

BACKEND WEB DEVELOPMENT

with

PYTHON & DJANGO

Workshop Outline

DAY 1:

Python Recap
Django Basics

DAY 2:

Project – News articles website
Django Models
Django Admin Site

DAY 3:

Project – Image Compression Website
Future Topics?

PYTHON

Why Python?

- **Simple and Readable Syntax:** has clean and easy-to-read syntax, making it beginner-friendly.
- **Interpreted Language:** is executed line-by-line, which helps with easy debugging.
- **Dynamic Typing:** No need to declare variable types explicitly; Python determines the type at runtime.
- **High-Level Language:** Abstracts complex details of the hardware, letting developers focus on writing logic.
- **Versatile Applications:** Used in web development, data science, machine learning, automation, and more.
- **Cross-Platform Compatibility:** runs on various platforms, including Windows, macOS, and Linux.
- **Extensive Libraries and Frameworks:** It has a vast collection of standard libraries and third-party packages, like NumPy, Pandas, and TensorFlow.

SYNTAX

Types

a = 28	int
b = 1.5	float
c = "Hello!"	str
d = True	bool
e = None	NoneType

Conditions

```
if x > 0:
    print("x is positive")
elif x < 0:
    print("x is negative")
else:
    print("x is 0")
```

Relational
Operators

< , >

<= , >=

==

!=

OR - AND

```
if x < y or x > y:  
    print("x is not equal to y")
```

```
if score >= 90 and score <= 100:  
    print("Grade: A")
```

OR

X	Y	X or Y
F	F	F
F	T	T
T	F	T
T	T	T

AND

X	Y	X and Y
F	F	F
F	T	F
T	F	F
T	T	T

Input / Output

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

Data Structures

list - sequence of mutable values

tuple - sequence of immutable values

set - collection of unique values

dict - collection of key-value pairs

...

Loops

```
print("hello")  
print("hello")  
print("hello")
```

while

```
i = 1  
while i <= 3:  
    print("hello")  
    i = i + 1
```

for

```
for i in range(3):  
    print("hello")
```

```
cartoons = ["Tom", "Jerry", "Spike"]  
for cartoon in cartoons:  
    print(cartoon)
```

Functions

```
def hello(name):  
    print(f"hello {name}")  
  
# this is a comment in Python  
n = input("What's your name? ")  
hello(n)
```

Exception Handling

What if the user enters wrong input?

```
try:  
    x = int(input("What's x?"))  
    y = 100 / x  
except ZeroDivisionError:  
    print("can't divide by zero")
```

Libraries

use pre-defined functions and data-types

```
import random  
number = random.randint(1, 10)  
print(number)
```

OR

```
from random import randint  
number = randint(1, 10)
```

OR

```
from random import *  
number = randint(1, 10)
```

Classes

```
class Student:
    department = "Computer Science"
    fee = 1000

    def __init__(self, n, g):
        self.name = n
        self.grade = g

s = Student("John", "A+")
print(s.name, s.grade, s.department, s.fee)
```

