Course	
Term	
Week	
Date	
Chapter. Topic	7. Lists and Tuples

List Comprehension

Siva R Jasthi
Computer Science and Cybersecurity
Metropolitan State University

Lists

List is a collection which is **ordered** and **changeable**. Allows **duplicate** members.

Lists: An introduction
https://www.w3schools.com/python/python_lists.asp

Lists: An introduction
https://openbookproject.net/thinkcs/python/english3e/lists.html

List Methods
http://www.python-ds.com/python-3-list-methods

Built-in Functions
https://docs.python.org/3/library/functions.html

An example of a list

We use SQAURE Brackets to indicate a List.

```
marks = [10, 8, 0,0,9,11,10,10,
```

You can visualize the list as follows:

marks

10	8	7	0	9	11	10	10	4	5
0	1	2	3	4	5	6	7	8	9

List Comprehension = Mapping and/or Filtering

List Comprehension is used to create new list from an existing list based on certain conditions.

List Comprehension → Mapping

List Comprehension → Mapping + Filtering

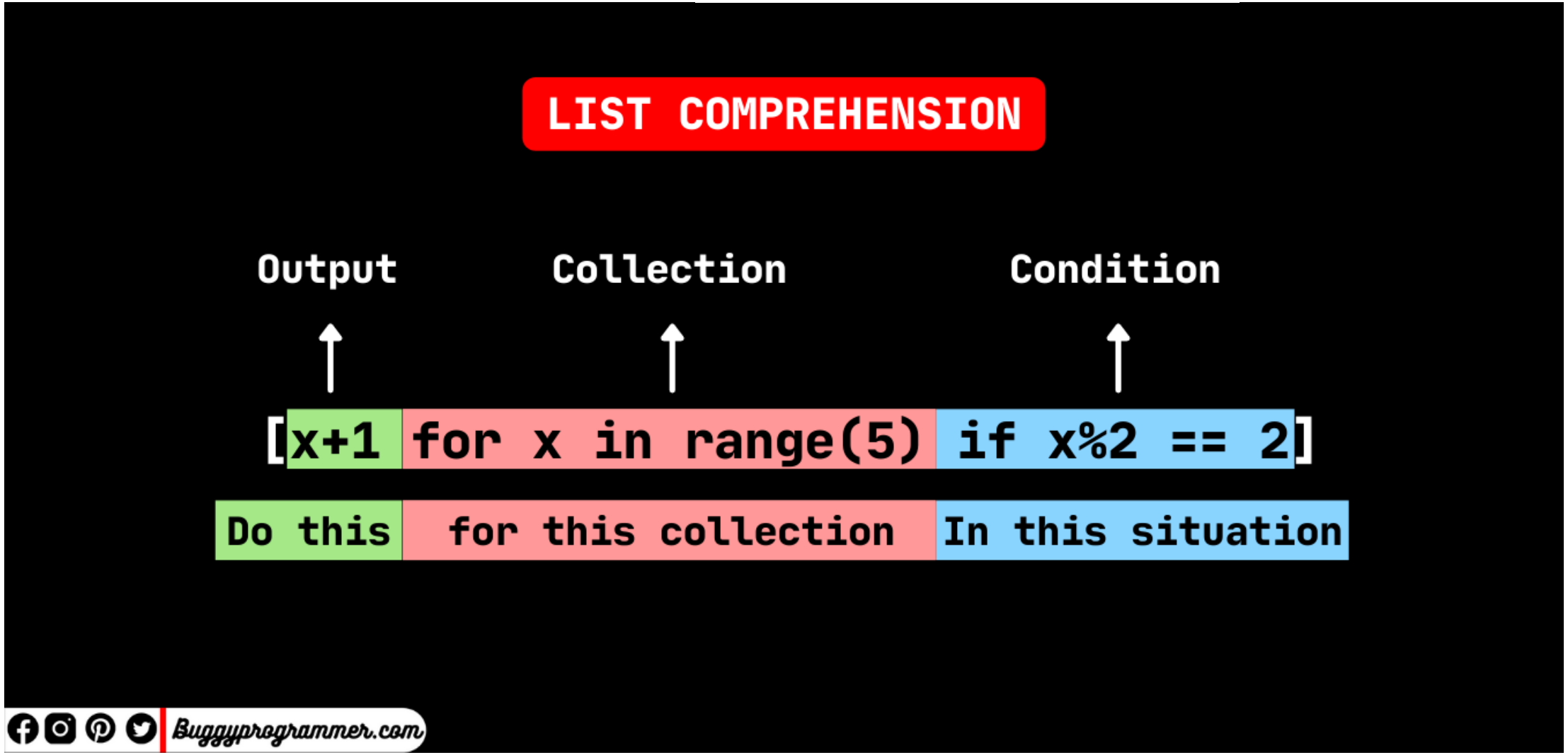
- Elegant way
- Faster way
- Less coding

