



- Lists are ordered.
- Lists can contain any arbitrary objects.
- List elements can be accessed by index.
- Lists can be nested to arbitrary depth.
- Lists are mutable.

```
• • •
# creating a list
nlis = ['python', 25, 2022]
nlis
```

```
['python', 25, 2022]
```

Indexing

```
• • •

print('Positive and negative indexing of the first element: \n -
Positive index:', nlis[0], '\n - Negative index:', nlis[-3])
print()
print('Positive and negative indexing of the second element: \n -
Positive index:', nlis[1], '\n - Negative index:', nlis[-2])
print()
print('Positive and negative indexing of the third element: \n -
Positive index:', nlis[2], '\n - Negative index:', nlis[-1])
```

Positive and negative indexing of the first element:

- Positive index: python
- Negative index: python

Positive and negative indexing of the second element:

- Positive index: 25
- Negative index: 25

Positive and negative indexing of the third element:

- Positive index: 2022
- Negative index: 2022

What can content a list?

- Strings
- Floats
- Integer
- Boolean
- Nested List
- Nested Tuple
- Other data structures

```
• • •

nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34],
('hello', 'python', 3,14, 2022)]
nlis
```

```
['python',
3.14,
2022,
[1, 1, 2, 3, 5, 8, 13, 21, 34],
('hello', 'python', 3, 14, 2022)]
```


List operations

```
● ● ●  
# take a list  
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34],  
('hello', 'python', 3,14, 2022)]  
nlis
```

```
['python',  
3.14,  
2022,  
[1, 1, 2, 3, 5, 8, 13, 21, 34],  
('hello', 'python', 3, 14, 2022)]
```

```
● ● ●  
# length of the list  
len(nlis)
```

5

Slicing

```
● ● ●  
# slicing of a list  
print(nlis[0:2])  
print(nlis[2:4])  
print(nlis[4:6])
```

```
['python', 3.14]  
[2022, [1, 1, 2, 3, 5, 8, 13, 21, 34]]  
[('hello', 'python', 3, 14, 2022)]
```

Extending the list

- we use the `extend()` function to add a new element to the list.
- With this function, we add more than one element to the list.

```
● ● ●  
# take a list  
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hel-  
lo', 'python', 3,14, 2022)]  
nlis.extend(['hello world!', 1.618])  
nlis
```

```
['python',  
3.14,  
2022,  
[1, 1, 2, 3, 5, 8, 13, 21, 34],  
('hello', 'python', 3, 14, 2022),  
'hello world!',  
1.618]
```

append() method

- As different from the `extend()` method, with the `append()` method, we add only one element to the list
- You can see the difference by comparing the above and below codes.

```
● ● ●  
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hel-  
lo', 'python', 3,14, 2022)]  
nlis.append(['hello world!', 1.618])  
nlis
```

```
['python',  
3.14,  
2022,  
[1, 1, 2, 3, 5, 8, 13, 21, 34],  
('hello', 'python', 3, 14, 2022),  
['hello world!', 1.618]]
```

len(), append(), count(), index(), insert(), max(), min(), sum() functions

```
lis = [1,2,3,4,5,6,7]
print(len(lis))
lis.append(4)
print(lis)
print(lis.count(4)) # How many 4 are on the list 'lis'?
print(lis.index(2)) # What is the index of the number 2 in the list 'lis'?
lis.insert(8, 9) # Add number 9 to the index 8.
print(lis)
print(max(lis)) # What is the maximum number in the list?
print(min(lis)) # What is the minimum number in the list?
print(sum(lis)) # What is the sum of the numbers in the list?
```

```
7
[1, 2, 3, 4, 5, 6, 7, 4]
2
1
[1, 2, 3, 4, 5, 6, 7, 4, 9]
9
1
41
```

