



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

Name: Mr. /Ms. \_\_\_\_\_ ANNE VERONICA \_\_\_\_\_

Roll No: \_\_\_\_01\_\_\_\_ 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: **2026UJSTP**) 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.	26-02-21	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.	20-03-21	PRESENTATION: PEP8	

## Practical 1: Introduction and Contribution to wikipedia.

### a) Description about Wikipedia and its features

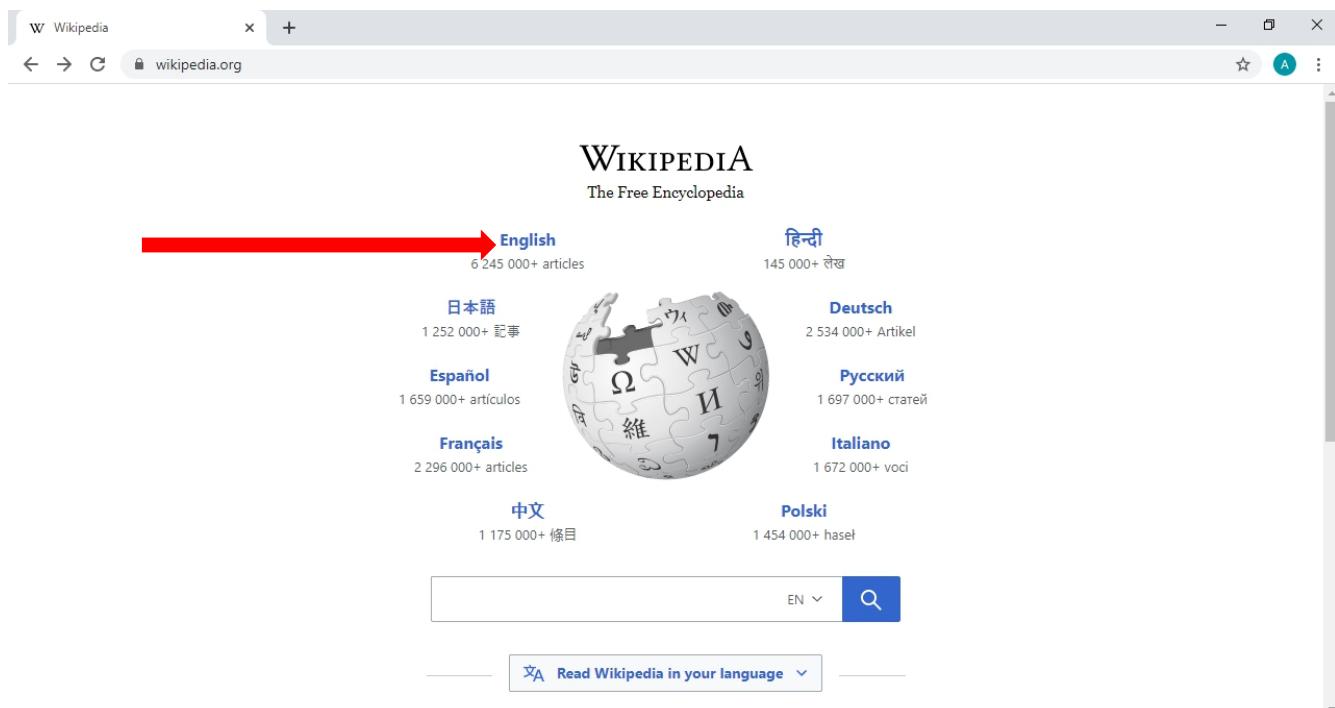
Wikipedia is a free, open content online encyclopedia created through the collaborative effort of a community of users known as Wikipedians. Anyone registered on the site can create an article for publication; registration is not required to edit articles. The site's name comes from wiki, a server program that enables anyone to edit Web site content through their Web browser. Anyone is allowed to add or edit words, references, images, and other media in wikipedia.

Features:

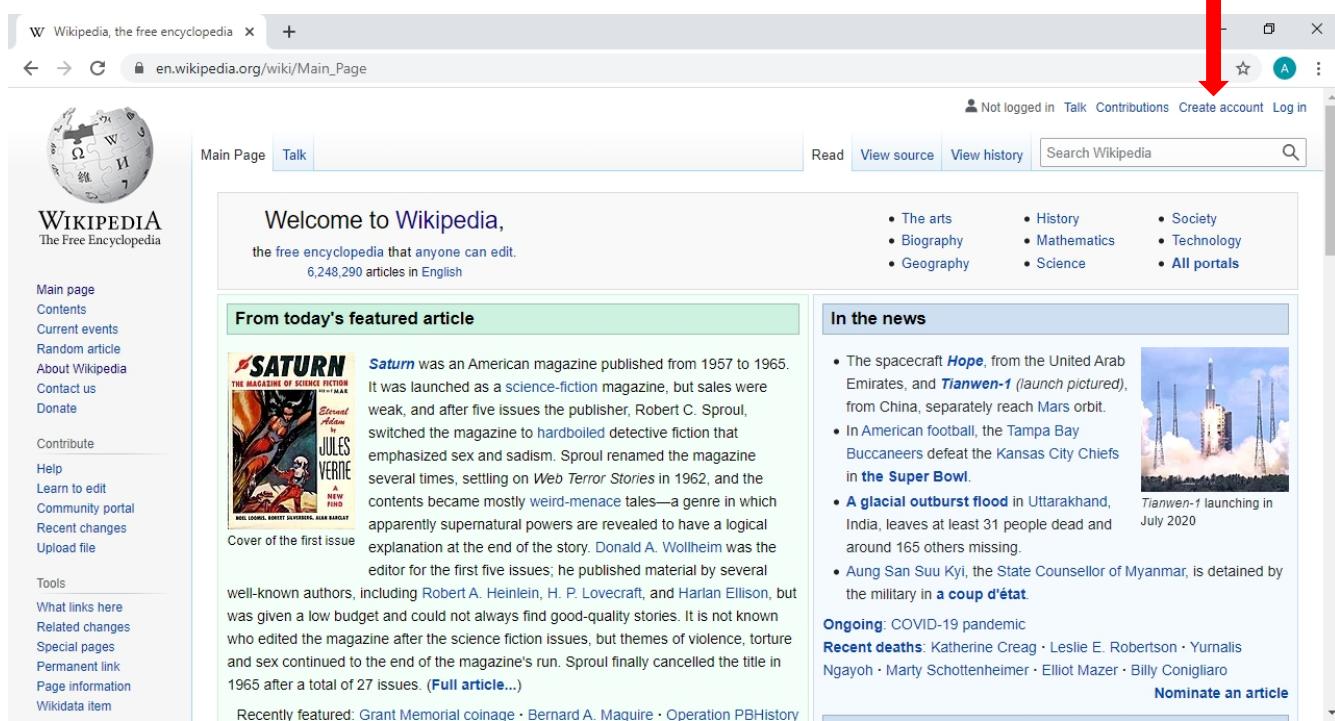
- A lot of information gets added in a short space of time
- You get a wide variety of subjects on the site, some of which are eclectic.
- If someone spots an error, they can quickly change it.
- You get a lot of different angles to a subject .
- People located in different parts of the world can work on the same document.