List Comprehension

```
given_list = [1, 4, 7, 5, 2, 6, 9]

new_list = [x+10 for x in given_list  if x%2 == 0]

print(new_list)
```



List Comprehension (Example 1)

given_list = [1, 4, 7, 1]

Create a new list by adding number 10 to each element (mapping)

new_list = [11, 14, 17, 11]

Without the
"List Comprehension"

```
given_list = [1, 4, 7, 1]

new_list = []

for elem in given_list:
    new_elem = elem + 10
    new_list.append(new_elem)

print(new_list)
```

List Comprehension (Example 2)

given_list = [1, 4, 7, 1, 2]

Create a new list by adding number 5 to each element (mapping).

However, do this only if the list element is EVEN. (filtering)

new_list = [9, 7] # mapping and filtering at the same time

Without the
"List Comprehension"

```python
given_list = [1, 4, 7, 1, 2]

new_list = []

for elem in given_list:
    if elem % 2 == 0:
        new_elem = elem + 5
        new_list.append(new_elem)

print(new_list)
```

List Comprehension (Example 1.c)

given_list = [1, 4, 7, 1]

Create a new list by adding number 10 to each element (mapping)

new_list = [11, 14, 17, 11]

With the
"List Comprehension"

```
#Using the list comprehension
given_list = [1, 4, 7, 1]

new_list = [x+10 for x in given_list]

print(new_list)
```

List Comprehension (Example 2.c)

given_list = [1, 4, 7, 1, 2]

Create a new list by adding number 5 to each element (mapping).

However, do this only if the list element is EVEN. (filtering)

new_list = [9, 7] # mapping and filtering at the same time

With the
"List Comprehension"

```
6
7  # Using the list comprehension (mapping & filtering)
8
9  given_list = [1, 4, 7, 1, 2]
10
11 new_list = [x + 5  for x in given_list if x % 2 == 0]
12
13 print(new_list)
14
```

List Comprehension (output, sequence, condition)

Syntax:
The return value is a new list, leaving the old list unchanged.

newlist = [output *expression* for *item* in *iterable* if *condition* == True]

Condition:
The condition is like a filter that only accepts the items that valuate to True.
The *condition* is optional and can be omitted

Iterable:
The *iterable* can be any iterable object, like a range, list, tuple, set, dictionary etc.

expression:
The *expression* can also contain conditions, not like a filter, but as a way to manipulate the outcome:

List Comprehension: Example 1 (ex.1, ex.2)

Based on a list of fruits, you want a new list, containing only the fruits with the letter "a" in the name.