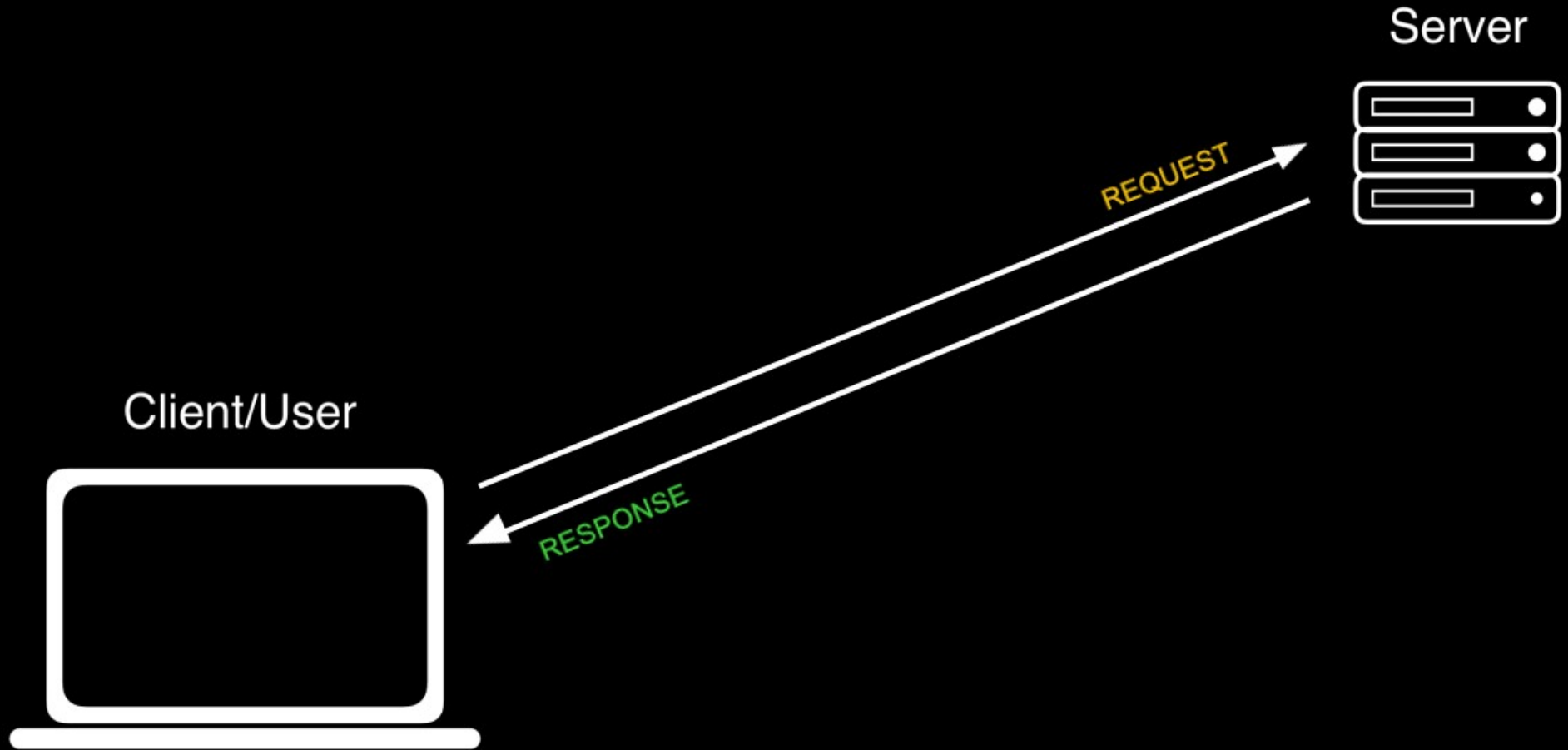
DJANGO

Server



Client/User





GET / HTTP/1.1

Host: www.example.com

...