### b) Creating Account on Wikipedia

Step 1: Go to wikipedia website and select your language.

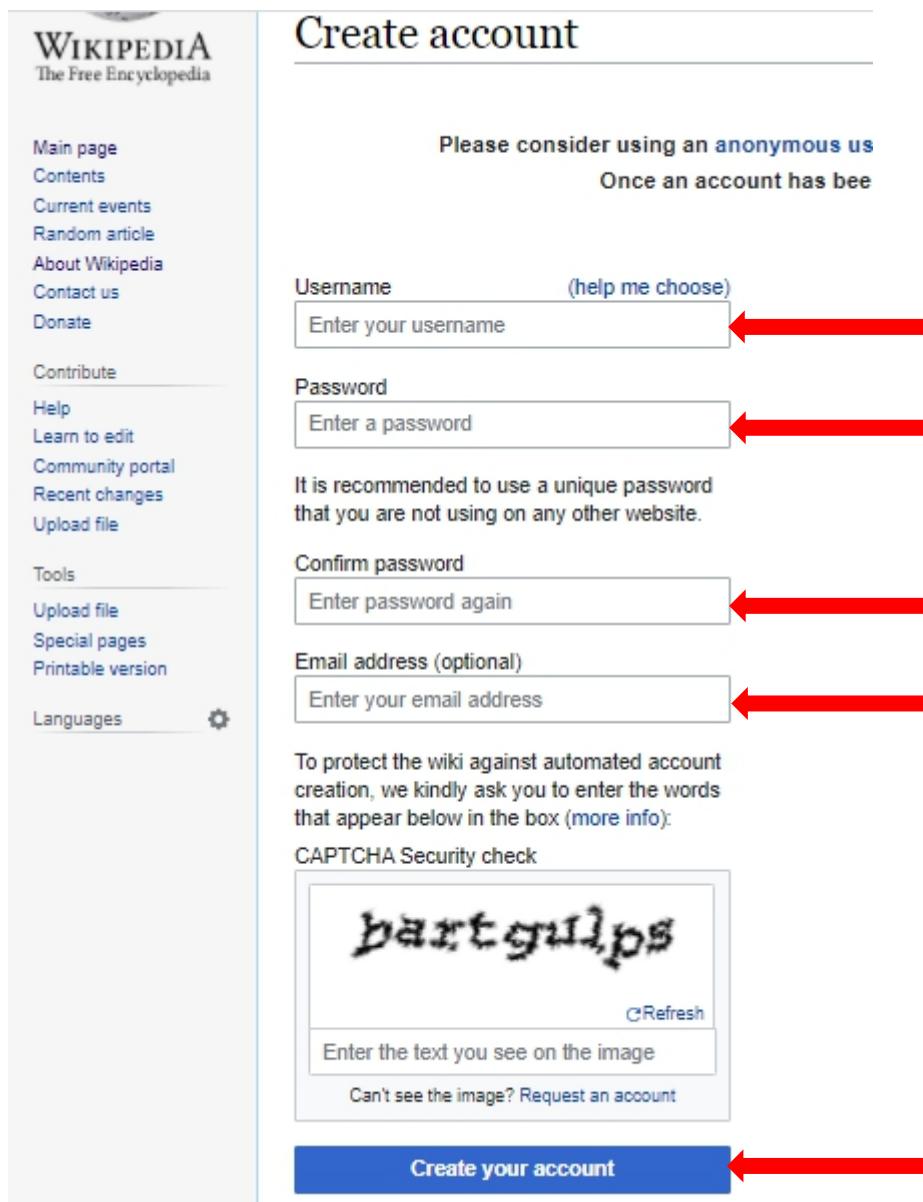


Step 2: Click “Create account” on the upper right side of your browser.



The screenshot shows the Wikipedia Main Page. At the top right, there are links for 'Not logged in', 'Talk', 'Contributions', 'Create account', and 'Log in'. Below these, there's a search bar labeled 'Search Wikipedia' and a magnifying glass icon. On the left sidebar, there's a 'WIKIPEDIA The Free Encyclopedia' logo and a list of navigation links including 'Main page', 'Contents', 'Current events', 'Random article', 'About Wikipedia', 'Contact us', 'Donate', 'Contribute', 'Help', 'Learn to edit', 'Community portal', 'Recent changes', 'Upload file', 'Tools', 'What links here', 'Related changes', 'Special pages', 'Permanent link', 'Page information', and 'Wikidata item'. The main content area features a 'Welcome to Wikipedia' message, a 'From today's featured article' section about 'SATURN THE MAGAZINE OF SCIENCE FICTION' with an image of its cover, and a 'In the news' section with various headlines and images.

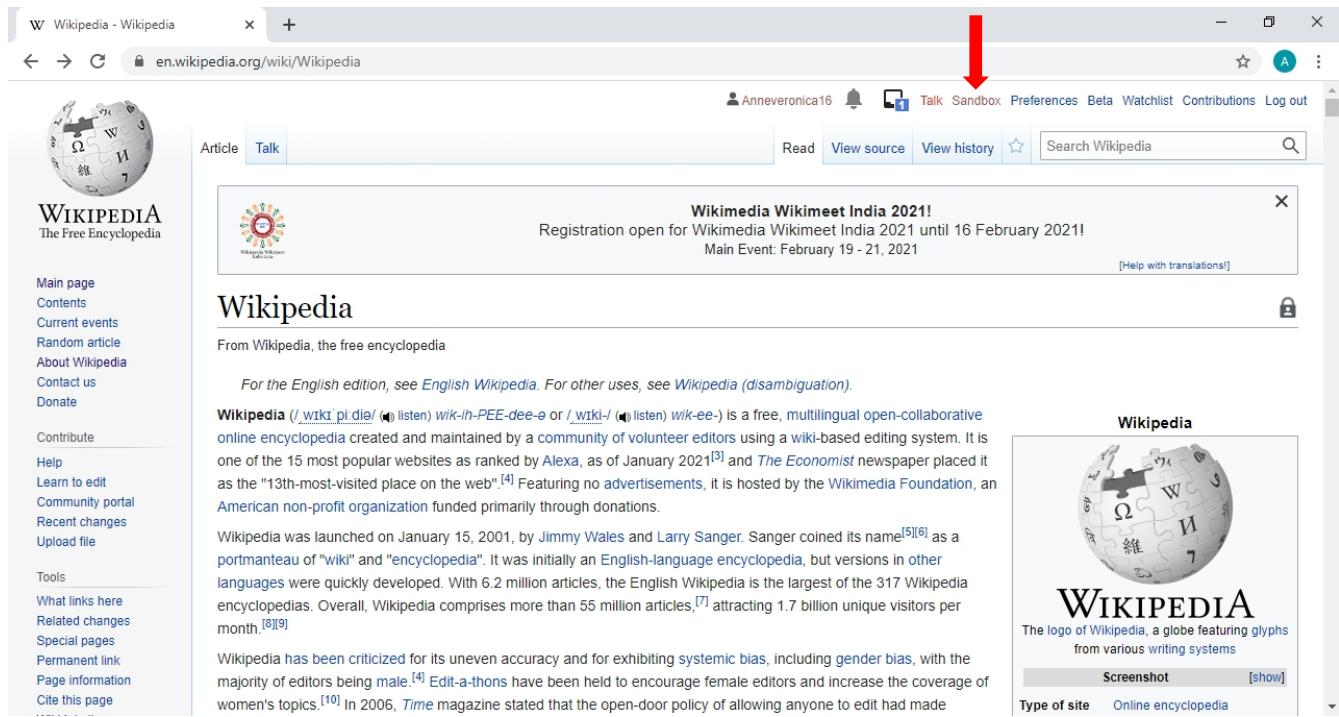
Step 3: Enter your account information and the captcha and click “Create your account”.



