

decorators in python.

def fun1 —

def wrapper

@ fun1

def fun0()

↑ [begin ^{start time}

Actual
logic

length end time

def func():
[] start time

func
logic

end time

def func
[]

func.
logic

(end)

decorators

def ~~decorative~~[✓](func):

def wrapper(arg, kw)

[]

→ result = func(arg, kw)

[]

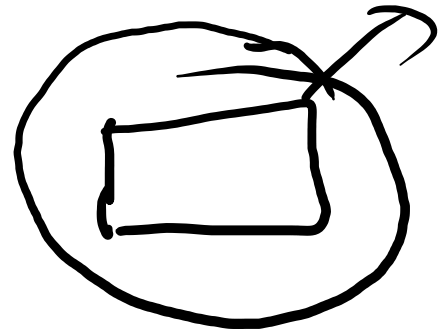
@decorative
def func(name):

logic
return

@decorative
def func():
logic
return

@decorative
def func

→ cutting down
the repetitive
code in
your function



def f1
logics

start

t1

end

start
+
end

→ execution
time

→ 10 milliseconds

→ time of execution

def f2
logics

start

t2

end

→ 15 milliseconds

```
def squares(nume)
    start time: time fin = [1, 4, 9]
```

```
    l = []
    for i in range(nume)
```

```
        l.append(i**2)
```

```
    return l
```

```
def cube
```

```
    i x i
```

$\Rightarrow [1, 4, 9]$

9:00:10sec
end = time fin
diff = end - start
10sec.

```
def cube  
    (start time  
    l = [ ]  
    for  
        i x i x i
```

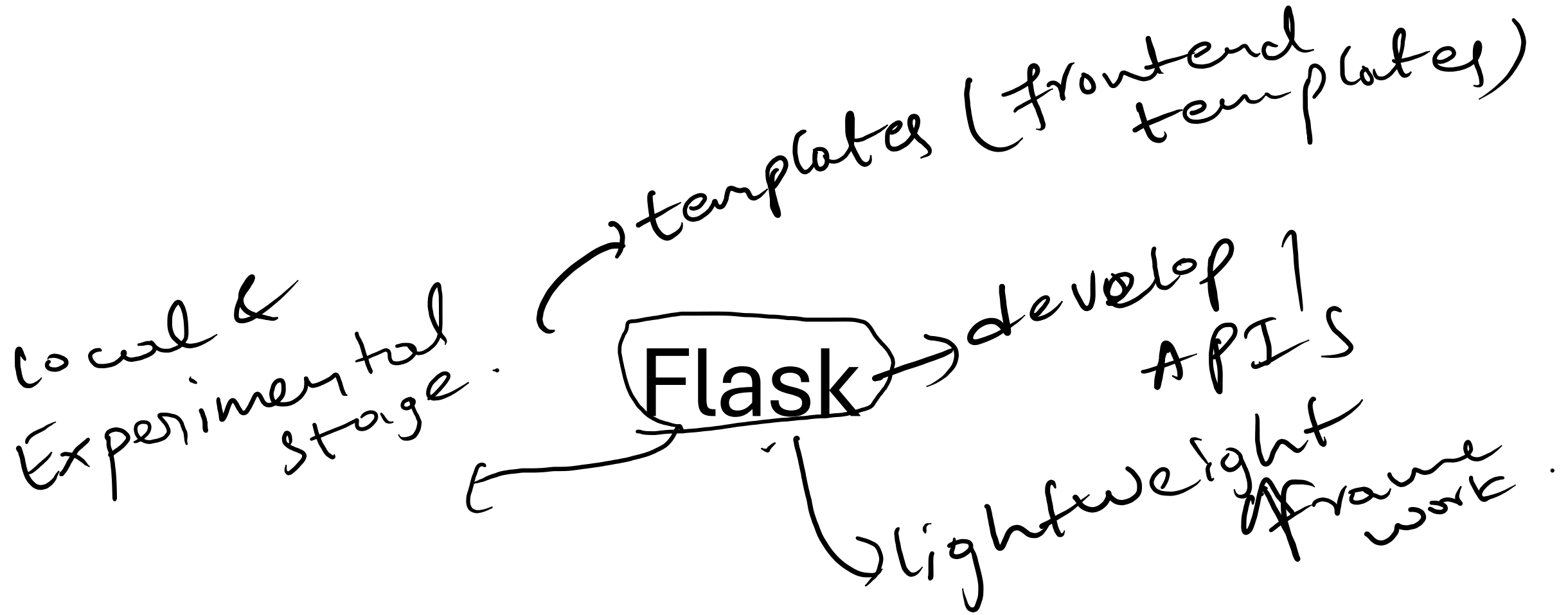
end

diff = end - start → 15 sec.

```
Square } S  
Cube } e  
        d
```

Django → ORM
Very flexible → Object
relational
mapping.

Fast → 4 1) light weight
2) Easier -
3) structured templates
Helps in building
prototypes.



WSGI → Web
Server
Gateway
Interface

web application
built by python

API
HTTPS

• 1. What is Flask?

- Microframework
 - WSGI
 - Werkzeug
 - ✓ Jinja2 templating
 - Flexibility
- <h1>Hello {name}</h1>

- **2. Why Use Flask?**

- Lightweight

- Modular

- Scalable

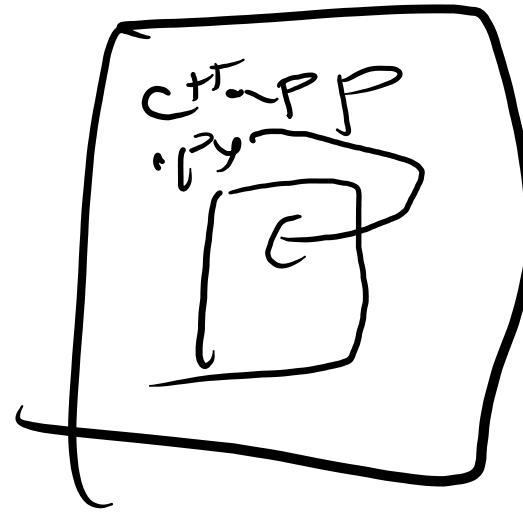
- Community and Ecosystem

- Learning Curve

- **3. Flask vs. Django**

Help you to showcase the core functionality only
→ template → logos, pics → static → Me frame work.

- 4/ **Setting Up Flask**
- ✓ Install Python
- Virtual Environment ✓
- Install Flask ✓
- Verify Installation ✓
- 5. **Creating a Simple Flask Application**
- Application Setup
- Route Definition
- View Function
- Running the Application



2020 / .05
2024 / 2-1
Codechef
Leet