

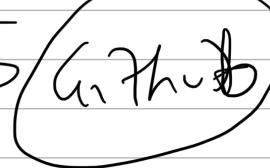
CI/CD

Integration Testing



Dev 1
Dev 2

Commit

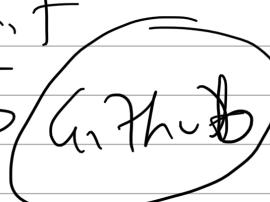


Automation
Test

Continuous Delivery

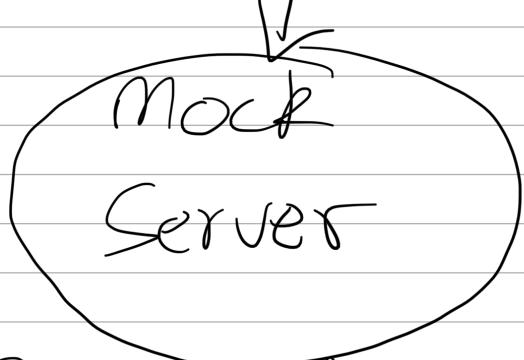
Dev 1
Dev 2

Commit



Build
&
Test

Automatic Product
Server



Continuous Deployment

Dev 1
Dev 2

Commit



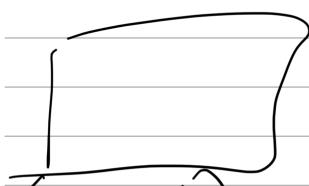
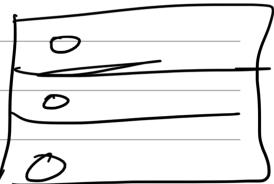
Build
&
Test

Automatic

Product
Server

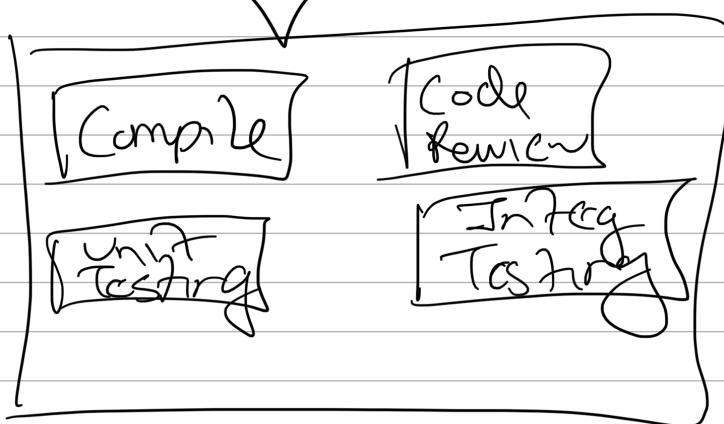
Jenkins

Jenkins Server



Commit
Shared
with
Repository

Build



Deploy
Build application
on test server

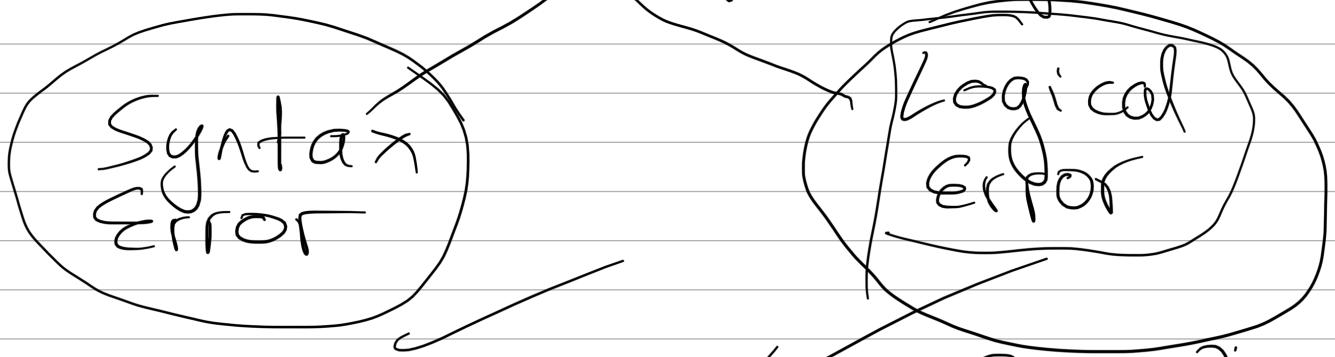
$$\text{Price} - D\% * \text{Price}$$

$$50000 - \frac{20}{100} \times 50000$$

$$50000 - 10000$$

$$= 40000$$

Errors: Problems in the program that stops the execution of program.



str = 'India'

a
g

5
0

a = 5
b = 0

→ Python
Interpreter Based Knowledge

Compiler

English Text

① My name is Interpreter
② My name is Rahu)
③ I Like to play football

213

Stop

① My name (is) Interpreter
② My name is Rahu)
③ I Like to play football

~~5/0~~ zero division
try: ~~5/0~~)
~~except:~~

try: code
except:

try: name='Interpreter'

except: print(" ")

finally: print("My name is Inder")

①

print(name2)

