

ใส่ค่าเข้าไปใน array

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```

1 A= []
2 A.append("A")
3 A.append("B")
4 A.append("C")
5 A.append("D")
6 A.append("E")
7 A.append("F")
8 A.pop()
9 A.pop()
10 A.pop()
11 A.pop()
12 A.pop()
13 A.pop()
14 print(A)

```

→ line that just executed
→ next line to execute

Print output (drag lower right corner to resize)

Frames: Global frame, A

Objects: list

0	1	2	3	4	5
"A"	"B"	"C"	"D"	"E"	"F"

Step 8 of 14

[Customize visualization](#)

หลังจากนั้นจะทำการ ลบข้อมูลในอเ โดยใช้คำสั่ง .pop

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```

1 A= []
2 A.append("A")
3 A.append("B")
4 A.append("C")
5 A.append("D")
6 A.append("E")
7 A.append("F")
8 A.pop()
9 A.pop()
10 A.pop()
11 A.pop()
12 A.pop()
13 A.pop()
14 print(A)

```

→ line that just executed
→ next line to execute

Print output (drag lower right corner to resize)

Frames: Global frame, A

Objects: empty list

Done running (14 steps)

[Customize visualization](#)

Lab-3.1.2

ใส่ค่าเข้าไปใน array

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```
1 A= []
2 A.append("A")
3 A.append("B")
4 A.append("C")
5 A.append("D")
6 A.append("E")
7 A.append("F")
8 A.pop(0)
9 A.pop(0)
10 A.pop(0)
11 A.pop(0)
12 A.pop(0)
13 A.pop(0)
14 print(A)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 8 of 14

[Customize visualization](#)

Print output (drag lower right corner to resize)

Frames

Global frame

A

Objects

list

0	1	2	3	4	5
"A"	"B"	"C"	"D"	"E"	"F"

ลบค่าใน Array แต่ลบจาก ตำแหน่ง 0 = ตำแหน่งแรกสุด

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```
1 A= []
2 A.append("A")
3 A.append("B")
4 A.append("C")
5 A.append("D")
6 A.append("E")
7 A.append("F")
8 A.pop(0)
9 A.pop(0)
10 A.pop(0)
11 A.pop(0)
12 A.pop(0)
13 A.pop(0)
14 print(A)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 10 of 14

[Customize visualization](#)

Print output (drag lower right corner to resize)

Frames

Global frame

A

Objects

list

0	1	2	3
"C"	"D"	"E"	"F"

Lab-3.2

ทำการโยกค่าใน Array โดยการนำค่าใน Array จากหลังสุดไปใส่ก่อนและทำไปจนครบตามภาพ

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```
1 n = ["A","B","C","D","E","F"]
2 p = []
→ 3 while n :
4     p.append(n.pop())
→ 5 for i in p:
6     n.append(i)
7 print(n)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 16 of 29

Print output (drag lower right corner to resize)

Frames

Global frame

n

p

Objects

empty list

list

0	1	2	3	4	5
"F"	"E"	"D"	"C"	"B"	"A"

[Customize visualization](#)

ทำการคืนค่าใส่ใน Array n และแสดงผล

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
(known limitations)

```
1 n = ["A","B","C","D","E","F"]
2 p = []
3 while n :
4     p.append(n.pop())
5 for i in p:
6     n.append(i)
→ 7 print(n)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Done running (29 steps)

Print output (drag lower right corner to resize)

['F', 'E', 'D', 'C', 'B', 'A']

Frames

Global frame

n

p

i

Objects

list

0	1	2	3	4	5
"F"	"E"	"D"	"C"	"B"	"A"

list

0	1	2	3	4	5
"F"	"E"	"D"	"C"	"B"	"A"

[Customize visualization](#)