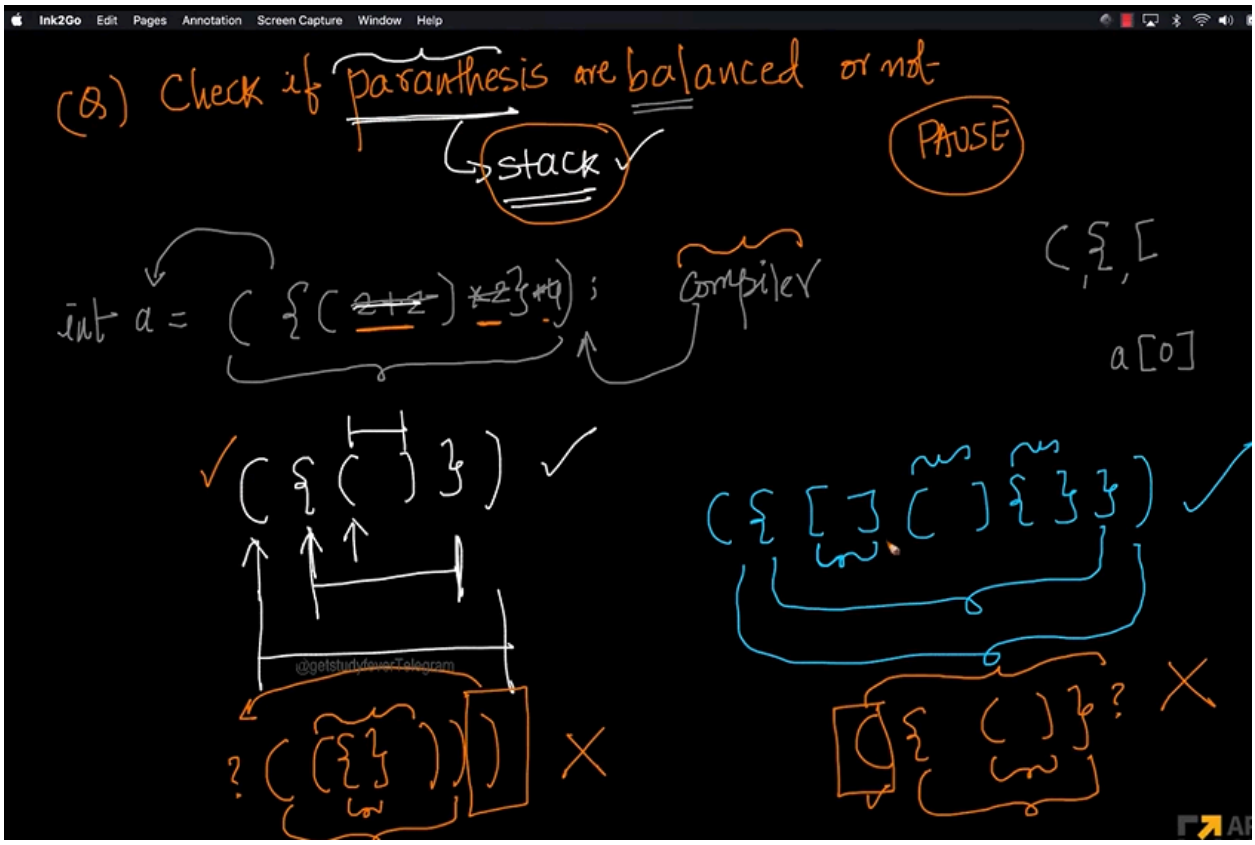
Check If parentheses are balanced or not

1. 

Python code :

1. The code defines two functions:

   - `stack_Stack()` : This function initializes an empty stack.
   - `checkParantheses(s)` : This function takes a string `s` as input and checks if it has balanced parentheses. It does this by iterating through the string and checking each character. If the character is an opening parenthesis ((, [, or {), it is pushed onto the stack. If the character is a closing parenthesis ( ),], or }), it is checked against the top element of the stack. If the characters match (i.e., '(' and ')' or '[' and ']'), then the top element is popped from the stack. If the characters do not match, then the function prints an error message and returns 0.

2.

```
stack_Stack()
def checkParantheses (s):

for i in range(len(s)):      # Iterate through the string
    if s[i]=='(' or s[i]=='[' or s[i]=='{':  # Check if the character

        if s[i]==')' or s[i]==']' or s[i]=='}':   # Check if the char
            stack.push(s[i])  # Push the closing parenthesis onto the

        if stack.isEmpty():  # Check if the stack is empty
            print("Right parentheses are more than left parentheses")

            return 0
        else:
            temp=stack.pop()  # Pop the top element from the stack

            if match(temp,s[i]) != 1:                  # Check if the
                print("Mismatched parentheses are: {} {}".format(temp

            return 0

if stack.isEmpty() is True:  # Check if the stack is empty
    print("Balanced Parentheses")  # Print a message if the parenthes

else:                        # If the stack is not empty, print an er
    return 1

    print("Left parentheses more than right parentheses")
    return 0
```