The screenshot shows the 'Create account' page. On the left sidebar, there are links for 'Main page', 'Contents', 'Current events', 'Random article', 'About Wikipedia', 'Contact us', 'Donate', 'Contribute', 'Help', 'Learn to edit', 'Community portal', 'Recent changes', 'Upload file', 'Tools', 'Upload file', 'Special pages', 'Printable version', and 'Languages'. The main content area has a heading 'Create account'. It includes a note about using an anonymous user and a warning about account creation. There are four input fields: 'Username' (with placeholder 'Enter your username'), 'Password' (with placeholder 'Enter a password'), 'Confirm password' (with placeholder 'Enter password again'), and 'Email address (optional)' (with placeholder 'Enter your email address'). Below these is a CAPTCHA box containing the text 'bartgulps' and a 'Refresh' button. At the bottom, there's a 'Create your account' button.

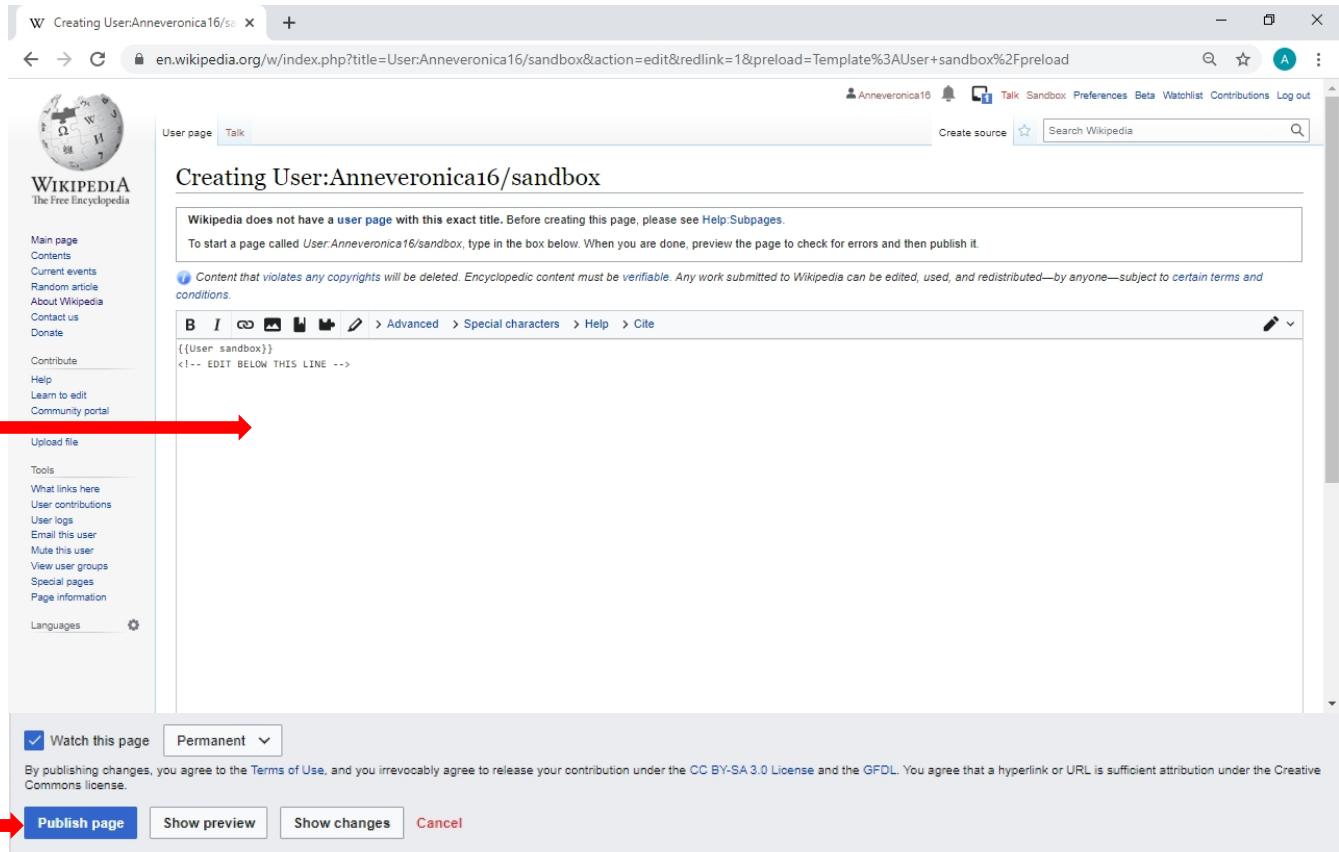
### c) Creating your page on Wikipedia

Step 1: Log in into your wikipedia account and then click on the option “Sandbox”.



The screenshot shows the main Wikipedia homepage. At the top, there is a navigation bar with links for 'Talk', 'Sandbox', 'Preferences', 'Beta', 'Watchlist', 'Contributions', and 'Log out'. A red arrow points to the 'Sandbox' link. On the left side, there is a sidebar with various links like 'Main page', 'Contents', 'Current events', etc. The main content area features a banner for 'Wikimedia Wikimeet India 2021!' and the Wikipedia logo.

Step 2: Type your content inside the text box and then click on “Publish page” and your page is created.



The screenshot shows a user creation page titled 'Creating User:Anneveronica16/sandbox'. The page has a warning message about not having a user page with that title. Below it is a text editor with a red arrow pointing to the 'Edit' button. At the bottom, there is a 'Publish page' button highlighted with a red arrow.

### d) Editing your page on Wikipedia

Step 1: Open your page which you want to edit and click on the option “Edit source”.

W Educational technology - Wikipedia

en.wikipedia.org/wiki/Educational\_technology

Article Talk Read Edit source View history Search Wikipedia

**Educational technology**

From Wikipedia, the free encyclopedia  
(Redirected from Online class)

"Teaching machines" redirects here. For the mechanical devices, see [Teaching machine](#).  
"E-learning" redirects here. It is not to be confused with [Online machine learning](#).

