

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

Python 3.6
(known limitations)

```

1 def linear_search(Data, n, UserInput):
2     for i in range(0, n):
3
4         if (Data[i] == UserInput):
5             return i
6
7     return -1
8
9 def linear_UserInput(Data, n, UserInput1):
10    for i in range(0, n):
11
12        if (Data[i] == UserInput1):
13            return i
14
15    return -1
16
17
18
19
20
21 Data = [7, 10, 12, 14, 1, 20, 29, 37]

```

Print output (drag lower right corner to resize)

Frames

Global frame

- linear_search
- linear_UserInput
- Data
- UserInput
- UserInput1
- n

Objects

function linear_search(Data, n, UserInput)

function linear_UserInput(Data, n, UserInput1)

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37

linear_search

- Data
- n
- UserInput
- i
- Return value

Step 18 of 38

Customize visualization

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

Python 3.6
(known limitations)

```

1 def linear_search(Data, n, UserInput):
2     for i in range(0, n):
3
4         if (Data[i] == UserInput):
5             return i
6
7     return -1
8
9 def linear_UserInput(Data, n, UserInput1):
10    for i in range(0, n):
11
12        if (Data[i] == UserInput1):
13            return i
14
15    return -1
16
17
18
19
20
21 Data = [7, 10, 12, 14, 1, 20, 29, 37]

```

Print output (drag lower right corner to resize)

Frames

Global frame

- linear_search
- linear_UserInput
- Data
- UserInput
- UserInput1
- n
- func1

Objects

function linear_search(Data, n, UserInput)

function linear_UserInput(Data, n, UserInput1)

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37

linear_UserInput

- Data
- n
- UserInput1
- i
- Return value

Step 36 of 38

Customize visualization

Print output (drag lower right corner to resize)

Answer of 14 is: 3
Answer of 29 is: 6

Frames

Global frame

- linear_search
- linear_UserInput
- Data
- UserInput
- UserInput1
- n
- func1
- func2

Objects

function linear_search(Data, n, UserInput)

function linear_UserInput(Data, n, UserInput1)

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37

Python 3.6
(known limitations)

```

1 def Binary_UserInput1(Data, l, r, UserInput1):
2
3     if r >= 1:
4         mid = 1 + (r - 1) // 2
5         if Data[mid] == UserInput1:
6             return mid
7         elif Data[mid] > UserInput1:
8             return Binary_UserInput1(Data, l, mid-1,
9             else:
10                return Binary_UserInput1(Data, mid + 1, r
11        else:
12            return -1
13
14 def Binary_UserInput2(Data, l, r, UserInput1):
15
16     if r >= 1:
17         mid = 1 + (r - 1) // 2
18         if Data[mid] == UserInput1:
19             return mid
20         elif Data[mid] > UserInput1:

```

Print output (drag lower right corner to resize)

Frames

Objects

Global frame

Binary_UserInput1

Binary_UserInput2

Data

UserInput1 14

UserInput2 29

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37

Binary_UserInput1

Data

l 0

r 7

UserInput1 14

mid 3

Return value 3

Step 12 of 35

Customize visualization

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

Python 3.6
(known limitations)

```

4         mid = 1 + (r - 1) // 2
5         if Data[mid] == UserInput1:
6             return mid
7         elif Data[mid] > UserInput1:
8             return Binary_UserInput1(Data, l, mid-1,
9             else:
10                return Binary_UserInput1(Data, mid + 1, r
11        else:
12            return -1
13
14 def Binary_UserInput2(Data, l, r, UserInput1):
15
16     if r >= 1:
17         mid = 1 + (r - 1) // 2
18         if Data[mid] == UserInput1:
19             return mid
20         elif Data[mid] > UserInput1:
21             return Binary_UserInput2(Data, l, mid-1,
22         else:
23             return Binary_UserInput2(Data, mid + 1, r

```

Print output (drag lower right corner to resize)

Frames

Objects

Global frame

Binary_UserInput1

Binary_UserInput2

Data

UserInput1 14

UserInput2 29

ans1 3

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37

Binary_UserInput2

Data

l 0

r 7

UserInput1 29

mid 3

Binary_UserInput2

Data

l 4

r 7

UserInput1 29

mid 5

Binary_UserInput2

Data

l 6

r 7

UserInput1 29

mid 6

Return value 6

Step 31 of 35

Customize visualization

Element 14 of index is 3
Element 14 of index is 6

Frames

Objects

Global frame

Binary_UserInput1

Binary_UserInput2

Data

UserInput1 14

UserInput2 29

ans1 3

ans2 6

list

0	1	2	3	4	5	6	7
7	10	12	14	1	20	29	37