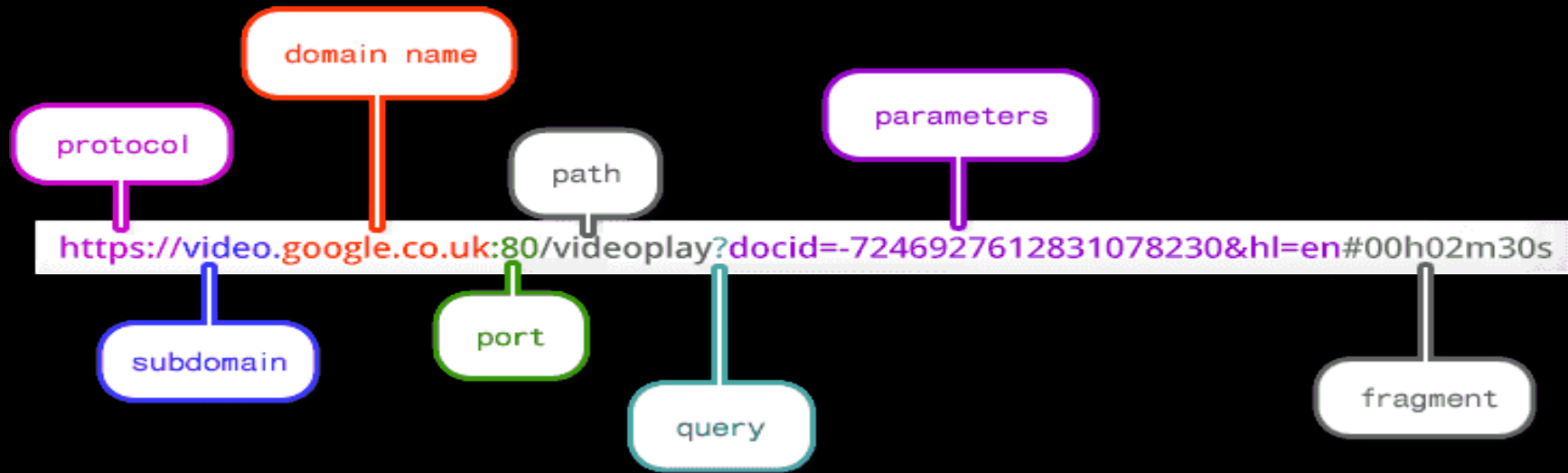
HTTP/1.1 200 OK

Content-Type: text/html

...

Status Code	Description
200	OK
301	Moved Permanently
403	Forbidden
404	Not Found
500	Internal Server Error

Parts of a URL

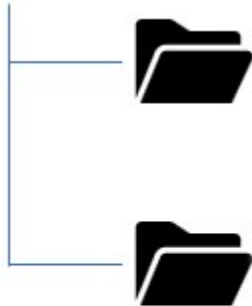


pip install Django

Virtual Environment 1



Python 3.8



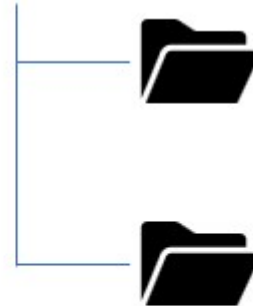
Django 5.0.0

Pandas 1.0.8

Virtual Environment 2



Python 3.4



Django 4.1.1

Pandas 0.2.6

Creating a Virtual Environment

Linux

```
python -m venv myenv
```

```
source myenv/bin/activate
```

```
deactivate
```