**Educational technology** (commonly abbreviated as **EduTech**, or **EdTech**) is the combined use of computer hardware, software, and educational theory and practice to facilitate learning.<sup>[1]</sup> When referred to with its abbreviation, EdTech, it is often referring to the industry of companies that create educational technology.<sup>[2][3]</sup>

In addition to practical educational experience, educational technology is based on theoretical knowledge from various disciplines such as communication, education, psychology, sociology, artificial intelligence, and computer science.<sup>[4]</sup> [full citation needed] It encompasses several domains including learning theory, computer-based training, online learning, and m-learning, where mobile technologies are used.

Contents [hide]

- 1 Definition
- 2 Related terms
- 3 History
- 4 Theory
  - 4.1 Behaviorism
  - 4.2 Cognitivism
  - 4.3 Constructivism

**Education Disciplines**

- Evaluation · History · Organization ·
- Philosophy · Psychology (school) ·
- Technology** (electronic marking) ·
- International education · School counseling ·
- Special education · Female education ·
- Teacher education · Right to education ·
- Free education

**Curricular domains**

- Arts · Business · Computer science ·
- Early childhood ·
- Engineering · Language · Literacy ·
- Mathematics · Performing arts · Science ·
- Social science · Technology · Vocational

**Methods**

- Case method · Conversation analysis ·
- Discourse analysis · Factor analysis ·
- Factorial experiment · Focus group ·

Step 2: Edit the page and click on “publish changes”.

W Editing Educational technology - en.wikipedia.org/w/index.php?title=Educational\_technology&action=edit

Article Talk Read Edit source View history Search Wikipedia

**Editing Educational technology**

Content that violates any copyrights will be deleted. Encyclopedic content must be verifiable. Any work submitted to Wikipedia can be edited, used, and redistributed—by anyone—subject to certain terms and conditions.

B I Advanced Special characters Help Cite

```
 {{short description|Use of technology in education to improve learning and teaching}}
 {{Redirect|teaching machines|the mechanical devices|teaching machine}}
 {{distinguish|redirect|E-learning|Online machine learning}}
 {{Educational research}}
'''Educational technology''' (commonly abbreviated as '''EduTech''' or '''EdTech''') is the combined use of computer hardware, software, and educational theory and practice to facilitate [[learning]].<ref>{{cite
web|url=http://www.aect.org/publications/EducationalTechnology/ER5861X_C002.pdf|title=Facilitating Learning
|last1=Robinson|first1=Rhonda|last2=Molenda|first2=Michael |website=Association for Educational Communications and Technology}}</ref>
<ref>{{cite
web|url=https://www.wsj.com/articles/whats-better-in-the-classroom-teacher-or-machine-1485772201|title=What's Better in the Classroom-Teacher or Machine?|website=Wall Street Journal}}</ref><ref>{{cite
web|url=https://techcrunch.com/2020/12/22/to-win-post-pandemic-edtech-needs-to-start-thinking-big|title=To win post-pandemic, edtech needs to start thinking big|website=TechCrunch}}</ref>
```

In addition to practical educational experience, educational technology is based on theoretical knowledge from various disciplines such as communication, education, psychology, sociology, artificial intelligence, and computer science.<ref>Draft National Education Policy 2019 by Govt. of India (2018)</ref>{{Full citation needed|date=November 2020}} It encompasses several domains including [[Learning theory (education)|learning theory]], computer-based training, online learning, and [[m-learning]], where mobile technologies are used.

Edit summary (Briefly describe your changes)

This is a minor edit  Watch this page Permanent

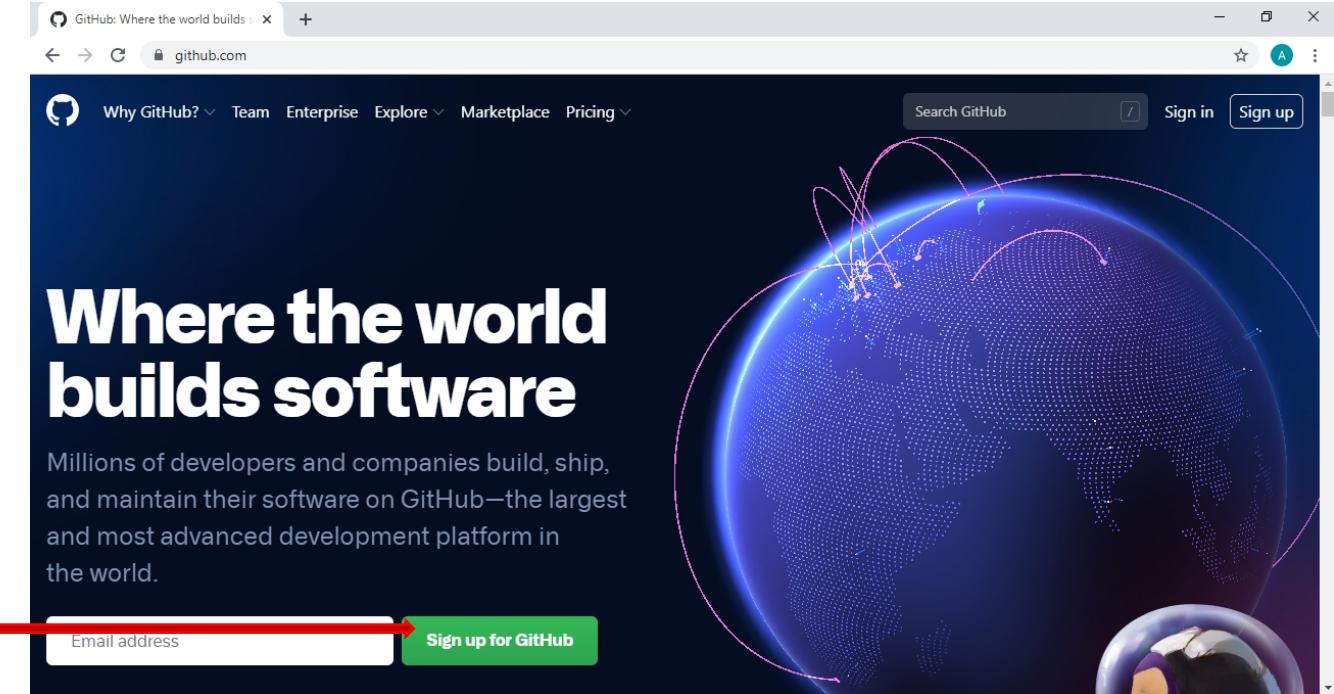
By publishing changes, you agree to the [Terms of Use](#), and you irrevocably agree to release your contribution under the [CC BY-SA 3.0 License](#) and the [GFDL](#). You agree that a hyperlink or URL is sufficient attribution under the Creative Commons license.

Publish changes Show preview Show changes Cancel

## Practical 2: Creating account, repository on Github and cloning repository in Github

### a) Creating Account

**Step 1:** Go to Github website and click on the “sign up” button.



**Step 2:** “Create your account” page will appear. Enter your username, email address and password. Then click on the “verify” button. Once it is verified then click on the “Create Account” button and your account is created.

