

Type Hints

~~i=0~~ → Python
in general

~~int i=0~~ → Java
additional support

~~int i = "monal"~~ → correct

- * additional support for editor
- * Benefit

* Example!

variable_name : str

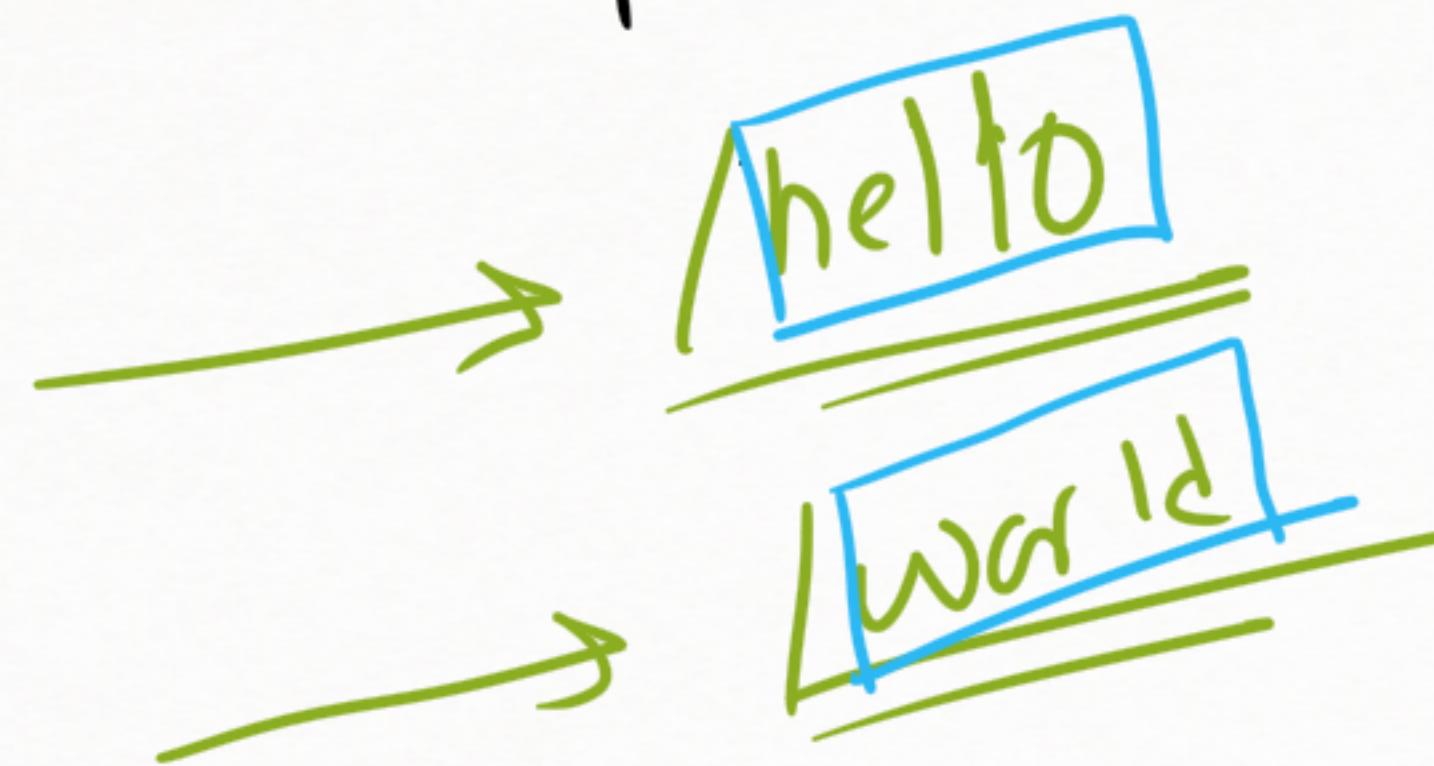
name of variable type

URL

→ https://www.google.com

path → path starts from first /

https://www.google.com /search?q=hi+to+google



HTTP method

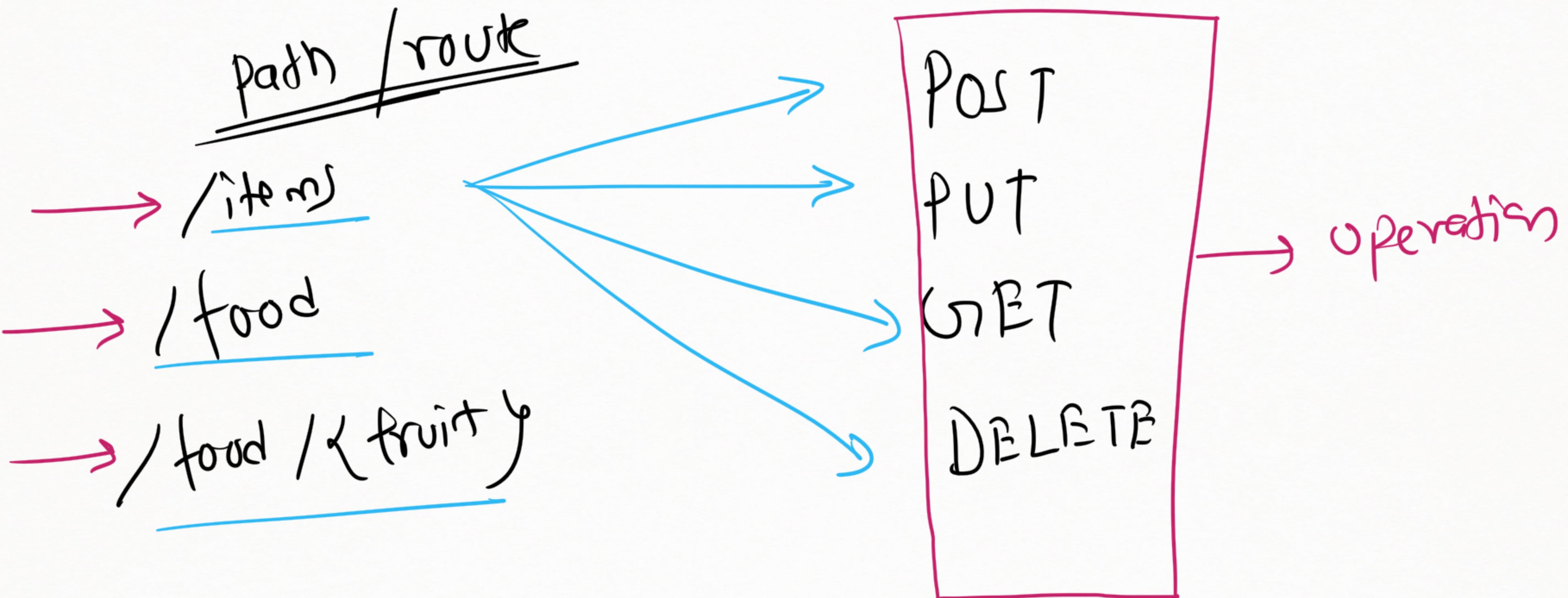
HTTP methods → Operations!

- (1) POST - Create
- (2) GET - Retrive
- (3) PUT - Update
- (4) DELETE - Delete

CRUD →

create , Retrievre/Read , Update ,
Delete

To deal with HTTP
protocols you need to
use one of the operation.



```
name(first-name)
  first-name.upper()
    return first-name
```

all
upper
case
words

```
greeting(func)
```

```
a = func("I am here to!")
  greet you
print(a)
```