Without list comprehension you will have to write a for statement with a conditional test inside:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)

print(newlist)
```

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist)
```

List Comprehension: Example 2

Create a new list with only even numbers of a given list

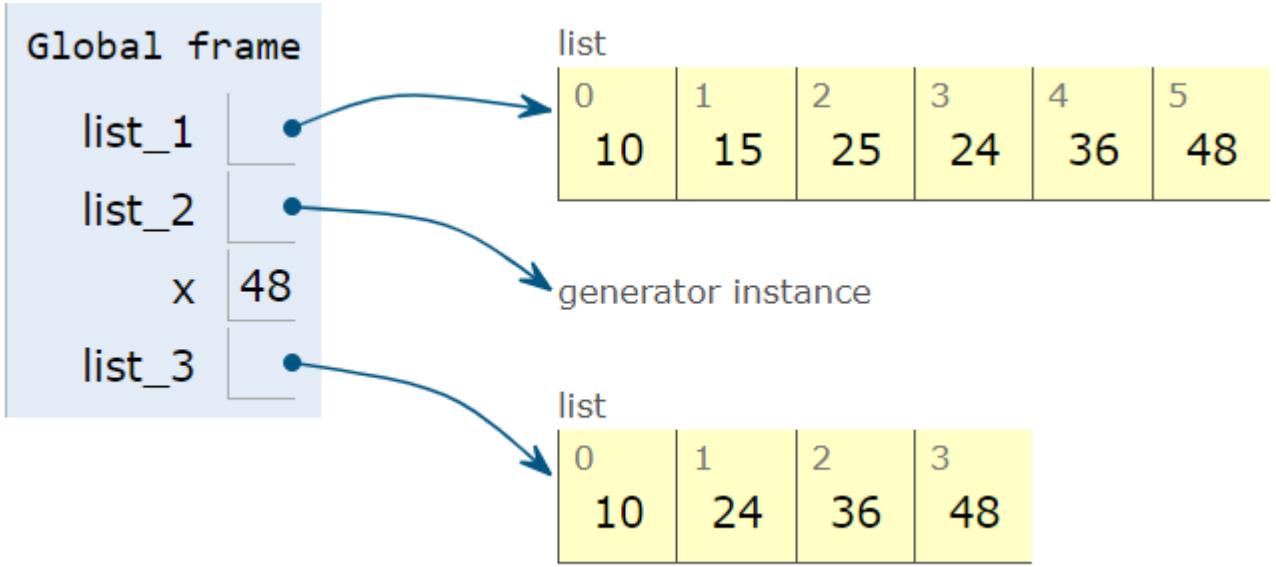
```
list_1 = [10, 15, 25, 24, 36, 48]

list_2 = []
for x in list_1:
    if x%2 == 0:
        list_2.append(x)
```

```
list_3 = [x for x in list_1 if x % 2 == 0)
```

Common Mistake

List Comprehension creates a new list.
Please make sure that you use square brackets.




```
1  list_1 = [10, 15, 25, 24, 36, 48]
2
3  list_2 = []
4  for x in list_1:
5      if x%2 == 0:
6          list_2.append(x)
7
8
9
10 list_2 = (x for x in list_1 if x % 2 == 0)
11
12 list_3 = [x for x in list_1 if x % 2 == 0]
13
```

List Comprehension: Condition

Condition is optional.

If you drop it, expression is applied on every element of the list.

expression **for** *item* **in** *iterable* **if** *condition*

expression **for** *item* **in** *iterable*

List Comprehension: Expression

Expressions can also contain conditions.
But the conditions should have both “if..else” clauses.

expression1 **if** *condition* **else** *expression2* **for** *item* **in** *iterable*

