PYTHON BASIC

Git & GitHub Setup

Step 1: Install Git

- Download Git: https://git-scm.com/downloads
- Check installation:

git --version

Step 2: Configure Git

git config --global user.name "Your Name"

git config --global user.email "youremail@example.com"

Step 3: Create a GitHub Account

- Sign up at https://github.com
- Create a new repository (e.g., python-course)

Step 4: Clone Repository to Local Machine

git clone https://github.com/your-username/python-course.git

Step 5: Git Commands You Must Know

git status # Check status of repo

git add filename.py # Stage a file

git add . # Stage all files

git commit -m "First commit" # Commit changes

git push origin main # Push to GitHub

git pull origin main # Pull latest changes

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

Git Commands Explained

1. Setup & Configuration

git config --global user.name "Your Name"

git config --global user.email "your@email.com"

✅ Sets your identity (needed for commits).

Check settings:

git config --list

---

2. Initialize a Repository

git init

✅ Creates a new Git repository in your project folder (starts tracking files).

---

3. Clone a Repository

git clone https://github.com/username/repo.git

✅ Copies a repository from GitHub to your local computer.

---

4. Check Status

git status

✅ Shows the state of files (modified, staged, or untracked).

---

5. Add Files to Staging Area

git add filename.py # Add a specific file

git add . # Add all changes

✅ Prepares files to be committed.

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

---

6. Commit Changes

git commit -m "Write a descriptive message"

✅ Saves a snapshot of your changes into Git history.

---

7. View Commit History

git log

✅ Shows past commits with author, date, and messages.

---

8. Push to Remote Repository

git push origin main

✅ Uploads your commits to GitHub (replace main with branch name).

---

9. Pull Latest Changes

git pull origin main

✅ Downloads updates from GitHub and merges them into your local branch.

---

10. Branching

git branch # List branches

git branch new-feature # Create a new branch

git checkout new-feature # Switch to branch

git checkout -b dev # Create and switch to "dev"

✅ Branches let you work on features separately without affecting the main code.

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

11. Merging

git checkout main

git merge new-feature

✅ Combines changes from another branch into the current branch.

---

12. Remove Files

git rm file.txt

✅ Removes a file from repo (staging + working directory).

---

13. Reset Changes

git reset filename.py # Unstage a file

git reset --hard # Remove all changes (CAUTION)

✅ Restores previous states of files/commits.

---

14. Check Differences

git diff

✅ Shows differences between working directory and staging area.

---

15. Remote Management

git remote -v

✅ Shows connected remote repositories (like GitHub).

---

16. Stashing

git stash # Save changes temporarily

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

git stash pop # Apply stashed changes back

✅ Useful if you want to switch branches but keep your uncommitted work safe.

📝 Quick Git Workflow (Daily Use)

1. git pull origin main → Get latest code from GitHub
2. git status → Check changes
3. git add . → Stage all files
4. git commit -m "Message" → Save snapshot
5. git push origin main → Upload to GitHub

Module 2: Python Setup

Step 1: Install Python

- Download from https://www.python.org/downloads/
- Verify installation:

python --version

Step 2: Install pip (Python package manager)

python -m ensurepip --upgrade

Step 3: Create Virtual Environment

python -m venv venv

Activate:

- Windows: venv\Scripts\activate
- Mac/Linux: source venv/bin/activate

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

Module 3: IDEs for Python

- VS Code (Recommended) → Install VS Code and Python extension
- PyCharm (For larger projects)
- Jupyter Notebook (For data science)

Module 4: Python Basics – Variables & Data Types

Step 1: Variables in Python

```
# A variable stores data
name = "Alice"
age = 25
height = 5.7
is_student = True

print(name, age, height, is_student)
```

🔷 **1. List Methods**

```
# Creating a list
fruits = ["apple", "banana", "cherry"]

# append() → add item
fruits.append("orange")
print(fruits)   # ['apple', 'banana', 'cherry', 'orange']

# insert() → insert at index
```


+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

```
fruits.insert(1, "grape")
print(fruits)   # ['apple', 'grape', 'banana', 'cherry', 'orange']

# remove() → remove value
fruits.remove("banana")
print(fruits)   # ['apple', 'grape', 'cherry', 'orange']

# pop() → remove by index (default last)
fruits.pop()
print(fruits)   # ['apple', 'grape', 'cherry']

# sort() → sort list
fruits.sort()
print(fruits)   # ['apple', 'cherry', 'grape']

# reverse() → reverse list
fruits.reverse()
print(fruits)   # ['grape', 'cherry', 'apple']

# count() → count occurrences
print(fruits.count("apple"))  # 1

# index() → find index of value
print(fruits.index("cherry"))  # 1

# extend() → add another list
```


+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

```
fruits.extend(["kiwi", "melon"])
print(fruits)   # ['grape', 'cherry', 'apple', 'kiwi', 'melon']

# clear() → remove all items
fruits.clear()
print(fruits)   # []
```

🔷 2. String Methods

```
text = "  Hello World  "

# upper() → convert to uppercase
print(text.upper())  # "  HELLO WORLD  "

# lower() → convert to lowercase
print(text.lower())  # "  hello world  "

# strip() → remove spaces
print(text.strip())  # "Hello World"

# replace() → replace substring
print(text.replace("World", "Python"))  # "  Hello Python  "

# split() → split into list
print(text.split())  # ['Hello', 'World']
```

```
# join() → join list into string
words = ["Python", "is", "fun"]
print(" ".join(words))  # "Python is fun"

# find() → return index of first occurrence
print(text.find("World"))  # 8

# count() → count occurrences
print(text.count("l"))  # 3

# startswith() → check start
print(text.strip().startswith("Hello"))  # True

# endswith() → check end
print(text.strip().endswith("World"))  # True
```