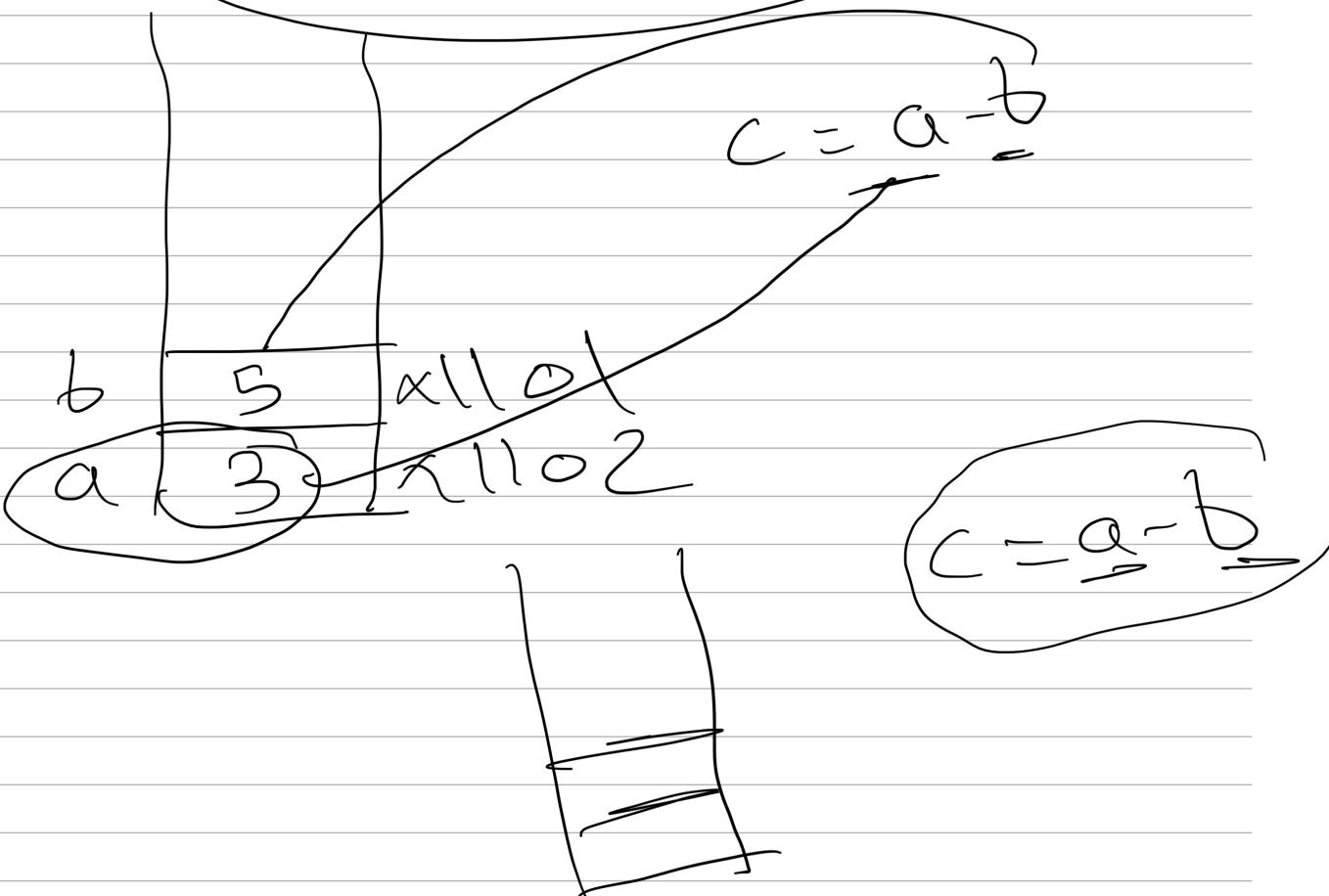
②

name2='Interpreter'

$$a = 3$$

$$b = 5$$

$$c = a - b$$



Functional Programming

→ Function = It is the main building block when we construct Large amount of Code.

What is a Function?

→ A group of statement together to run the statement more than once.

len(str)

Name = 'Independent'

It = 0 2 2 3 4 5 6 7 8 9 to

`len("Interpret")`

def name-of-function(arg1, arg2) :
 Block of code

return

name-of-function(a, b)

def sum-two-number(num1, num2):
 print("Sum is --", num1+num2)

sum-two-number(2, 3.)

Sum is 5

def

Func()

-

X

8

aⁿ

n=3

) Func()

o

$$a^{++2} = a^2$$

$$a^{++3} = a^3$$

$$\text{num}^{++3}$$

Prime No = 1 & number itself

2

1

8 2

3

1

8 3

4 X

1, 2, 4

2 3 5 7, 11 13

%

3 / 2

2 9 0 8 6

1 2 3 4 5 6 7

2 1. Check if $\text{num} \% 2 == 0$
Not prime

2. Number

if $\text{num} \% 2 == 0$ and $\text{num} \geq 2$: X
 return False

for i in range (3, ~~num~~⁶):

 if $\text{num} \% i == 0$:
 return False

$\text{num} = 5$

$5 \% 3$

$5 \% 3 = 2$