# Create your account

Username \*

Email address \*

Password \*

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

Email preferences

Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we know you are a real person

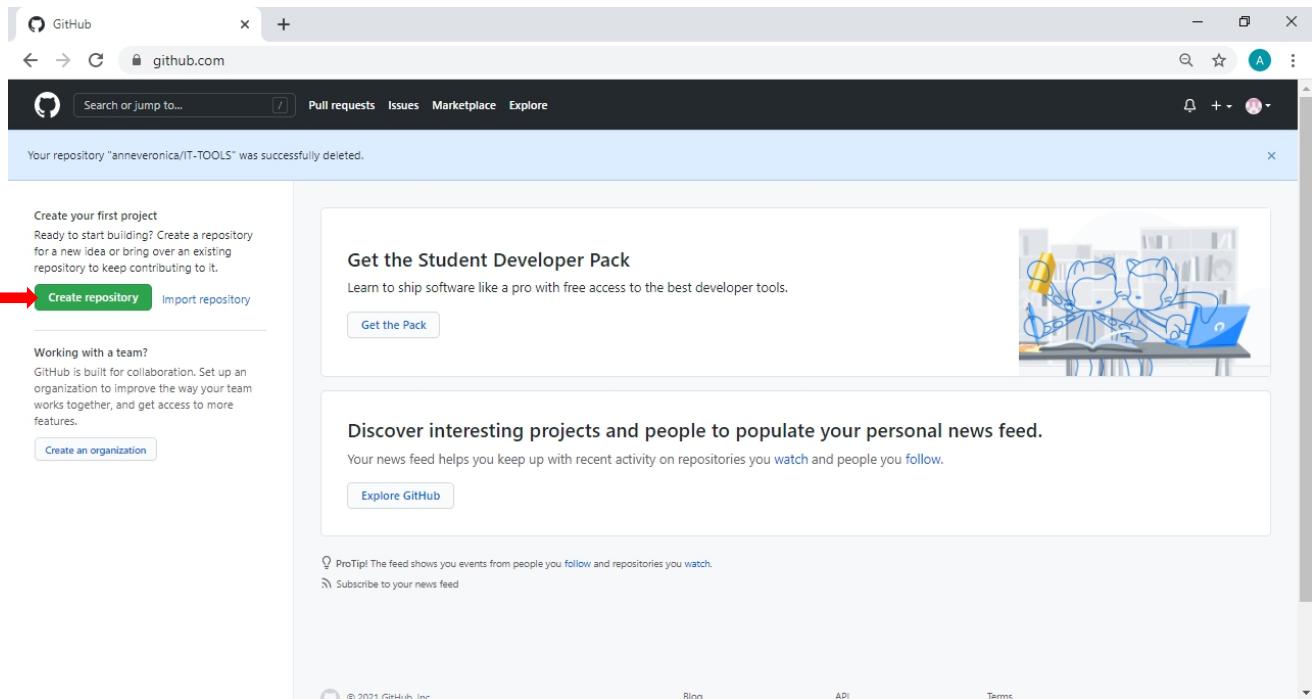
  
 





## b) Creating Repository

**Step 1:** Click on the “Create repository” button which is in the left side of the page.



**Step 2:** Once you click on the create repository button, a page is displayed. Provide your repository name and you can give some description about your repository. Then choose a repository visibility option and click on “Create repository” option.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner \* Repository name \*

anneveronica / IT TOOLS ✓

Great repository names are short and about friendly-octo-funicular? Your new repository will be created as IT-TOOLS. About friendly-octo-funicular?

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](#).

Choose a license A license tells others what they can and can't do with your code. [Learn more](#).

This will set `main` as the default branch. Change the default name in your [settings](#).

**Create repository**

### c) Cloning Repository

**Step 1:** In the main page of the repository, click on the option “Code” and then click on the copy symbol and your repository has been cloned.

The screenshot shows a GitHub repository page for 'anneveronica/IT-TOOLS'. At the top, there are tabs for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the tabs, there's a search bar and a navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A red arrow points to the 'Code' button in the navigation bar. The main content area shows a file list with 'Practical1.pdf' and 'README.md'. On the right side, there are sections for 'No description, website, or topics provided.', 'Readme', 'Releases', and 'Packages'. At the bottom, there's a footer with links for 'About', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

This screenshot is identical to the one above, but the 'Code' dropdown menu is now expanded. It contains options for 'Clone' (with 'HTTPS', 'SSH', and 'GitHub CLI' choices), a copy icon (highlighted with a red arrow), 'Readme', 'Open with GitHub Desktop', and 'Download ZIP'. The rest of the page content, including the file list and sidebar sections, remains the same.

## **PRACTICAL 3: BASIC UNDERSTANDING ON FREE AND OPEN-SOURCE SOFTWARE**

### A) Describe Open Source Software with Example.

Open source software is software with source code that anyone can inspect, modify, and enhance. 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 the design and functionality of the overall project.

Some widely used programs, platforms, and languages which are considered open source are:

- Linux operating system
- Android by Google
- Open office
- Firefox browser
- VCL media player
- Moodle

### B) Describe Free Software with Example.

Freeware is software that you don't have to pay for. Free software is software you are free to modify and use for your own purposes. Freeware does not necessarily have to be free software as the source code can still be protected. "Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software.

Free Source Software Example:

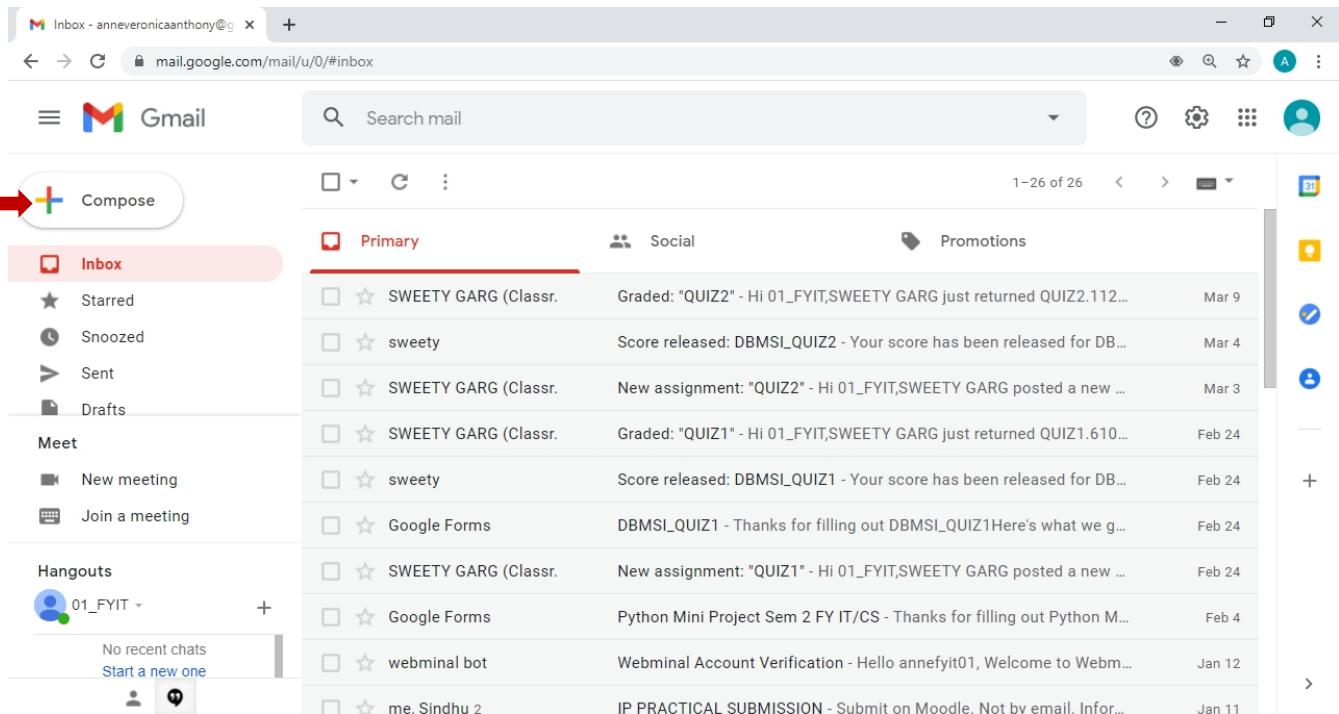
- 1.Linux
- 2.Apache
- 3.Google Chrome
- 4.Internet Explorer
- 5.Whatsapp

### C) Difference between Free and Open Source Software.

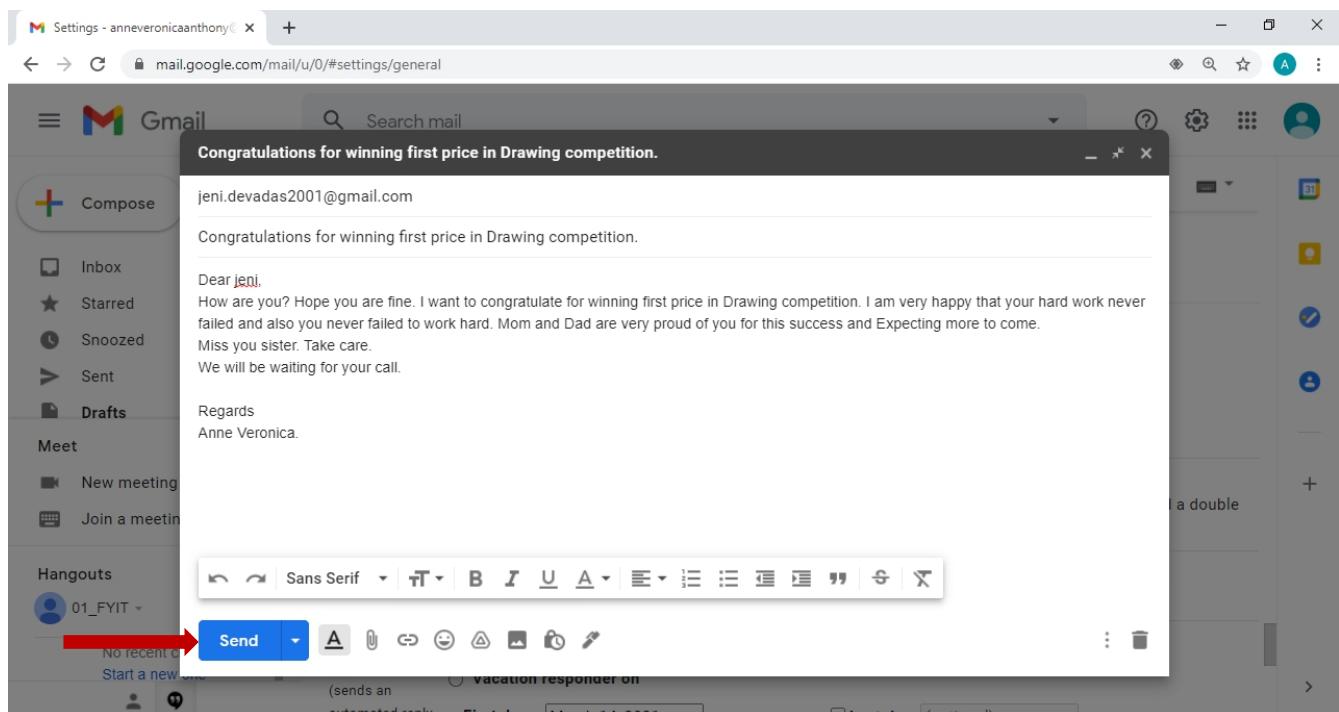
- 1.Free Software is Open Source Software, but Open Source software may not necessarily be Free Software.
2. Open Source allows the coder more control over his program compared to Free Software.
3. Open Source software can work with other proprietary software, while Free Software does not allow the same.

## Practical 4: Writing EMAIL

**Step 1:** Open your Gmail and click on “Compose”.



**Step 2:** Write your Email and click on “Send” option.



## **Practical 5**

**Using practical examples, describe green computing. List and explain the steps that you take to contribute to green computing.**

Green computing is the environmentally responsible and eco-friendly use of computers and their resources. In broader terms, it is also defined as the study of designing, engineering, manufacturing, using and disposing of computing devices in a way that reduces their environmental impact.

To promote green computing concepts at all possible levels, the following four approaches are employed:

- Green use: Minimizing the electricity consumption of computers and their peripheral devices and using them in an eco-friendly manner.
- Green disposal: Repurposing existing equipment or appropriately disposing of, or recycling, unwanted electronic equipment.
- Green design: Designing energy-efficient computers, servers, printers, projectors and other digital devices.
- Green manufacturing: Minimizing waste during the manufacturing of computers and other subsystems to reduce the environmental impact of these activities.

Steps to contribute to 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. This can be done even if the computer is working on something. Screensavers do not save power. Same goes for computers , you don't have to shut it down completely if you don't want to reboot, just use sleep or hibernation mode. This will help save energy and keep the system to its current state when you need it again.
- 2) Use the power saving features: All computers include power saving options. Using these features you can command the computer to do various energy-saving tasks automatically, including shutting off unused hard disks, powering off a monitor after a given time or even placing the computer into sleep mode when not in use. This is very useful on laptops to help preserve battery life.
- 3) Purchase energy saving hardware: If you don't need super-fast computing power then look out for energy efficient components when buying a new computer, such as green hard drives and low-energy processors. While performance is slower they can use remarkably less power. Purchasing an energy saving power supply unit for a desktop PC can help the environment and save money, they're often quieter too.
- 4) Use a laptop instead of desktop: Laptops are much better for the environment than desktop computers as they have components which require less power. If you don't need a desktop computer consider buying a laptop instead, or if you have both use the laptop as much as possible before considering the desktop.
- 5) 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. There should be companies which can remove the metals which may fix or furnish items. You should check with your local authorities to find out what facilities they offer for safe disposal of old computing parts.