🔷 3. Tuple Methods

Tuples are **immutable**, so only two main methods exist:

```
numbers = (1, 2, 3, 2, 4, 2)

# count() → count occurrences
print(numbers.count(2))  # 3

# index() → find index of value
print(numbers.index(3))  # 2
```


+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

🔷 4. Dictionary Methods

```
student = {"name": "Alice", "age": 20, "grade": "A"}


# keys() → get all keys

print(student.keys())  # dict_keys(['name', 'age', 'grade'])


# values() → get all values

print(student.values())  # dict_values(['Alice', 20, 'A'])


# items() → get key-value pairs

print(student.items())  # dict_items([('name', 'Alice'), ('age', 20), ('grade', 'A')])


# get() → get value safely

print(student.get("age"))  # 20


# update() → update dict

student.update({"age": 21, "subject": "Math"})

print(student)  # {'name': 'Alice', 'age': 21, 'grade': 'A', 'subject': 'Math'}


# pop() → remove key

student.pop("grade")

print(student)  # {'name': 'Alice', 'age': 21, 'subject': 'Math'}


# popitem() → remove last key-value
```


+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

```
student.popitem()

print(student)  # {'name': 'Alice', 'age': 21}


# clear() → empty dict

student.clear()

print(student)  # {}
```

🔷 5. Integer Methods

⚠️ **Integers don't have built-in methods** like lists or strings (since they are immutable and simple).
But Python provides **functions** for integers through int and math operations.

```
# Basic integer operations

x = 10

y = -25


print(abs(y))       # 25 (absolute value)

print(pow(x, 2))    # 100 (power function)

print(divmod(10, 3)) # (3, 1) → quotient and remainder

print(bin(5))       # '0b101' (binary representation)

print(hex(255))     # '0xff' (hexadecimal representation)

print(oct(10))      # '0o12' (octal representation)


# Type casting

print(int("20"))    # 20 (string to int)

print(float(5))     # 5.0 (int to float)
```


+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

🎓 Classwork Assignments

Git & GitHub (1–5)

1. Install Git and check the version on your system. Submit a screenshot.
2. Configure your Git username and email, then check with git config --list.
3. Create a new GitHub repository named my-first-repo.
4. Clone your GitHub repository to your computer. Add a file called hello.txt with any text, commit, and push it.
5. Make a change to hello.txt, commit it, and push again. Then use git pull to sync your local repo.

Python Setup & IDEs (6–8)

6. Install Python and show the output of python --version.
7. Create and activate a virtual environment (venv) and install a package like requests.
8. Install VS Code, open a Python file, and run a simple “Hello, World!” script.

Variables & Data Types (9–20)

9. Create a Python script with variables for your name, age, and whether you’re a student. Print them.
10. Write a program to store and print the sum of two integers.
11. Create two float variables and print their product.
12. Store your favorite color in a string variable and print it in uppercase.
13. Create a list of five fruits and print the second fruit.
14. Create a tuple with your birth year, birth month, and birth day. Print them individually.
15. Make a dictionary that stores your name, age, and country. Print each key-value pair.
16. Write a program to add two numbers entered by the user (using input()).

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

17. Create a set of numbers {1, 2, 3, 4, 4, 5} and print the unique values.
18. Create variables of type int, float, str, bool, list, tuple, dict, and set. Print their data types using type().
19. Write a program that swaps the values of two variables.
20. Create a list of your three favorite movies and use indexing to print the last one.

📝 **Quiz**

Part A: Multiple Choice (5 questions)

1. Which command initializes a new Git repository?
    - a) git init
    - b) git start
    - c) git create
    - d) git begin
2. Which Python data type is immutable?
    - a) list
    - b) tuple
    - c) dict
    - d) set
3. Which IDE is recommended for beginners in Python?
    - a) Notepad
    - b) VS Code
    - c) MS Word
    - d) Photoshop
4. What does git add . do?
    - a) Adds all files to staging area
    - b) Adds only Python files

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud

- c) Creates a commit
- d) Deletes all files

5. What is the result of print(type(3.14)) in Python?
    - a) <class 'int'>
    - b) <class 'float'>
    - c) <class 'str'>
    - d) <class 'bool'>

Part B: Short Answer (5 questions)

6. What is the difference between git commit and git push?
7. How do you create and activate a virtual environment in Python?
8. What is the difference between a list and a tuple?
9. Give an example of a Python dictionary with at least two key-value pairs.
10. Explain what a Boolean variable represents in Python.

Part C: Coding (5 questions)

11. Write a program that asks the user for their name and prints: Hello, <name>!
12. Write a program that creates a list of 5 numbers and prints their sum.
13. Write a program that stores your school subjects in a tuple and prints the first subject.
14. Write a program that creates a dictionary with "name": "John", "age": 20 and prints the age.
15. Write a program that takes two numbers from the user and prints their product.

+2348037137902, developers@applinet.com.ng, genius@codehouse.cloud
www.applinet.com.ng, www.codehouse.cloud