax

```
class Employee:
    def accept(self):
        print("Enter Id:")
        self.Id=int(input())
        print("Enter Name:")
        self.name= str(input())
    def display(self):
        print("ID: %d \nName: %s"%(self.Id,self.name))

emp=Employee()
emp.accept()
emp.display()
```

```
Enter Id:
66
Enter Name:
bhavana
ID: 66
Name: bhavana
```

Method: A Python method is a label that you can call on an object; it is a piece of code to execute on that object.

```
#Wrong Way to Initialize or assigning a name to a method  
#Name Should not start with a number  
#Name should be intuitive and not too common.
```

```
1class Method:  
    def display(self):  
        print("This is method function. ")  
  
c = Method()  
c.display()
```

File "<ipython-input-26-3e88b14da450>", line 6

```
1class Method:
```

^

SyntaxError: invalid syntax

```
#Wrong Way to Initialize or assigning a name to a class  
#Name Should not start with a number  
#Name should be intuitive and not too common
```

```
class Product:  
    def __init__(self):  
        self.prod_id = input("Enter the Product ID: ")  
        self.prod_name = input("Enter the Product Name: ")  
        self.total_no = int(input("Enter the total no. of Items Purchase: "))  
        self.unit_price=float(input("Enter the unit Price: "))  
    def display(self):  
        print("Total Price of %d units of Product %s is: %0.2f" %(self.total_no,self.prod_name,self.total_no*self.unit_price))  
  
p1=Product()  
p1.display()
```

```
Enter the Product ID: A20134  
Enter the Product Name: Chocolate  
Enter the total no. of Items Purchase: 7  
Enter the unit Price: 75.50  
Total Price of 7 units of Product Chocolate is: 528.50
```


Constant: A constant is a type of variable whose value cannot be changed.

```
pi = 3.14                #pi is constant
radius=5
print("Area of circle: %0.2f" %(pi*radius*radius))
```

Area of circle: 78.50

Modules: Modules refer to a file containing Python statements and definitions.

```
# to import standard module math

import math
print("The value of pi is", math.pi)
```

The value of pi is 3.141592653589793

Packages: A package is basically a directory with Python files and a file with the name `__init__.py`



Code layout

WITHOUT SPACE

These conventions lead to text that you can read easily, like this:

This would become increasingly hard to read. For example have a look at the example below

```
howwillitlookifwedonothavethespace
```

WITH SPACE

Now here, we will use space and write it in regular English language, so it will be very easy to read.

```
How will it look if we do not have the space
```

Maximum line length and line breaking

PEP 8 guidelines suggest that each line of code (as well as comment lines) should be 79 characters wide or less. This is a common standard that is also used in other languages including R.

CORRECT

```
# Perform some math
a = 1+2
b = 3+4
c = a+b

# Read in and plot some
precip_timeseries = pd.readcsv("precip-2019.csv")
precip_timeseries.plot()
```

#WRONG

```
#Perform some math and do some things
a=1+2
b=3+4
c=a+b
data=pd.readcsv("precip-2019.csv")
data.plot()
```

Should a line break Before or After a Binary Operator

Here, it's harder to see which variable is being added and which is subtracted.

WRONG

```
Total = (Number 1+  
          Number 2-  
          Number 3)
```

You can immediately see which variable is being added or subtracted, as the operator is right next to the variable being operated on.

