

# Python argparse — Beginner Notes

---

## What is argparse?

- A **tool in Python** to accept inputs from the command line.
  - Instead of asking with `input()`, you pass values when running the script.
  - Makes scripts more flexible → great for **command-line utilities**.
- 

## Example Without argparse

```
name = input("Enter your name: ")  
print(f"Hello, {name}")
```

Run:

```
python script.py
```

👉 You must type inside the program.

---

## Example With argparse

```
import argparse  
  
parser = argparse.ArgumentParser()  
parser.add_argument("name", help="Your name")  
args = parser.parse_args()  
  
print(f"Hello, {args.name}")
```

Run:

```
python script.py Zahid
```

Output:

Hello, Zahid

👉 You pass input **directly** in the command.

---

## Why use argparse?

👉 No need to edit code. 👉 Inputs are passed at runtime. 👉 Perfect for making mini tools.

---

## Positional vs Optional Arguments

### 1. Positional

- Must be given (no `-` or `--`).
- Example:

```
parser.add_argument("name", help="Your name")
```

Run:

```
python script.py Zahid
```

👉 Missing it = error.

### 2. Optional

- Start with `-` (short) or `--` (long).
- Example:

```
parser.add_argument("-n", "--name", help="Your name")  
parser.add_argument("-g", "--greet", action="store_true", help="Say hello")
```

Run:

```
python script.py -n Zahid --greet
```

Output:

```
Hello, Zahid
```

---



- `-n` → short option (1 letter, quick)
- `--name` → long option (full word, clear)
- You can define **both** for flexibility.

Example:

```
parser.add_argument("-n", "--name", help="Your name")
```

Run:

```
python script.py -n Zahid  
python script.py --name Zahid
```

👉 Both work the same.

## Common Parameters in `add_argument()`

### 1. Name of Argument

```
parser.add_argument("filename")
```

👉 Positional (must give).

### 2. Short & Long Options

```
parser.add_argument("-n", "--name")
```

👉 `-n` = short, `--name` = long.

### 3. help

```
parser.add_argument("--age", help="Enter your age")
```

👉 Shown in `--help`.

### 4. type

```
parser.add_argument("--age", type=int)
```

👉 Converts to integer.

## 5. default

```
parser.add_argument("--city", default="Istanbul")
```

👉 If missing → use default.

## 6. required

```
parser.add_argument("--email", required=True)
```

👉 Must be given.

## 7. action

```
parser.add_argument("--greet", action="store_true")
```

👉 Flag (True if used, False otherwise).

---

## 🎯 Quick Recap

- **Positional** → no `-` or `--`, must provide.
- **Optional** → starts with `-` / `--`, flexible.
- **help, type, default, required, action** → useful extras.

👉 argparse = turn Python scripts into real **command-line tools** 😭