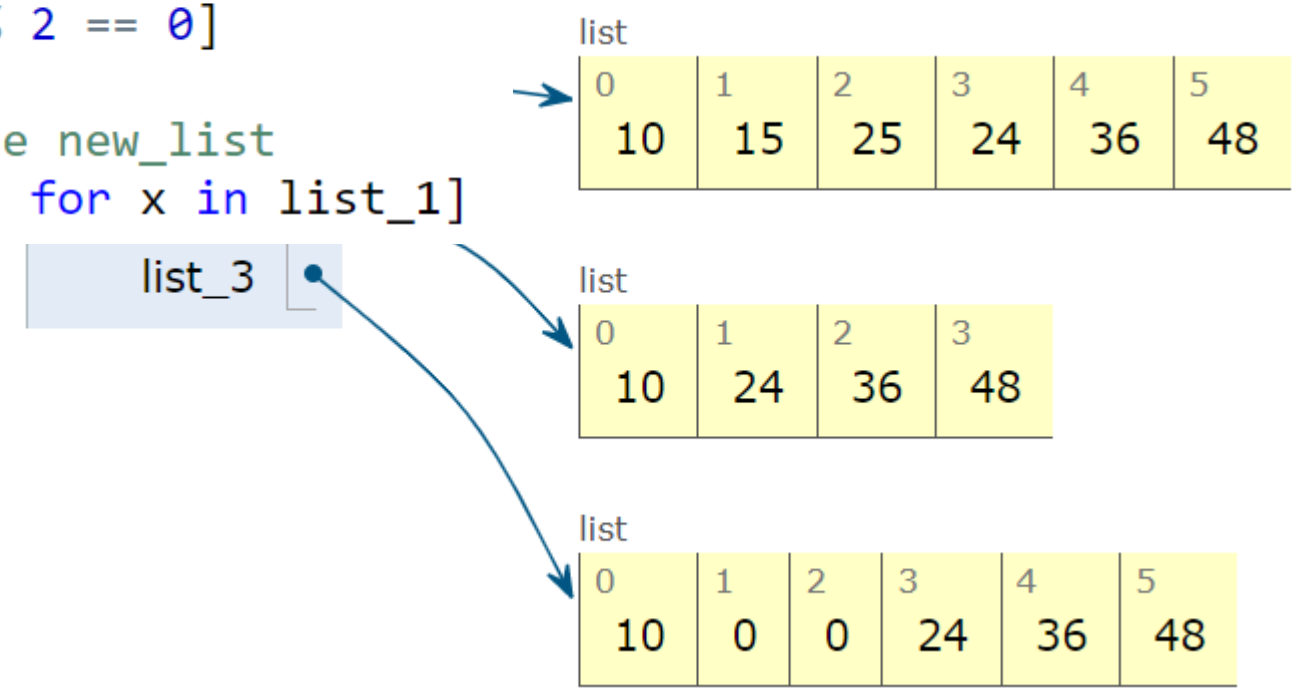
If you want to apply some condition on every element of the list, embed the condition in the expression.

If you want to filter out some elements, then apply the condition to every element.

Condition in Output Expression

expression1 **if** *condition* **else** *expression2* **for** *item* **in** *iterable*

```
1  list_1 = [10, 15, 25, 24, 36, 48]
2
3
4  # Remove the odd elements fromt he list
5  list_2 = [x for x in list_1 if x % 2 == 0]
6
7  # Mark the odd elements as 0 in the new_list
8  list_3 = [(x if x % 2 == 0 else 0) for x in list_1]
```



Condition in Output Expression (ex)

expression1 **if** *condition* **else** *expression2* **for** *item* **in** *iterable*

```
numbers = [10, 13, 42, 24, 47]

# just copy the number as-is
#expression (x = x)
new_list1 = [x for x in numbers]

# add 1 to every number
#expression (x = x+1)
new_list2 = [x+1 for x in numbers]

## add 1 if even number
# subtract 1 if odd number
# expression (x+1) if x % 2 == 0 else (x-1)
new_list3 = [ (x+1) if x % 2 == 0 else (x-1) for x in numbers]

print(even_numbers)
```

Condition in Output Expression (ex)

expression1 **if** *condition* **else** *expression2* **for** *item* **in** *iterable*