Windows

```
python -m venv myenv
```

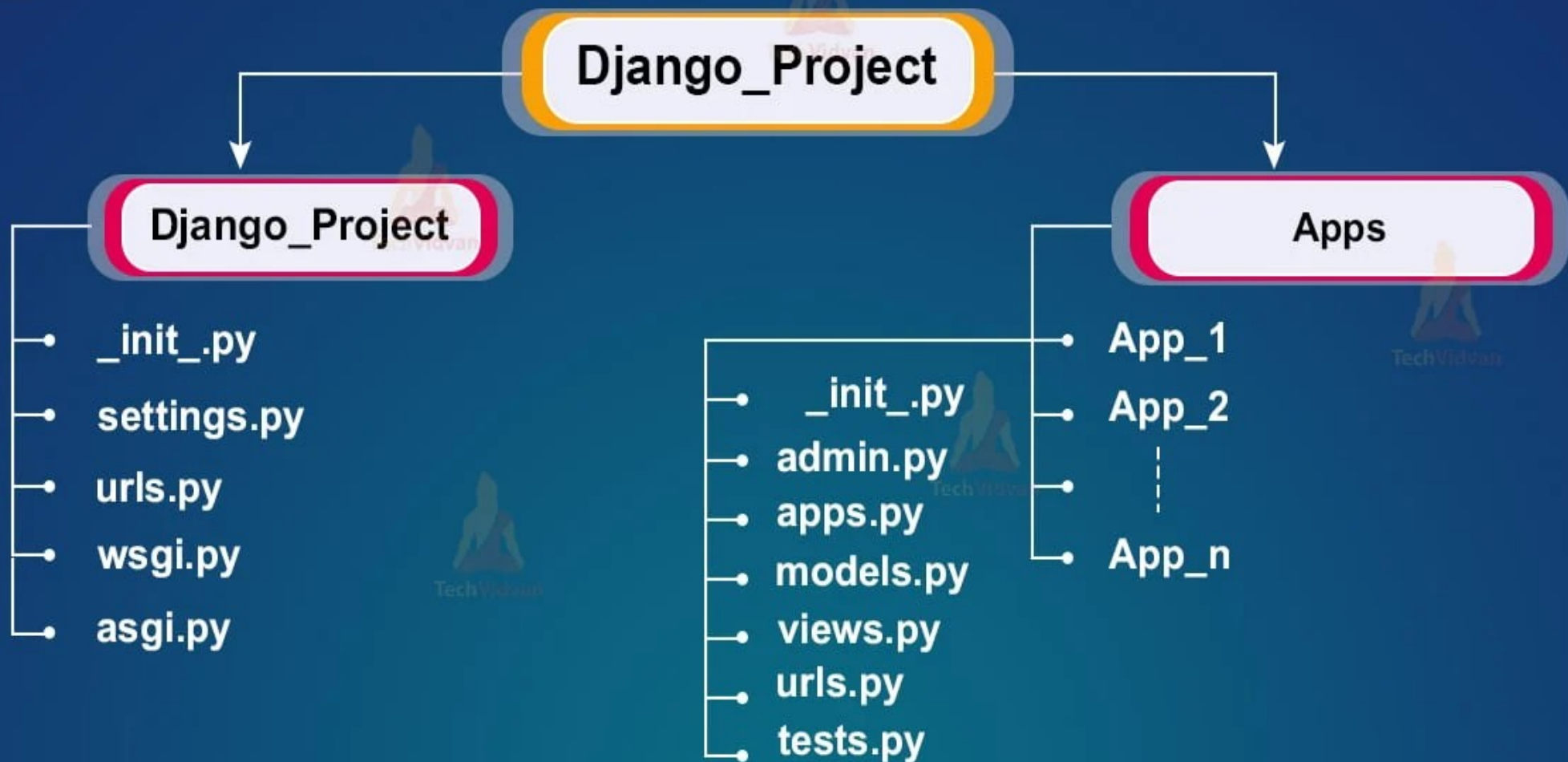
```
.\myenv\Scripts\activate
```

```
deactivate
```

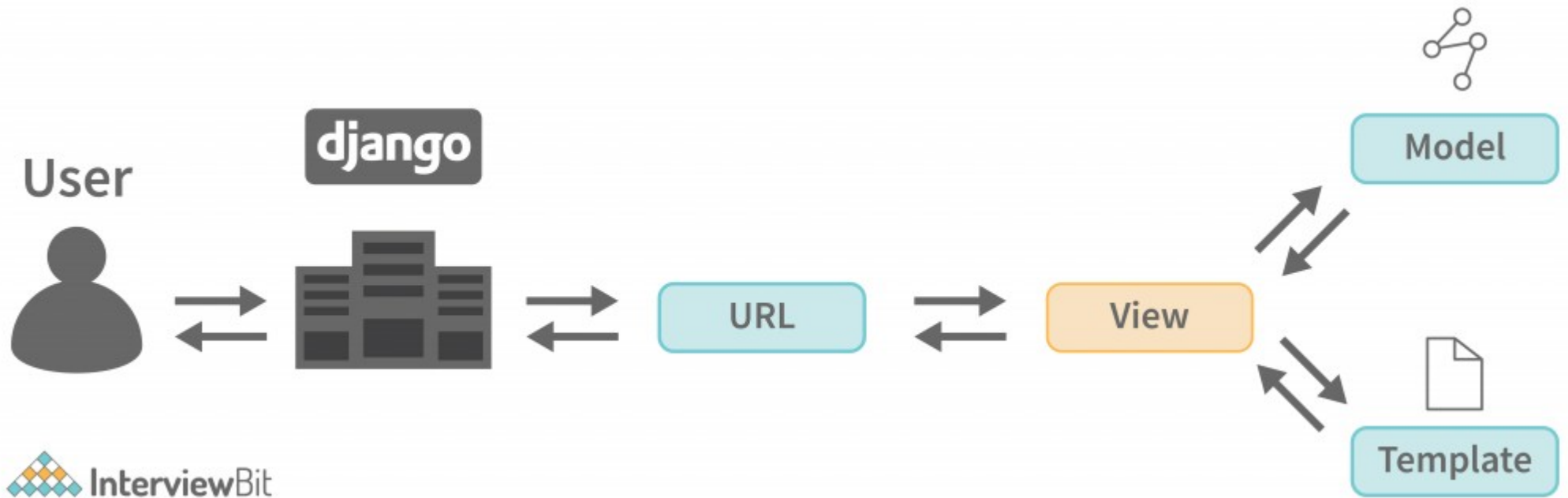
django-admin startproject project-name

```
python manage.py runserver
```