## Practical 6 : Writing Blogs

**ME and DANCE**

- March 22, 2021



**There are shortcuts to HAPPINESS and DANCING is one of them.**

If someone asks me what's the thing which you Love the most to do then definitely my answer is DANCE 🎉. I've never thought that I will have so much craze and love ❤️ for Dance. But Dance has always been the greatest Happiness for me. I have participated in most of the dance events and have received compliments for my Dance. The thing which I like about my dance is my energy level and when that has become the compliment for me, I was overjoyed 😊. I feel very happy when I get compliments for my dance compared to other compliments. Through Dance I've learned Teamwork, co-ordination and many more things and also Dance is a good way to express one's emotions.

The Dance performance which I've performed with my college mates in the annual day during my junior college is my favourite and memorable performance till now. During rehearsals for that dance, I had a great time with my Friends. We enjoyed a lot by chit chatting 😊, singing songs 🎵, playing Damsharas and so on. Also we got a chance to skip some of the boring lectures 😊😊. So all that memories made the day to be cherished forever ❤️.

Now, I remember my favourite quote by Dr. APJ Abdul Kalam which says...

*"Sometimes, it's better to bunk a class and enjoy with friends because now when I look back marks never make me laugh but Memories do".*

I feel very Happy, relaxed when I Dance. I wish I could dance even more better than now I am. So, I'll end up by my favourite quote which says...

*"It is with your feet that you move...  
but it is with your Heart that you Dance."*

Link of the blog: <https://anneveronicaantony.blogspot.com/>

## Practical 7: Implementing coding practices in python using PEP8

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

5

```
x=10  
if x>5 and x%2==0:  
    print('x is larger than 5 and divisible by 2!')
```

x is larger than 5 and divisible by 2!

```
x = 5 # This is an inline comment  
if x%2==0:  
    print(x,"is even number")  
else:  
    print(x,"is odd number")
```

5 is odd number

# PEP 8 STYLE GUIDE

## GROUP No. 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 4(): #Function name is too generic and it can create a confusion in enterprise programming
    print("number name is too generic, you can use it but it is not recommended as it do not tell explanatory and descriptive")
4()

Function Name is too generic, you can use it but it is not recommended as it do not tell explanatory and descriptive
```

```
#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
```

```
The very first line for class has "1class" instead of "class"
Name Should not start with a number
Name should be intuitive and not too common
```

```
class Product:
    def __init__(self):
        self.product_id = input("Enter the product id: ")
        self.product_name = input("Enter the product Name: ")
        self.product_qty = int(input("Enter the quantity of the product: "))
        self.product_price = float(input("Enter the unit price: "))

    def display(self):
        print("Product Id: ", self.product_id, "Product Name: ", self.product_name, "Product Quantity: ", self.product_qty, "Product Price: ", self.product_price)

    def calculate_total(self):
        total = self.product_qty * self.product_price
        print("Total Price of the product is: ", total)

    def calculate_discount(self, discount_percent):
        discount = (self.product_qty * self.product_price) * (discount_percent / 100)
        print("Discount amount is: ", discount)

    def calculate_final_price(self):
        final_price = self.product_qty * self.product_price - discount
        print("Final Price after discount is: ", final_price)
```

**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()
```

```
# 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)
```

Y  
Operated on.

able is being added or subtracted, as the operator is right next to the variable being

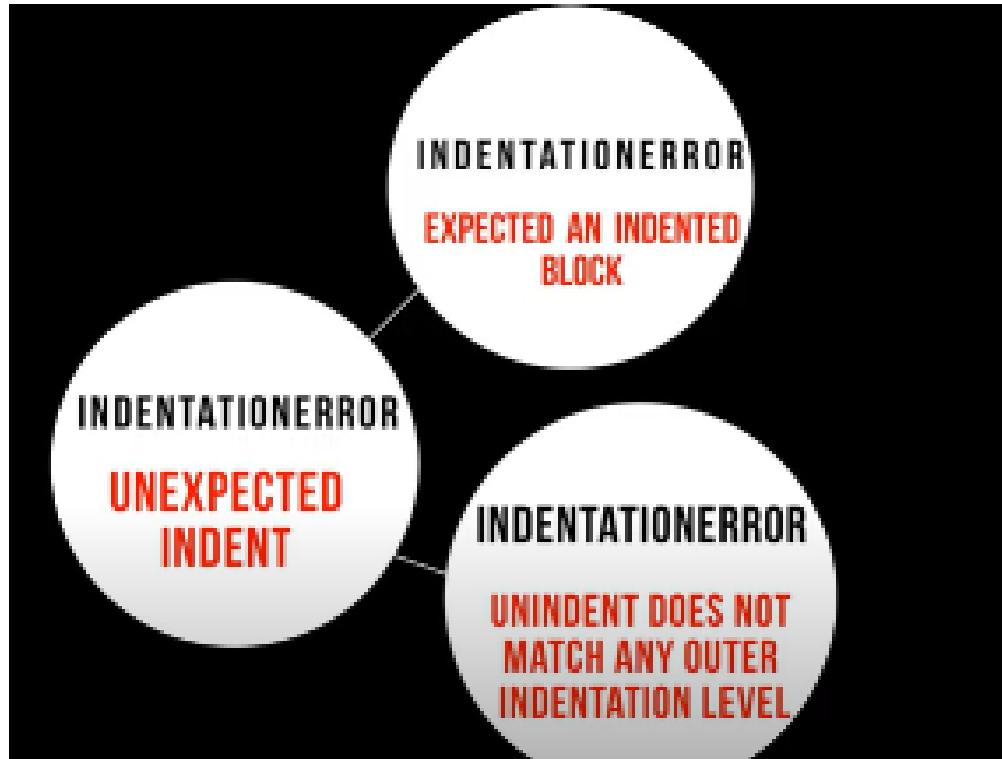
In the below Example

#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)
```



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

# Not Recommended

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

### 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
```

# 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

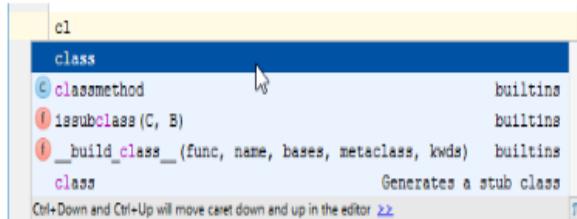
ANSWER: NEVER

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

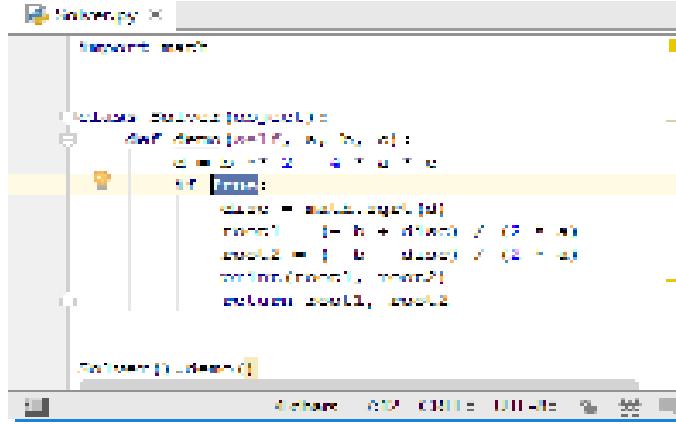
Highlighting code style violations:



(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:



## 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 tools 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.

# Recommended

Python

- Ex-

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

# Not Recommended

Python

- Ex-

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

# Should a line break Before or After A Binary Operator.

Python

- Ex-

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

Python

- Ex-

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

Example of comments.

