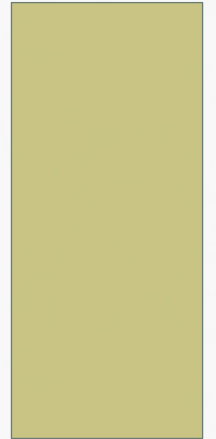


# PYTHON LIST

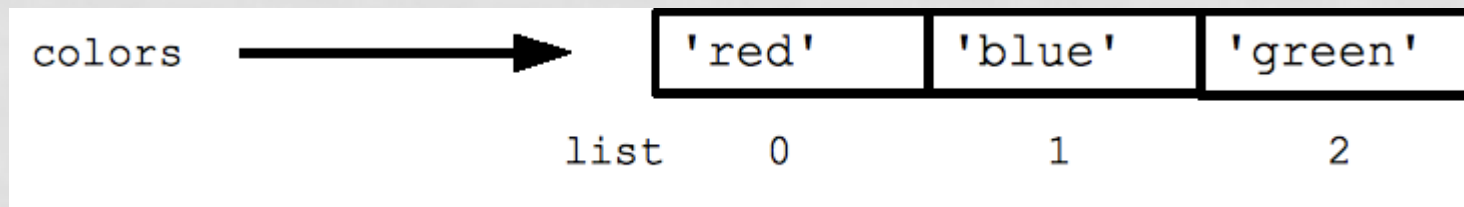
-PYTHON SERIES



- List items are ordered, changeable, and allow duplicate values.
- It is created by placing all the items (python objects) inside square brackets [], separated by commas.
- Python objects may be of different types (integer, float, string etc.)

# CREATE

- Lists in Python can be created by just placing the sequence inside the square brackets[].
- `colors = ['red', 'blue', 'green']`



# CONTINUE

- To ensure List are ordered you can check below statements
- `a = [1,2,"Peter",4.50,"Ricky",5,6]`
- `b = [1,2,5,"Peter",4.50,"Ricky",6]`
- `a == b`
- A list with a single object is sometimes referred to as a singleton list

# THE LIST() CONSTRUCTOR

- It is also possible to use the list() constructor when creating a new list.
- `x= list(("apple", "banana", "cherry"))`
- # note the double round-brackets

# TYPE CASTING-LIST

- `X = 'hello'`
- `Y = list(x)`
- `List(Iterable_object)`

# READ OR ACCESS

- List items are indexed and you can access them by referring to the index number that may be positive or negative
- Positive index start with zero ( First Element)
- Negative index start with -1 ( Last Element)
- You can also use slicing to select multiple element in the list. `[:]` and `[start:stop:step]`

# UPDATE

- refer the index number

```
x= ["apple", 56.32, 23]
```

```
x[0] = "blackcurrant"
```

#you can also mention range of index

```
x[1:3] = [99.99, "watermelon"]
```



# CONTINUE..

- In case you specify more elements during update then it will insert the remaining value

```
x[1:2] = [99.99, "watermelon"]
```

- In case you specify less items than you replace , then you will lose particular index value

# *INSERT NEW ELEMENTS*

- Append
- Concatenation
- Insert
- Extend

# APPEND

- *It will add* an item to the end of the list

```
x= ["Hello", "banana", 23]
```

```
x.append("orange")
```

# CONCATENATION

- Syntax
- List A
- List B
- List C = List A + List B

# INSERT

- The insert() method inserts an item at the specified index
- Syntax
- `X.insert(index,value)`

# EXTEND

- To append elements from *another list* to the current list, use the `extend()` method
- Syntax

```
x.extend(y)
```

#you can extend any iterable object (tuples, sets, dictionaries etc.).