Django File Structure

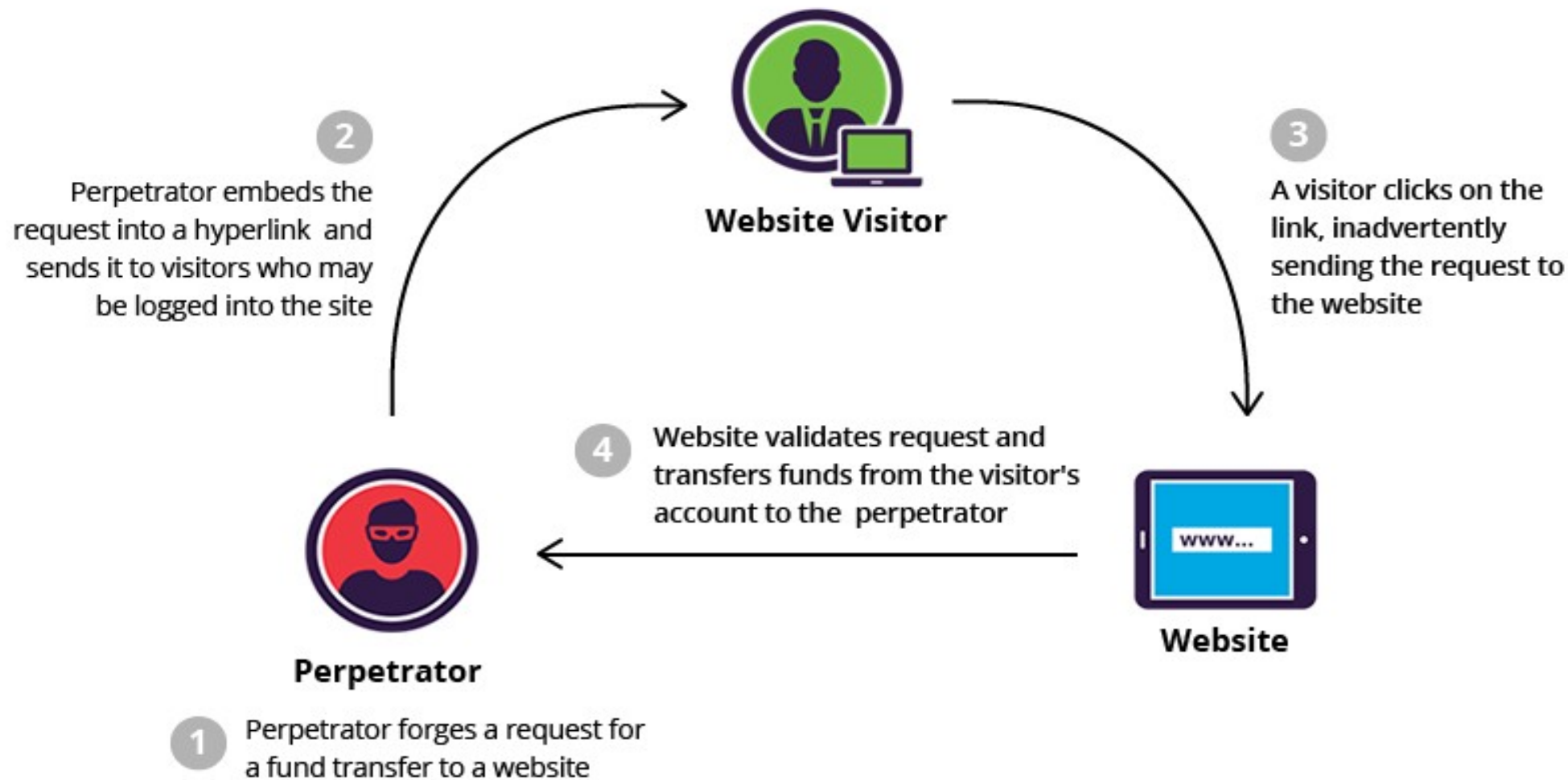


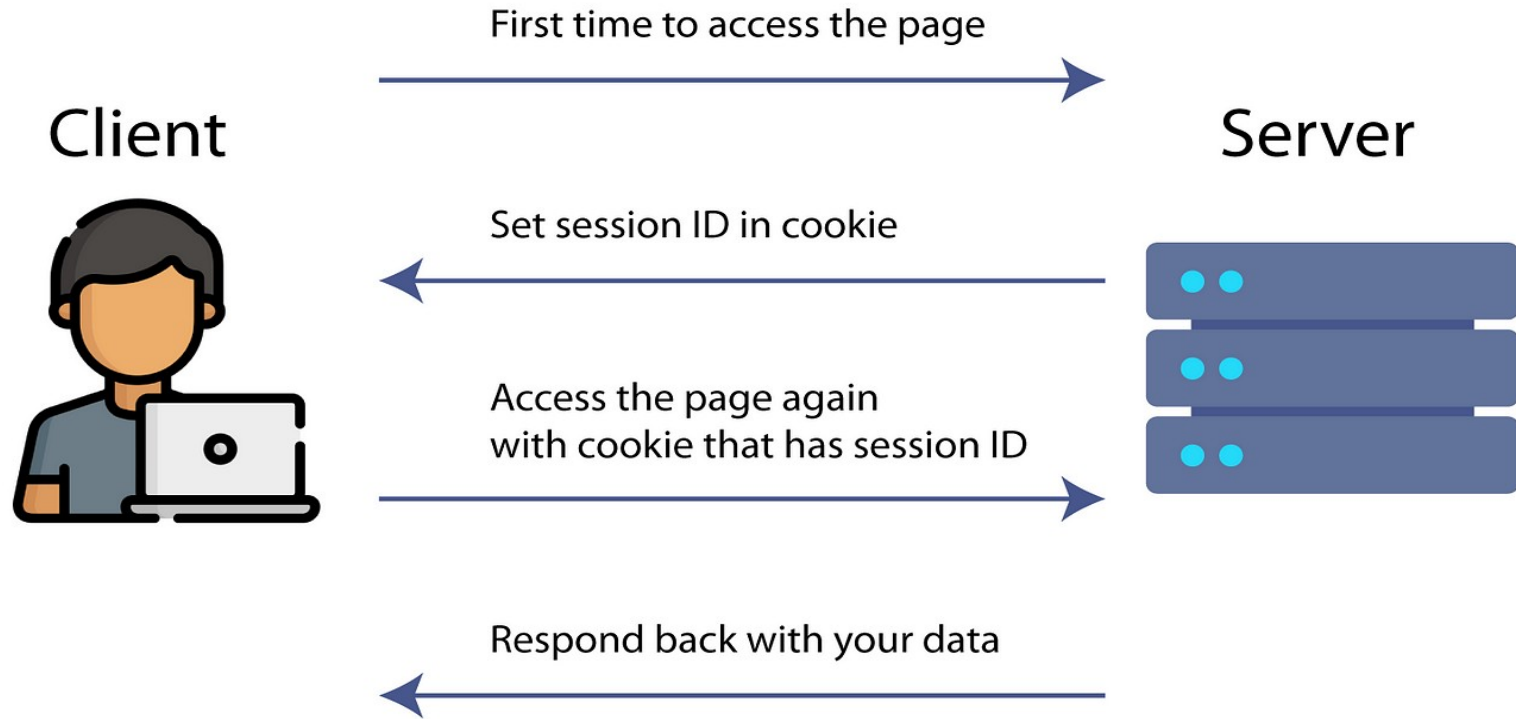
```
python manage.py startapp app-name
```



HTTP Methods and Their Meaning

Method	Meaning
GET	Read data
POST	Insert data
PUT or PATCH	Update data, or insert if a new id
DELETE	Delete data





```
python manage.py migrate
```