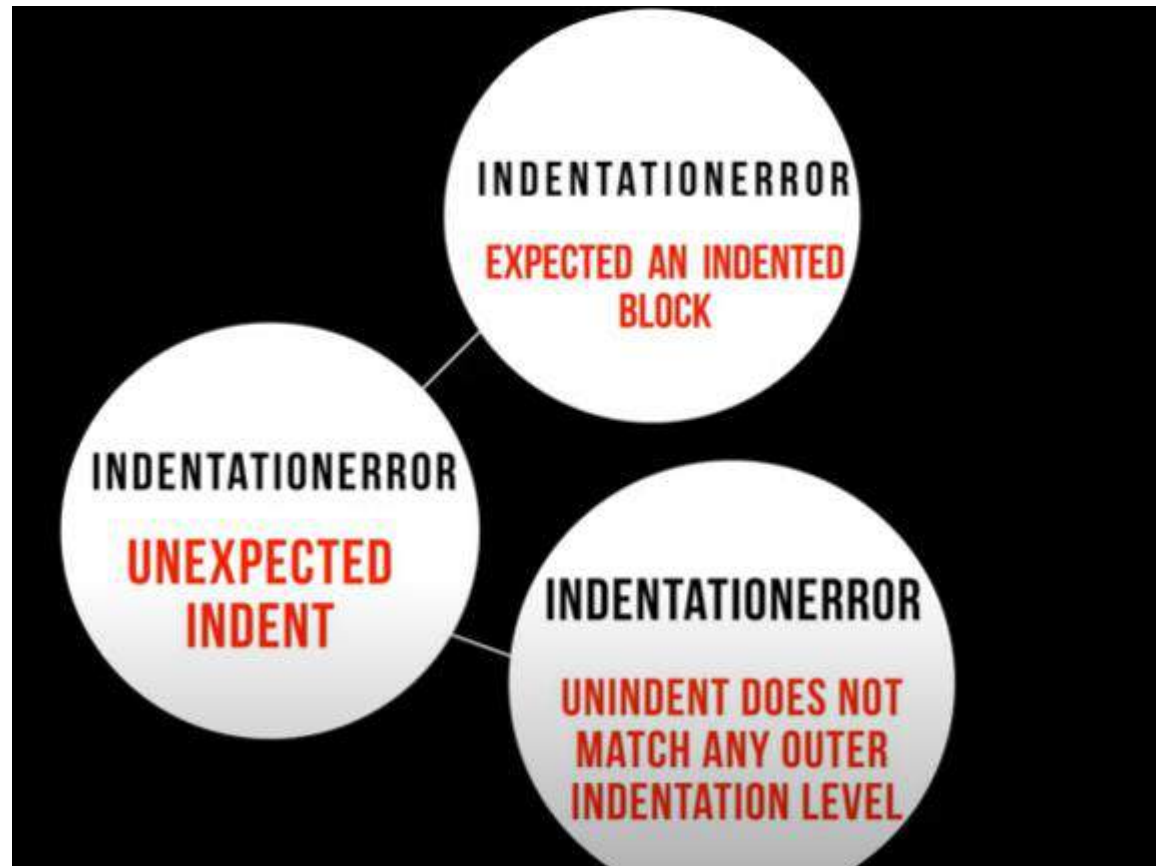
In the below Example

#CORRECT

```
Total = (Number 1  
          + Number 2  
          - Number 3)
```

Indentation

- Indentation is extremely important in Python.
- The Indentation level of lines of code in python determines how statements are grouped together.



1. Expected an indented block



```
x = 2
if x % 2 == 0:
    print("It is an even number")
```



```
File "<ipython-input-20-d2c95d58e212>", line 3
    print("It is an even number")
    ^
```

IndentationError: expected an indented block



```
x = 2
if x % 2 == 0:
    print("It is an even number")
```

It is an even number


2. Unexpected Indent



```
x = 2
if x % 2 == 0:
    print("It is an even number")
```



```
File "<ipython-input-24-a296ed44a7f2>", line 2
    if x % 2 == 0:
      ^
IndentationError: unexpected indent
```



```
x = 2
if x % 2 == 0:
    print("It is an even number")
```

```
It is an even number
```

3. Unindent does not match any outer indentation level

```
def greeting():  
    print("Greetings of the day")  
    return  
  
greeting()
```

File "<ipython-input-30-698032a46f85>", line 3
 return
 ^
IndentationError: unindent does not match any outer indentation level

```
def greeting():  
    print("Greetings of the day")  
    return  
  
greeting()
```

Greetings of the day

Tabs vs. Spaces

➤ Tabs vs. Spaces

The key indentation rules laid out by PEP 8 are the following:

- Use 4 consecutive spaces to indicate indentation.
- Prefer spaces over tabs.