print it to
console

```
greeting(name)
```

function

Path Operation function

path → /Hello
→ /fruit
ex. www.google.com/hello

- ✓ get → retrieve
- ✓ put → update
- ✓ post → create
- ✓ delete → delete

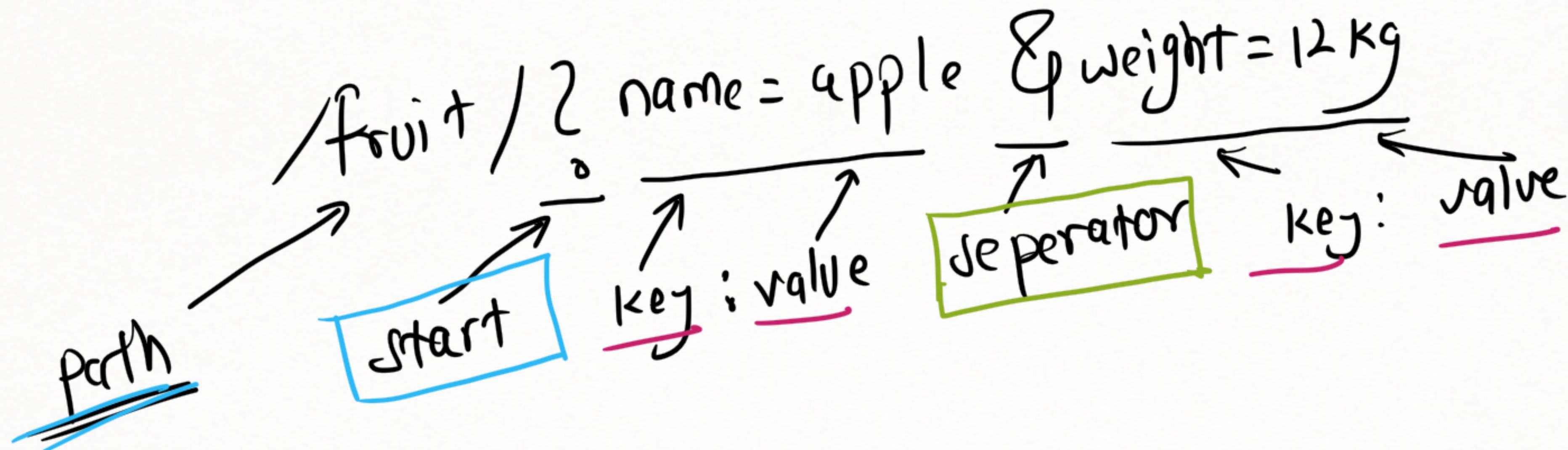
@app.get('/')

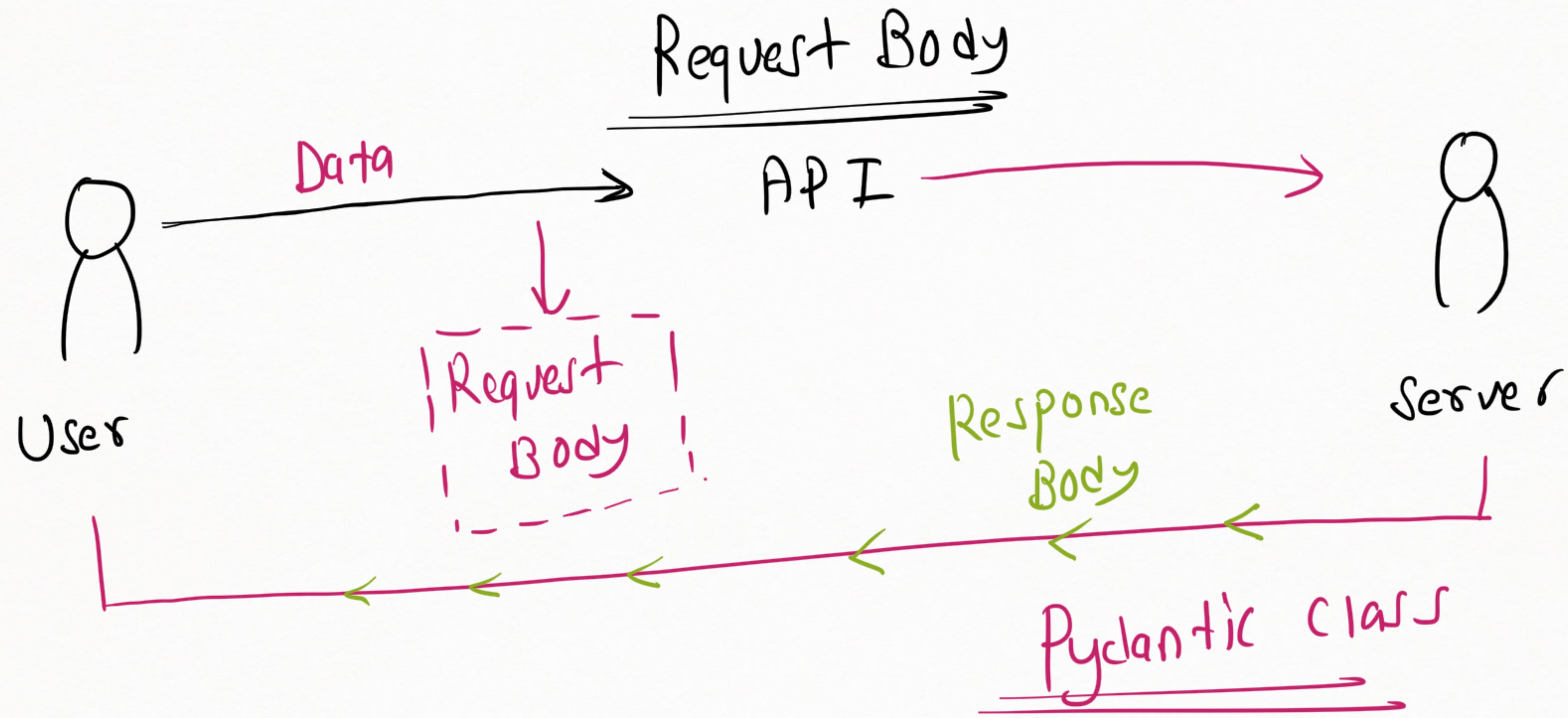
def add():

