



Functional Programming

① Comeback feature →

PSP > 75% for 12 lectures + You have attended all lectures

extra course page / reset credit

Functional Programming

⇒ • FP

- Lambda Function
- Higher Order function
- Decorators

Programming Paradigms

way of writing a code

⇒ FP → A way of coding

where each computation is viewed as evaluation
of a function

⇒ • Code should be readable (Concise)

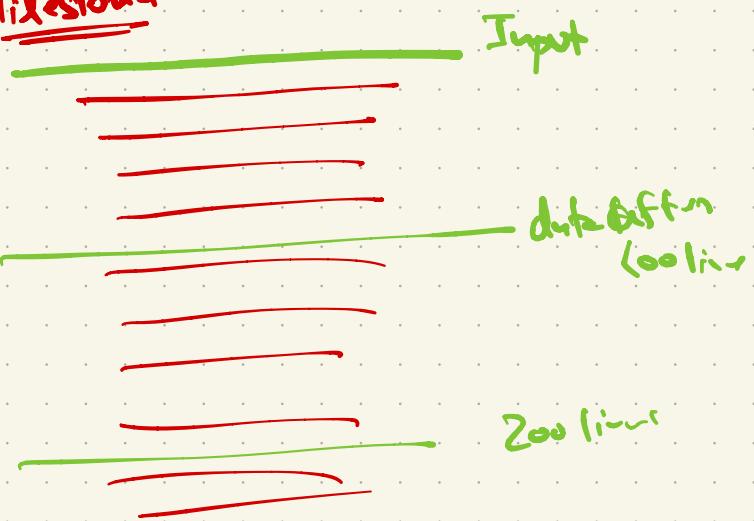
- More maintainable → Easy to debug & solve any issue.

- Efficient → if same activity needs to be repeated

⇒ Functions are used for 2 specific reasons

① Immutable data → it doesn't change your data initial

② Milestones →



③ Functional Programming is a Declarative Style →

We tell Python what to do

rather than

How to do [imperative approach]

① Immutable data → ? Original data can be kept

② Milestones

③ declarative style → tell what to do

How → may be controlled by you
How → some one else.

⇒ All programming language Java Python C++ ...

⇒ Combination of OOPS + FP + Imperative Style

Lambda Function →

- Anonymous functions → don't have a name.
- generally 1 liners

functions = lambda args : return value

Higher Order Function →
Functions which get a funct.

⇒ Why to use HOF →

Scalable ↗ Instr → ↗
↗ Admin → ↗
↗ Student → ↗
↗ TA → ↗

⇒ def attempt_Que():

if user == Inst.:

if StuId

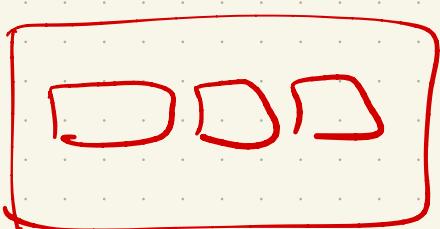
| def stats():

magi-print

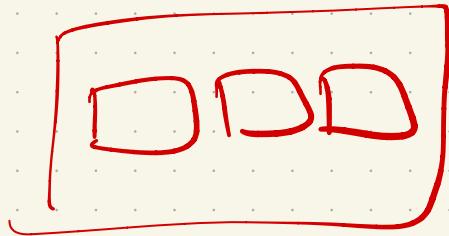
⇒ for based on type of user
return a function

(Send-mail
attempt
calc-work
=)

If else:
return []
- .



Inst



\Rightarrow HOF \rightarrow • Do a pre vs post logging activity

\Rightarrow (Create a function that do certain activities whenever another function is called.)

\Rightarrow Decorators \rightarrow A special use case of HOF.

\Rightarrow HOF that takes another functions as argument.

HOF \rightarrow Function whose output is a function



Decorator \rightarrow Input \rightarrow Function A
Output \rightarrow Function B

\Rightarrow Decorator is very highly used in deployment

- Flask
- Fast API

.

$\Rightarrow \underline{\text{Private}}$

\hookrightarrow Only inside class they can be accessed

Protected

\hookrightarrow Only inside class they can be updated



$\Rightarrow [C, C, C, C, C]$

$A \rightarrow \text{if } A \text{ exist in } \{ :0 \} = \text{?}$

not exist \rightarrow add A

$B \rightarrow [:1] = \{ A \}$

not exist \rightarrow add B

$A \rightarrow [:2] = \{ A, B \}$

any \rightarrow True
 \hookrightarrow False \rightarrow Don't add

$C \rightarrow [:3] \Rightarrow \{ A, B, A \}$

not exist \rightarrow add S

COllab : <https://colab.research.google.com/drive/1HgpSZAedm7rh5nIQ036vkwpmH7BKqATP?usp=sharing>