➤ Indentation following line breaks

- Add a comment after the final condition. Due to syntax highlighting in most editors, this will separate the conditions from the nested code:

Not Recommended

```
▶ x = 5  
  if (x > 3 and  
      x < 10):  
      print(x)
```

Recommended

```
▶ x = 5  
  if (x > 3 and  
      x < 10):  
      # Both conditions satisfied  
      print(x)
```

Not Recommended



```
var = function(arg_one, arg_two,  
               arg_three, arg_four)
```

Recommended



```
var = function(  
    arg_one, arg_two,  
    arg_three, arg_four)
```

Not Recommended



```
def function(  
    arg_one, arg_two,  
    arg_three, arg_four):  
    return arg_one
```

Recommended

```
def function(  
    arg_one, arg_two,  
    arg_three, arg_four):  
    return arg_one
```

➤ Where to put the closing Braces

1. Method

```
▶ list_of_numbers = [  
    1, 2, 3,  
    4, 5, 6,  
    7, 8, 9  
]
```

2. Method

```
▶ list_of_numbers = [  
    1, 2, 3,  
    4, 5, 6,  
    7, 8, 9  
]
```

COMMENTS:

Comments are lines that exist in computer programs that are ignored by compilers and interpreters.

Comment begins with a hash mark (#)

Generally, comment looks like this:

this a comment

Because comment does not execute ,when you will run program you will not see any indication of the comment there.

BLOCK COMMENTS:

Each line of block comments starts with a `#` and a single space

Paragraphs inside a block comment are separated by a line containing a single `#`.

Anti-pattern

```
#This comment needs a space  
def print_name(self):  
    print(self.name)
```

Best practice

```
# Comment is correct now  
def print_name(self):  
    print(self.name)
```

INLINE COMMENTS:

Inline comment should be separated by at least two spaces from the comment.

They should start with a `#` and a single space

Inline comments are unnecessary and in fact distracting if they state the obvious

Anti-pattern

```
def print_name(self):  
    print(self.name) #This comment needs a space
```

Best practice

```
def print_name(self):  
    print(self.name) # Comment is correct now
```

DOCSTRING COMMENTS:

A docstring is added as a comment string right below the function, module, or object

RULES:

A docstring is either a single line, or a multi-line comment

In latter case, the first line is short description, and after the first line an empty line follows

This is a basic example of what it looks like:

```
def add(value1, value2):  
    """Calculate the sum of value1 and value2."""  
    return value1 + value2
```

In the Python interactive help system, the docstring is then made available via the `__doc__` attribute.

```
>>> print add.__doc__  
Calculate the sum of value1 and value2.
```


Inline Comments vs Block Comments

Inline comments look like this

```
x = x + 1          # Compensate for border
```

While block comments look like this

```
# Compensate for border.  These comments  
# often cover multiple lines.  
x = x + 1
```

Sample code.

table.py - C:\Users\priya\AppData\Local\Programs\Python\Python38-32\table.py (3.8.2)

File Edit Format Run Options Window Help

#This program returns table for entered number.

def table(n,a):

#Definition of function table()

if a<11:

Expression

print(n,"X",a,"=",n*a)

return table(n,a+1)

else:

pass

x=int(input("Enter no:-"))

table(x,1)

Comments

Function

Indentation

Statement

**Inline
Comment**

Block

Whitespace in Expressions and Statements

1) Whitespace Around Binary Operators

Surround the following binary operators with a single space on either side:

- Assignment operators (=, +=, -=, and so forth)
- Comparisons (==, !=, >, <, >=, <=) and (is, is not, in, not in)
- Booleans (and, not, or)

Note: When = is used to assign a default value to a function argument, do not surround it with spaces.