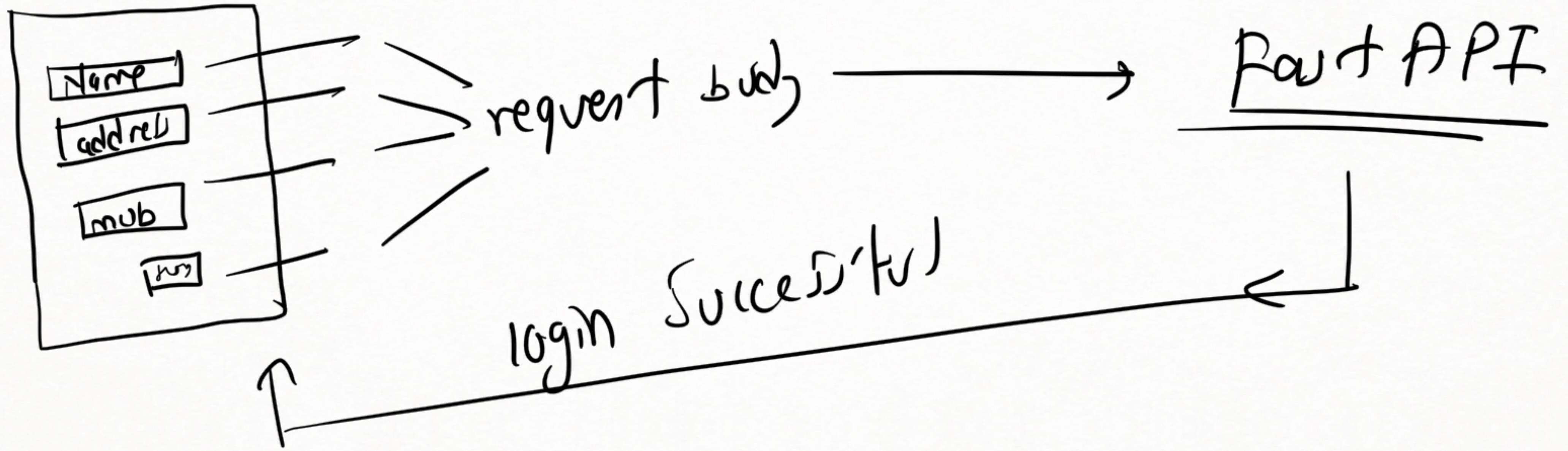
Query Parameters

→ function parameter that are not part of path parameter

Default value → name : str = "monal"





