Mark Lloyd Yadao

IDB2

09/27/22

```python
class Stack:
    new *
    def __init__(self):
        self.items = []

    6 usages  new *
    def isempty(self):
        return len(self.items) == 0

    35 usages  new *
    def push(self, item):
        self.items.append(item)

    24 usages  new *
    def pop(self):
        if not self.isempty():
            return self.items.pop()
        else:
            return "Stack is empty"

    new *
    def size(self):
        return len(self.items)

    4 usages  new *
    def top(self):
        if not self.isempty():
            return self.items[-1]
        else:
            return "Stack is empty"


S = Stack()
S.push(5)
S.push(3)
```

```
print("Lenght:",len(S.items))
print("Pop:",S.pop())
print("Stack Empty:",S.isempty())
print("Pop:",S.pop())
print("Stack Empty:",S.isempty())
print("Pop:",S.pop())
S.push(7)
S.push(9)
print("Top Element:",S.top())
S.push(4)
print("Lenght:",len(S.items))
print("Pop:",S.pop())
S.push(6)
S.push(8)
print("Pop:",S.pop())

S.push(5)
S.push(3)
```

```
Lenght: 2
Pop: 3
Stack Empty: False
Pop: 5
Stack Empty: True
Pop: Stack is empty
Top Element: 9
Lenght: 3
Pop: 4
Pop: 8
Pop: 3
Top Element: 9
Lenght: 7
Pop: 4
Pop: 8
```

2.

```
def push(val):
    stack.append(val)

9 usages  new *
def pop():
    if len(stack) == 0:
        return None
    return stack.pop()

push(5)
push(3)
print(pop())
push(2)
push(8)
print(pop())
print(pop())
push(9)
push(1)
print(pop())
push(7)
push(6)
print(pop())
print(pop())
push(4)
print(pop())
print(pop())
print(pop())
```

OUTPUT

```
3
8
2
1
6
7
4
9
5
```

OUTPUTS

```
Lenght: 2
Pop: 3
Stack Empty: False
Pop: 5
Stack Empty: True
Pop: Stack is empty
Top Element: 9
Lenght: 3
Pop: 4
Pop: 8
Pop: 3
Top Element: 9
Lenght: 7
Pop: 4
Pop: 8
3
8
2
1
6
7
4
9
5
```