Python

```
# Recommended
def function(default_parameter=5):
    # ...

# Not recommended
def function(default_parameter = 5):
    # ...
```

- Adding space when there is more than one operator in a statement.

Python

Recommended

```
y = x**2 + 5
```

```
z = (x+y) * (x-y)
```

Not Recommended

```
y = x ** 2 + 5
```

```
z = (x + y) * (x - y)
```

- Adding space to if statements where there are multiple conditions.

Python

Not recommended

```
if x > 5 and x % 2 == 0:
```

```
    print('x is larger than 5 and divisible by 2!')
```

Python

Recommended

```
if x>5 and x%2==0:
```

```
    print('x is larger than 5 and divisible by 2!')
```

Note : Use the same amount of whitespace either side of the operator.

The following is not acceptable :

Python

```
# Definitely do not do this!  
if x >5 and x% 2== 0:  
    print('x is larger than 5 and divisible by 2!')
```

When to Avoid Adding Whitespace

- Trailing space
- Immediately inside parentheses, brackets, or braces:

Python

```
# Recommended
my_list = [1, 2, 3]

# Not recommended
my_list = [ 1, 2, 3, ]
```

- Before a comma, semicolon, or colon:

Python

```
x = 5
y = 6

# Recommended
print(x, y)

# Not recommended
print(x , y)
```

Before the open parenthesis that starts the argument list of a function call:

```
Python

def double(x):
    return x * 2

# Recommended
double(3)

# Not recommended
double (3)
```

Before the open bracket that starts an index or slice:

```
Python

# Recommended
list[3]

# Not recommended
list [3]
```

- Between a trailing comma and a closing parenthesis:

Python

```
# Recommended
tuple = (1,)

# Not recommended
tuple = (1, )
```

- To align assignment operators:

Python

```
# Recommended
var1 = 5
var2 = 6
some_long_var = 7

# Not recommended
var1      = 5
var2      = 6
some_long_var = 7
```


Programming Recommendations

❖ Two Programming Recommendations by PEP-8

A) # Not recommended

```
my_bool = 6 > 5
if my_bool == True:
    return '6 is bigger than 5'
```

B) # Recommended

```
if my_bool:
    return '6 is bigger than 5'
```

In the above program B is recommended over A by the PEP-8

C) # Not recommended

```
my_list = []
if not len(my_list):
    print('List is empty!')
```

D) # Recommended

```
my_list = []
```

```
if not my_list:
```

```
    print('List is empty!')
```

In the above program D is recommended over C by the PEP-8

Q. When to Ignore PEP-8

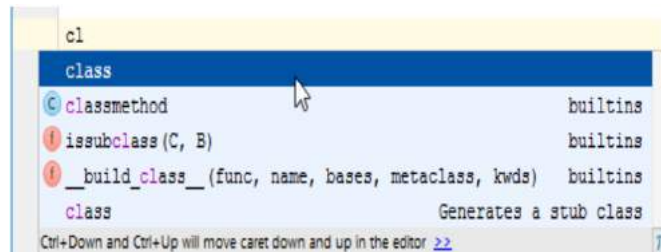
ANSWER: NEVER

Though, there are some guidelines in PEP-8 that are inconvenient in some instances:

- Complying with PEP-8
 - Code surrounding
 - Code compatibility

Tips and Tricks to Help Ensure Your Code Follows PEP 8

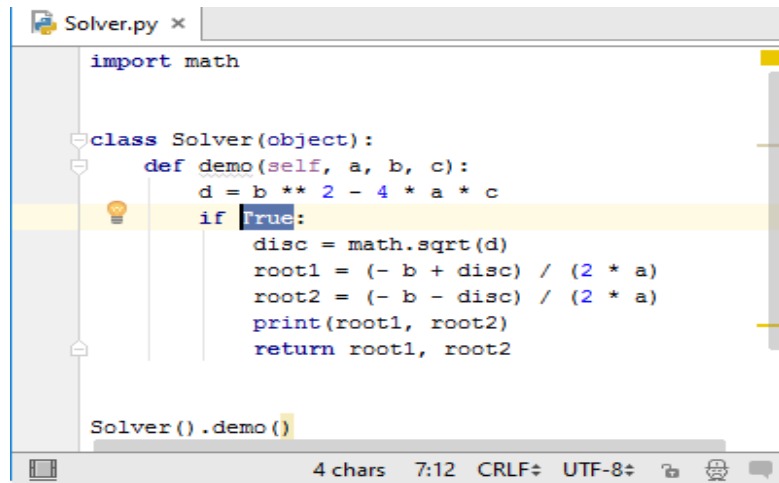
Highlighting code style violations:



(Refer to Code Completion page of the product documentation for details.)

Generating Source code:

Select `if` option from the suggestion list. As you see, PyCharm automatically adds `if True:` and indents the selected lines:



```
import math

class Solver(object):
    def demo(self, a, b, c):
        d = b ** 2 - 4 * a * c
        if True:
            disc = math.sqrt(d)
            root1 = (- b + disc) / (2 * a)
            root2 = (- b - disc) / (2 * a)
            print(root1, root2)
            return root1, root2

Solver().demo()
```

Linter-python-pep8 package

This linter-python-pep8 plugin or Linter provides an interface to pep8. It will be used with files that have the Python syntax.

Installation:

Before using this plugin, you should make sure that pep8 is installed on your system. You can follow following instructions to install pep8:

Install python.

Install pep8 by typing the following in a terminal:

```
pip install pep8
```

Black

Black can be installed by running `pip install black`. It requires Python 3.6.0+ to run. Once Black is installed, you will have a new command-line tool called `black` available to you in your shell, and you're ready to start.

```
$ pip install black
```

Format a Single File:

Let's look at this simple example: here are my two python functions in my python file called `sample_code.py`.

```
def add(a, b):  
    answer = a + b  
  
    return answer  
  
def sub(c, d):  
    answer = c - d  
  
    return answer
```

You can use `black sample_code.py` in the terminal to change the format. After running Black, you will see the following output:

```
reformatted sample_code.py
All done! ✨💎✨
1 file reformatted.
```

Then you open `sample_code.py` to see formatted python code:

```
def add(a, b):
    answer = a + b

    return answer

def sub(c, d):
    answer = c - d

    return answer
```

Example of code and layout.

With space and without space.

WITH SPACE.

*ex- MY NAME IS NITESH

WITHOUT SPACE.

*ex- MYNAMEISNITESH

Maximum line length and line breaking.

Python

- Ex-

```
def function(arg_one, arg_two,  
             arg_three, arg_four):  
    return arg_one
```

Python

- Ex-

```
from mypkg import example1, \  
    example2, example3
```

Should a line break Before or After A Binary Operator.

- Ex-

Python

```
# Recommended
total = (first_variable
        + second_variable
        - third_variable)
```

- Ex-

Python

```
# Not Recommended
total = (first_variable +
        second_variable -
        third_variable)
```

Example of comments.

block comment.

Anti-practice.

Example

```
#This is a comment  
print("Hello, World!")
```

Best-practice.

Example

```
#This is a comment  
#written in  
#more than just one line  
print("Hello, World!")
```

Inline comments.

Anti-pattern.

Python

```
x = 5 # This is an inline comment
```

Best practice.