## block comment.

Anti-partten.

### Example

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

Best-pratice.

### 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
```

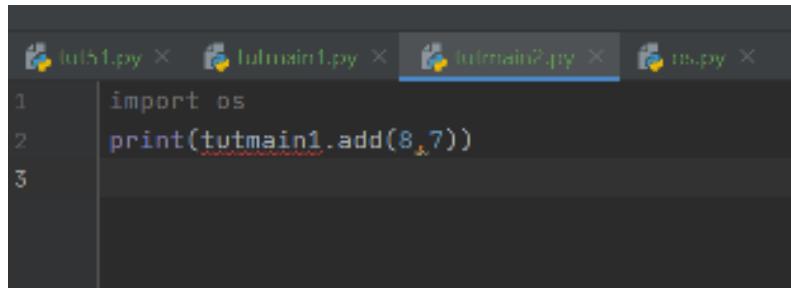
## Documentation string comment.

```
"""Return a foobang  
  
Optional plotz says to  
frobinate the bizbaz first.  
"""
```

# EXAMPLE OF NAMING CONVENTION

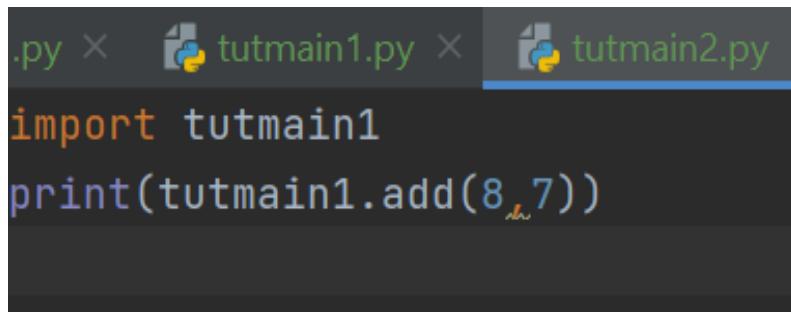
## NAMING MODULE WITH HELP OF PEP8

# Not recommended



```
tut51.py × tutmain1.py × tutmain2.py × ms.py ×  
1 import os  
2 print(tutmain1.add(8, 7))  
3
```

# Recommended



```
.py × tutmain1.py × tutmain2.py × ms.py ×  
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'
```

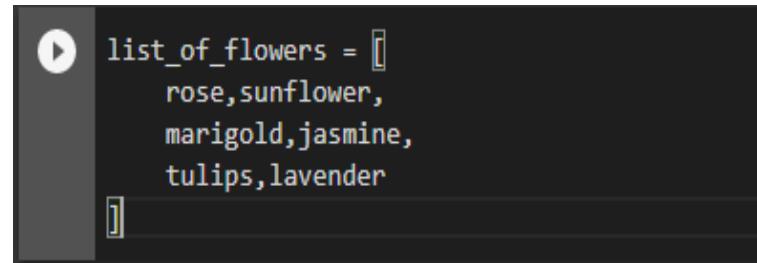
>>>

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

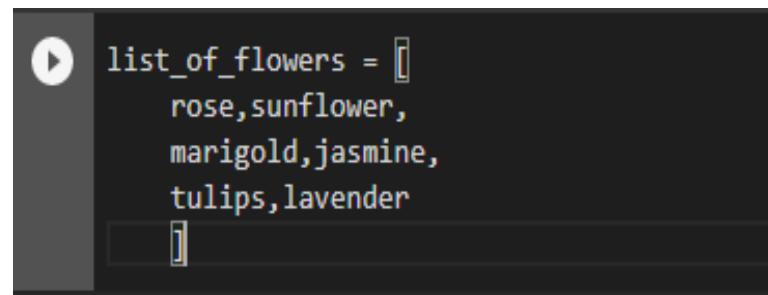
## ❖ EXAMPLES OF INDENTATION

### ❖ METHODS OF WHERE TO PUT CLOSING BRACES:-

# Recommended



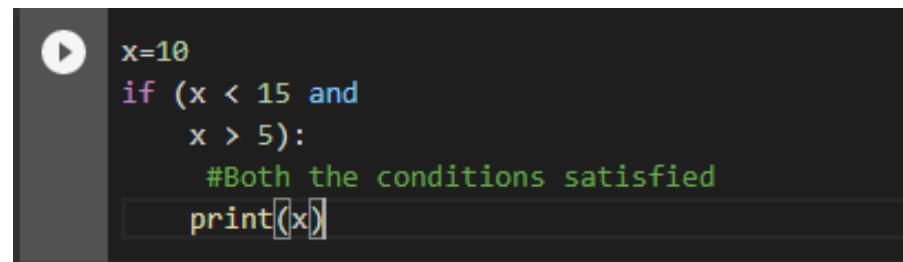
```
list_of_flowers = [rose,sunflower,  
                   marigold,jasmine,  
                   tulips,lavender]  
[]
```



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

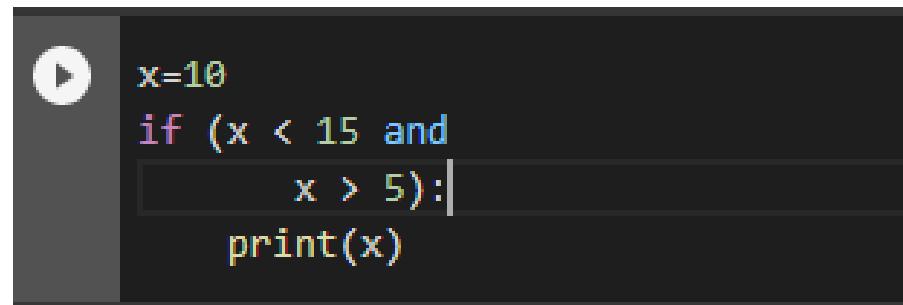
## ❖ Methods for following line breaks

### # Recommended



A screenshot of a code editor window. On the left is a dark sidebar with a play button icon. The main area contains Python code:

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



A screenshot of a code editor window. On the left is a dark sidebar with a play button icon. The main area contains Python code, identical to the one above but with different line breaks:

```
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!')
```

# THANK YOU