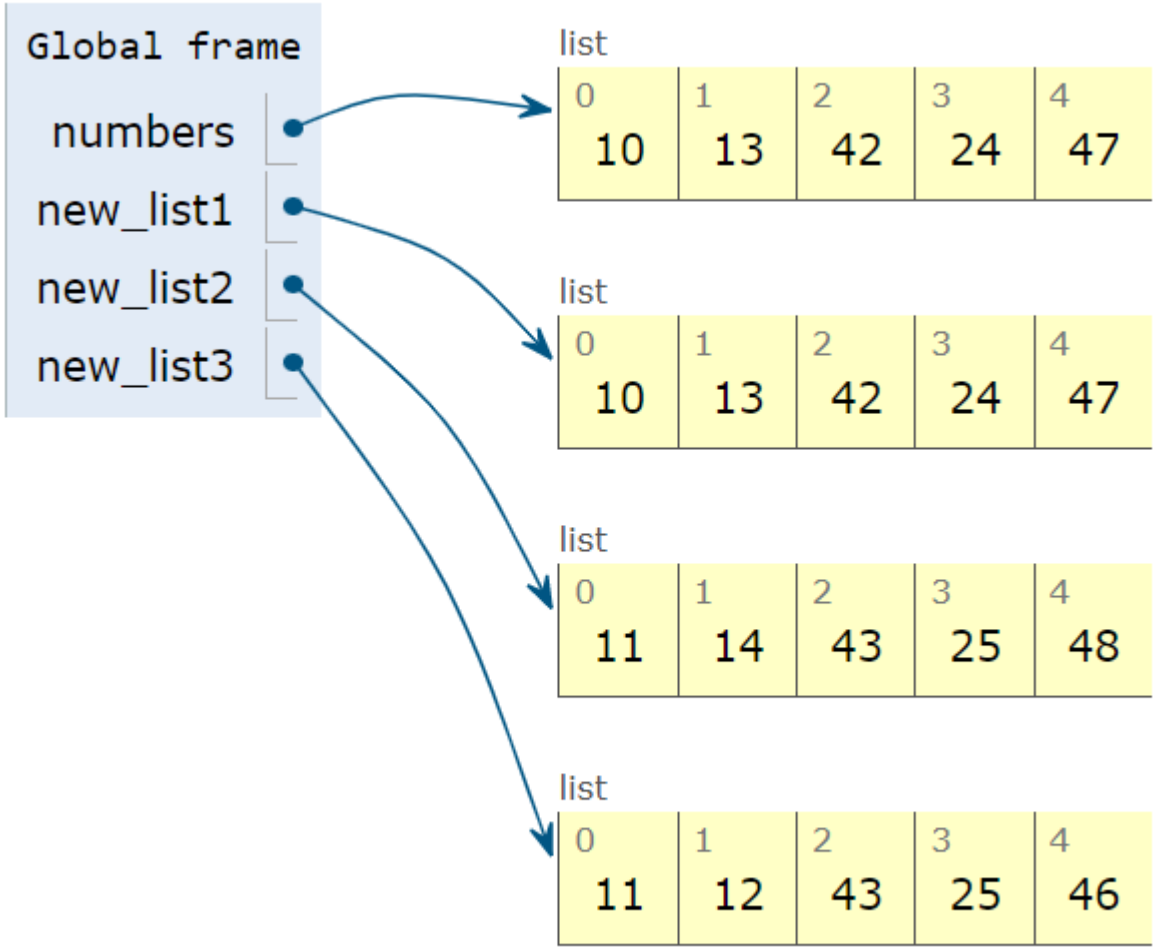
```python
numbers = [10, 13, 42, 24, 47]

# just copy the number as-is
#expression (x = x)
new_list1 = [x for x in numbers]

# add 1 to every number
#expression (x = x+1)
new_list2 = [x+1 for x in numbers]

## add 1 if even number
# subtract 1 if odd number
# expression (x+1) if x % 2 == 0 else (x-1)
new_list3 = [ (x+1) if x % 2 == 0 else (x-1) for x in numbers]

print(even_numbers)
```



