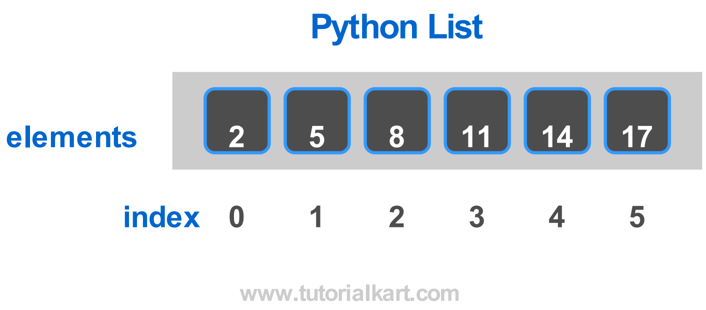
Python Lists

Python Lists

Python List is an ordered collection of elements and you can store objects of any datatype in the list.

As Python List is an ordered collection, you can access the elements using index (position of the element in the element). Like in other programming languages like Java, C, C++, etc, the index starts from `0` until the number of items in the list.



Create an Empty List

To create an empty list in Python, use list() constructor.

```python
aList = list()
```

Or you can also use empty square brackets as shown below.

```python
aList = []
```

Initialize a List in Python

To initialize a list in Python, assign comma separated elements, enclosed in squared bracket to a variable.

```python
aList = [21, 'John', 541, 84.25, True]
```

In addition, let us find out datatype of the variable `aList` programmatically using type() function.

Python Program

```python
aList = [21, 'John', 541, 84.25, True]
print(type(aList))
```

Run the program, and Python interpreter shall print the datatype of `aList` to standard output.

Output

```
<class 'list'>
```

Python List – Access Elements

To access elements of a list in Python, use variable name followed by square brackets, and the index inside the square brackets. An example is provided below.

```
element = aList[index]
```

We already know that index starts from `0`. Therefore, `aList[0]` fetches first element of the list and `aList[5]` fetches 6th element in the list.

Python Program

```
aList = [21, 'John', 541, 84.25, True]

element = aList[2]
print(element)

element = aList[4]
print(element)
```

Output

```
541
True
```

`aList[2]` returns third element of the list and `aList[4]` returns fifth element of the list.

Modify List Element

You can change the value of element at a given index by assigning the value to the element using index.

```
aList[index] = new_value
```

In the following program, we shall modify the element present at the index 2.

Python Program

```
aList = ['apple', 'banana', 'cherry', 'orange', 'papaya']

aList[2] = 'mango'

for element in aList:
    print(element)
```

Output

```
apple
banana
mango
orange
papaya
```

Is given element is present in List

To check if given element is present in the Python list, you can use **in** keyword. The syntax to check if element is in list is

```
element in aList
```

The above expression returns a boolean value. True if element is present in the list and false if not.

Python Program

```
aList = ['apple', 'banana', 'cherry', 'orange', 'papaya']

element = 'banana'
if element in aList:
    print('The element is present in list.')
else:
    print('The element is not present in list.')
```

Output

```
The element is present in list.
```

Python List Length

You can use len() function to get the length of the list. Pass the list as argument to len() builtin function, and it returns an integer.

Python Program

```
aList = [21, 541, 84.25]
print(len(aList))
```

Output

```
3
```

Append Element to Python List

list.append() function appends element to the end of the list.

Python Program

```
aList = [21, 53, 84]
aList.append(96)
print(aList)
```

Output

```
3
```

Insert Element in Python List

list.insert(index, element) method inserts the element at specified index of the list. The index of elements that were originally present from that index are shifted right by one position.

Python Program

```
aList = ['apple', 'banana', 'cherry']

aList.insert(2, 'mango')

for element in aList:
    print(element)
```

Output

```
apple
banana
mango
cherry
```

Iterate over Python List Elements

You can use any looping technique to iterate over elements of a Python list. Let us go through examples with each of the looping technique.

In the following is a program, we will use Python For loop to iterate over list. We are just printing the element, buy you may write multiple statement to process or transform the list element.

Python Program

```
aList = [21, 53, 84]

for element in aList:
    print(element)
```

Output

```
21
53
84
```

In the following is a program, we will use Python While loop to iterate over list.

Python Program

```
aList = [21, 53, 84]

index = 0
while index < len(aList):
    print(aList[index])
    index += 1
```

Output

```
21
53
84
```

Sort a Python List

list.sort() function sorts the list in ascending order by default. You can specify to sort in descending order using ‘reverse’ attribute.

In the following program, we will sort the list of numbers in ascending order.

Python Program

```
aList = [21, 53, 84, 5, 62]

aList.sort()

for element in aList:
    print(element)
```

Output

```
5
21
53
62
84
```

In the following program, we will sort the list of numbers in descending order.

Python Program

```
aList = [21, 53, 84, 5, 62]

aList.sort(reverse=True)

for element in aList:
    print(element)
```

Output

```
84
62
53
21
5
```

Python List Operations

There are many operations that you could perform on a List. Following are some of them. A detailed explanation with example programs have been provided for the each of the below operations.

- Get length of Python List
- Iterate over Elements of Python List
    - Python List While Loop
    - Python List For Loop
- Append an element to Python List
- Add element at specific position in Python List
- Sorting a Python List
- Pop last element from Python List
- Delete specific element or item at given index from Python List
- Convert Tuple into a Python List
- Reverse a Python List

Conclusion

In this Python Tutorial, we learned about List in Python programming, how to access its elements and some of the operations that can be performed on a List.

Python Programming

Control Statements

Python String

Functions

Python Collections

Advanced

Useful Resources