Python

```
x = 'John Smith' # Student Name
```

Documentation string comment.

```
"""Return a foobang
```

```
Optional plotz says to  
froblicate the bizbaz first.
```

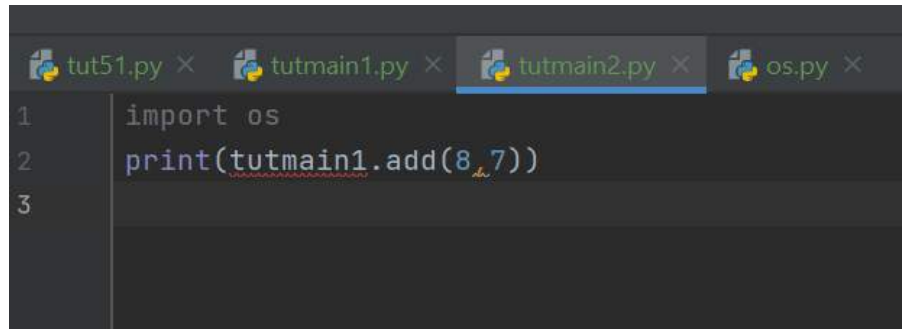
```
"""
```

EXAMPLE OF NAMING CONVENTION

NAMING MODULE WITH HELP OF

PEP8

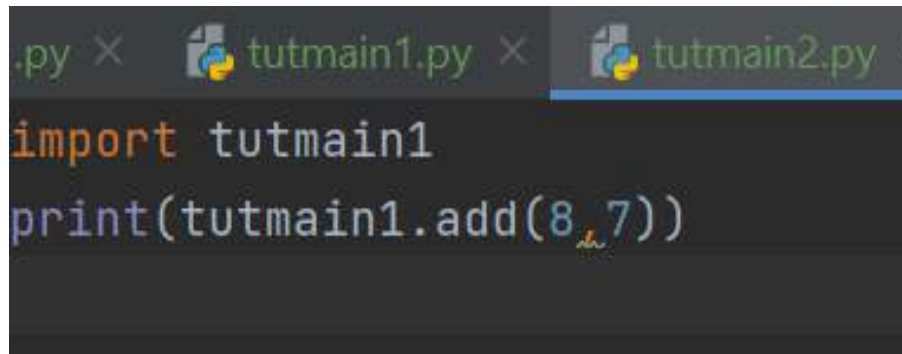
Not recommended



A screenshot of a code editor with four tabs: tut51.py, tutmain1.py, tutmain2.py (selected), and os.py. The code in the selected tab is:

```
1 import os
2 print(tutmain1.add(8,7))
3
```

Recommended



A screenshot of a code editor with three tabs: .py, tutmain1.py, and tutmain2.py (selected). The code in the selected tab is:

```
import tutmain1
print(tutmain1.add(8,7))
```

NAMING VARIABLE WITH HELP OF PEP8

Variable:

```
>>> # Not recommended
>>> x = 'John Smith'
>>> y, z = x.split()
>>> print(z, y, sep=', ')
'Smith, John'
```

```
>>>
```

```
>>> # Recommended
>>> name = 'John Smith'
>>> first_name, last_name = name.split()
>>> print(last_name, first_name, sep=', ')
'Smith, John'
```

EXAMPLES OF INDENTATION

❖ METHODS OF WHERE TO PUT CLOSING BRACES:-

```
list_of_flowers = [  
    rose,sunflower,  
    marigold,jasmine,  
    tulips,lavender  
]
```

```
list_of_flowers = [  
    rose,sunflower,  
    marigold,jasmine,  
    tulips,lavender  
]
```


❖ Methods for following line breaks



```
x=10
if (x < 15 and
    x > 5):
    #Both the conditions satisfied
    print(x)
```



```
x=10
if (x < 15 and
    x > 5):
    print(x)
```

EXAMPLE OF WHITESPACING

1. Adding space when there is more than one operator in a statement.



#recommended

```
b = a**8 + 5
```

```
c = (a+b) * (a-b)
```



#not recommended

```
b = a ** 8 + 5
```

```
c = (a + b) * (a - b)
```

1. Adding space to if statements where there are multiple conditions.

```
#Recommended  
if x>8 and x%2== 0:  
    print('x is larger than 8 and divisible by 2!')
```

```
#not Recommended  
if x > 8 and x % 2 == 0:  
    print('x is larger than 8 and divisible by 2!')
```