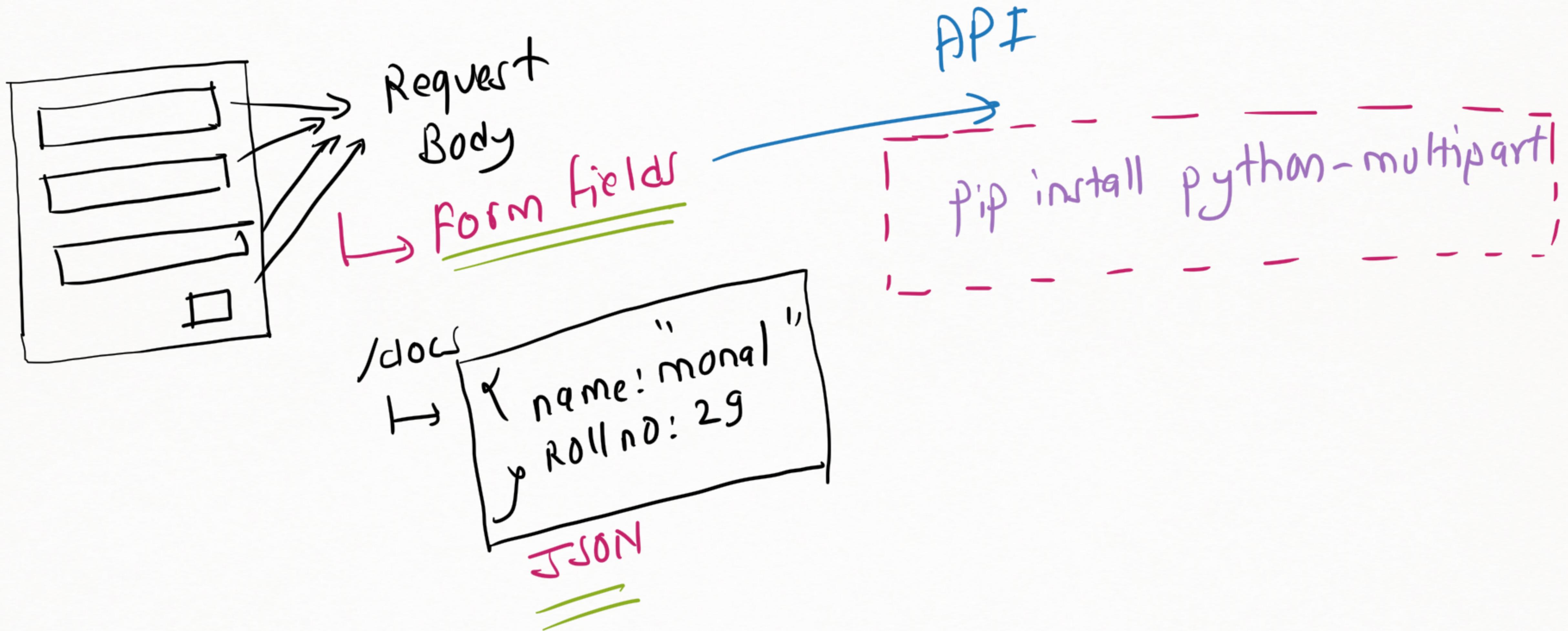


fruit-name = Orange

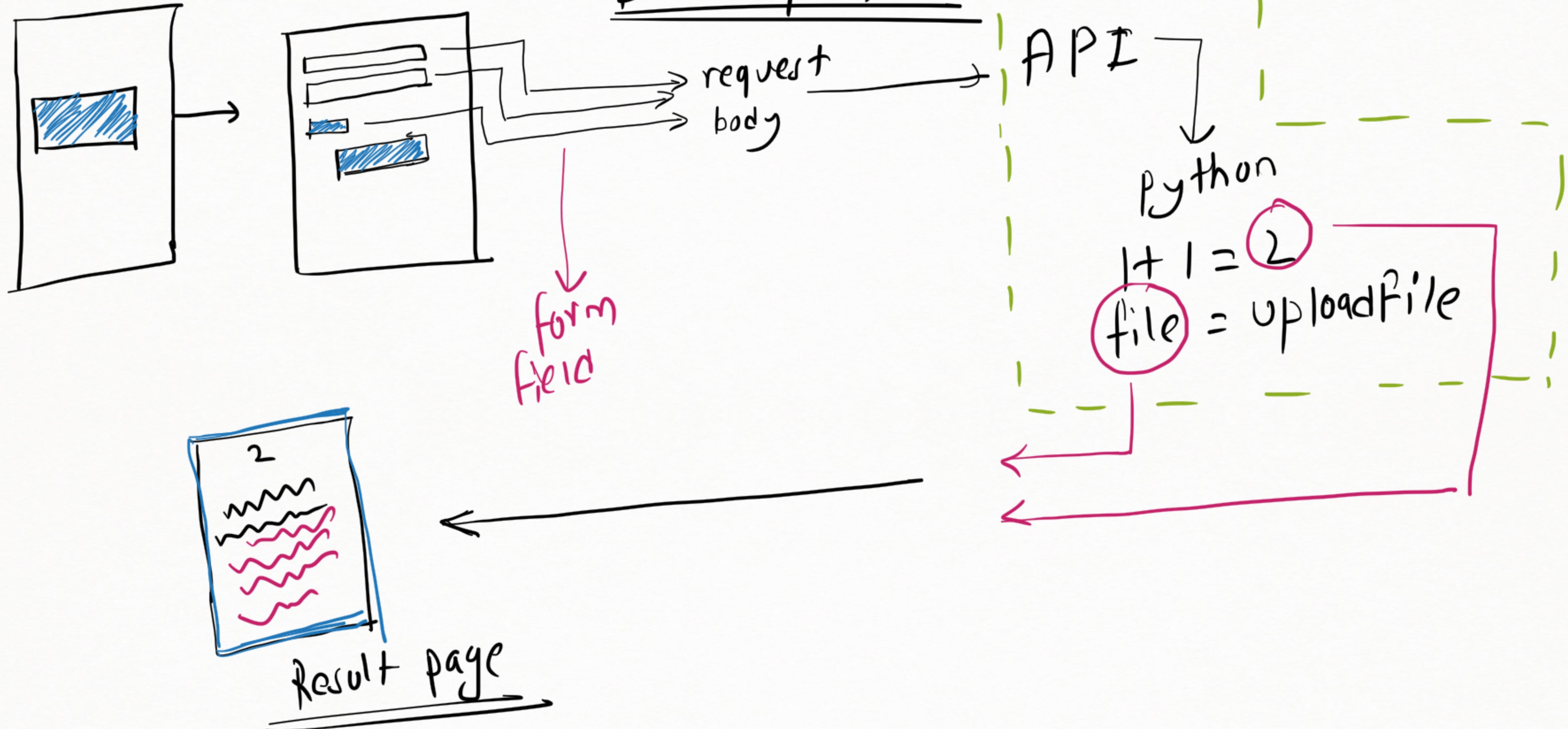
= A - . . . Z → 26 characters

5

Form Data



Demo : project



form → submit → action = "/form/1"

button → href → "/form/1" →

Handling errors

There are many situations in which you need to tell the client about the errors.

- ~~item → item is not found~~
- ~~2 mb/h → internet is not good enough~~
- ~~access → access denied~~
- item → item is not found
- 2 mb/h → internet is not good enough
- access → access denied

Things we have learned

- ✓ what is API & how it works?
 - ✓ fastAPI in-depth overview
 - ✓ Uvicorn
 - ✓ Type hints, Decorator & Enum
 - ✓ Path parameter
 - ✓ Query parameter
 - ✓ Request & Response Body
 - ✓ Form & file Data
- ✓ Handling Error