Changing the element of a list since it is mutable

```
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]
print('Before changing:', nlis)
nlis[0] = 'hello python!'
print('Aft er changing:', nlis)
nlis[1] = 1.618
print('Aft er changing:', nlis)
nlis[2] = [3.14, 2022]
print('Aft er changing:', nlis)
```

Extending the list

Before changing: ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
Aft er changing: ['hello python!', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
Aft er changing: ['hello python!', 1.618, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
Aft er changing: ['hello python!', 1.618, [3.14, 2022], [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]

Deleting the element from the list using del() function

```
print('Before changing:', nlis)
del(nlis[0])
print('After changing:', nlis)
del(nlis[-1])
print('After changing:', nlis)
```

Before changing: ['hello python!', 1.618, [3.14, 2022], [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
After changing: [1.618, [3.14, 2022], [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
After changing: [1.618, [3.14, 2022], [1, 1, 2, 3, 5, 8, 13, 21, 34]]

```
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]
print('Before deleting:', nlis)
del nlis
print('After deleting:', nlis)
```

Before deleting: ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]

NameError

Traceback (most recent call last)

Input In [19], in <cell line: 4>()

2 print('Before deleting:', nlis)

3 del(nlis)

----> 4 print('After deleting:', nlis)

NameError: name 'nlis' is not defined

Conversion of a string into a list using split() function

```
message = 'Python is a programming language.'  
message.split()
```

['Python', 'is', 'a', 'programming', 'language.']

Use of split() function with a delimiter.

```
text = 'p,y,t,h,o,n'  
text.split(",")
```

['p', 'y', 't', 'h', 'o', 'n']

Basic operations

```
nlis_1 = ['a', 'b', 'hello', 'Python']  
nlis_2 = [1,2,3,4, 5, 6]  
print(len(nlis_1))  
print(len(nlis_2))  
print(nlis_1+nlis_2)
```

4
6
['a', 'b', 'hello', 'Python', 1, 2, 3, 4, 5, 6]

```
print(nlis_1*3)  
print(nlis_2*3)
```

['a', 'b', 'hello', 'Python', 'a', 'b', 'hello', 'Python', 'a', 'b', 'hello', 'Python']
[1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]

```
print(4 in nlis_1)  
print(4 in nlis_2)
```

False
True

Copy the list

```
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]  
copy_list = nlis  
print('nlis:', nlis)  
print('copy_list:', copy_list)
```

nlis: ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
copy_list: ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]

```
# The element in the copied list also changes when the element in  
# the original list was changed.  
# See the following example  
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]  
print(nlis)  
copy_list = nlis  
print(copy_list)  
print('copy_list[0]:', copy_list[0])  
nlis[0] = 'hello python!'  
print('copy_list[0]:', copy_list[0])
```

['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
copy_list[0]: python
copy_list[0]: hello python!

Clone the list



```
# The cloned list is a new copy or clone of the original list.
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]
clone_lis = nlis[:]
clone_lis
```

```
['python',
3.14,
2022,
[1, 1, 2, 3, 5, 8, 13, 21, 34],
('hello', 'python', 3, 14, 2022)]
```



```
# When an element in the original list is changed, the element in
the cloned list does not change.
nlis = ['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3,14, 2022)]
print(nlis)
clone_list = nlis[:]
print(clone_list)
print('clone_list[0]:', clone_list[0])
nlis[0] = 'hello, python!'
print('nlis[0]:', nlis[0])
```

```
['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
['python', 3.14, 2022, [1, 1, 2, 3, 5, 8, 13, 21, 34], ('hello', 'python', 3, 14, 2022)]
clone_list[0]: python
nlis[0]: hello, python!
```

Concatenate the list



```
a_list = ['a', 'b', ['c', 'd'], 'e']
b_list = [1,2,3,4,5,(6,7), True, False]
new_list = a_list + b_list
print(new_list)
```

```
['a', 'b', ['c', 'd'], 'e', 1, 2, 3, 4, 5, (6, 7), True, False]
```