Cool List Comprehensions #1 (example)

Split a string into a character list

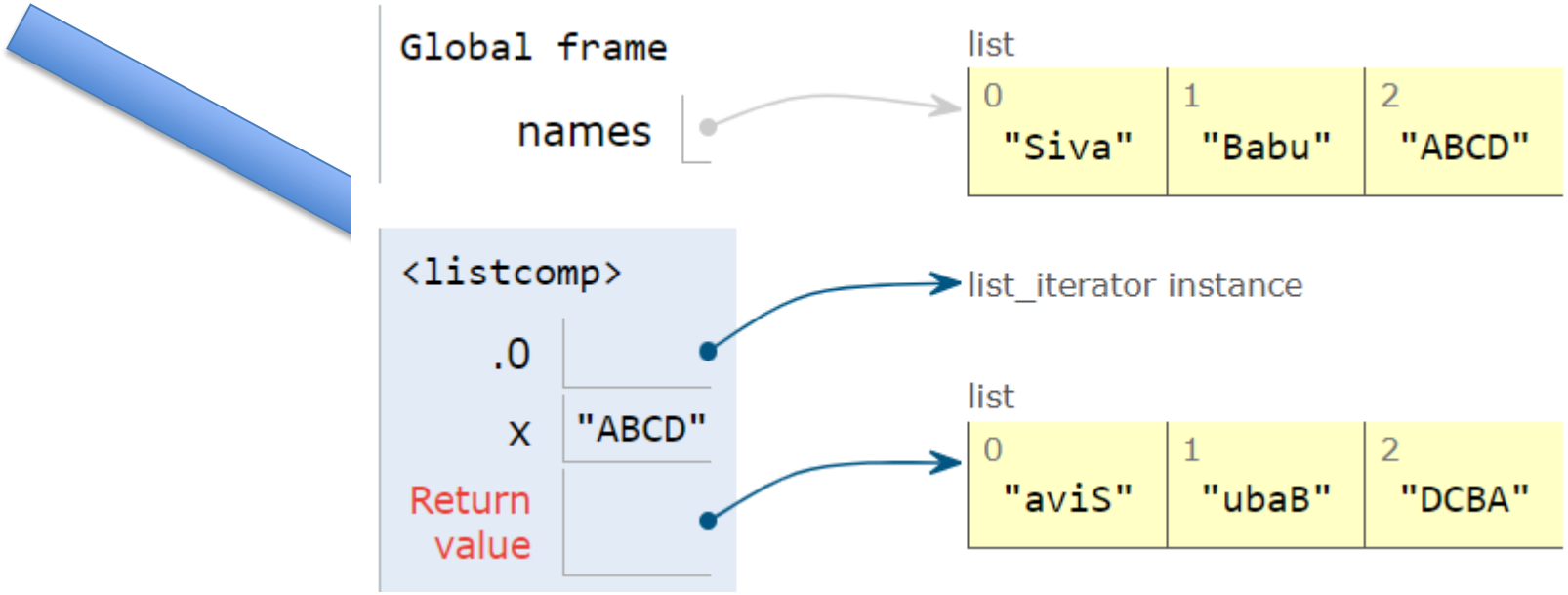
```
1  str = 'Programming'
2  chars = []
3
4  for x in str:
5      chars.append(x)
6
7  print(chars)
```

```
1  str = 'Programming'
2
3  chars = [x for x in str]
4
5  print (chars)
```

Cool List Comprehensions #2

Reverse each string in a list

```
1  names = ['Siva', 'Babu', 'ABCD']
2
3  names_r = [x[::-1] for x in names]
4
5  print (names_r)
```



Summary

List Comprehensions are used to create new lists based on existing lists.

The following elements come into play during the list comprehension.

- Output Expression
- Input Sequence
- **Optional** Conditional predicate