# REMOVE

- Del
- Pop
- Remove
- clear

# DEL

`del x[index]` #For single item deletion

`del x[start : stop]` #For Mutiple item deletion

`del x` #For list deletion



# POP

- The `pop()` method removes the specified index.
- Syntax
- `X.pop(index)`

If you will not specify index it will remove last element from the list

# REMOVE

- The `remove()` method removes the specified item.
- Syntax
- `X.remove(value)`

# CLEAR

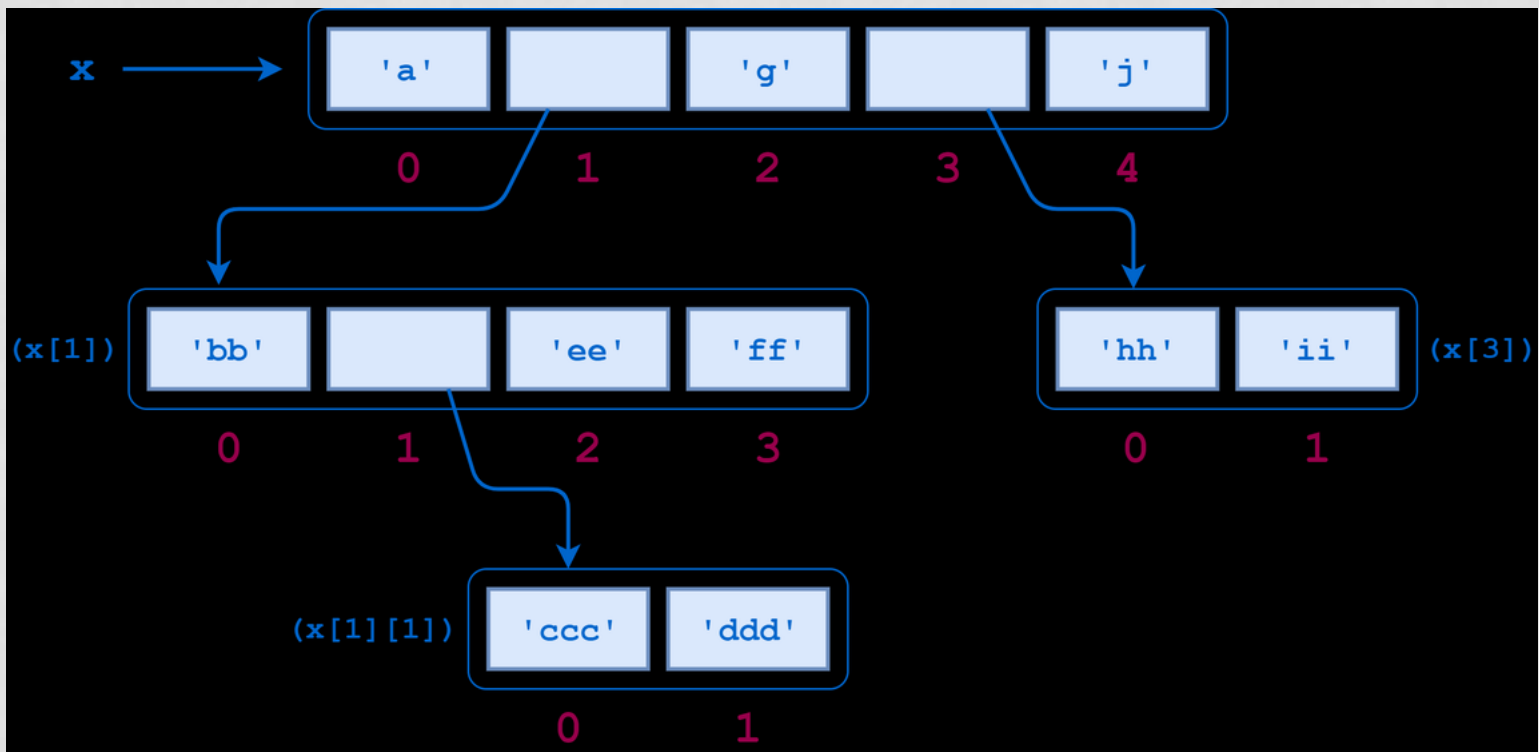
- The `clear()` method empties the list.
- The list still remains, but it has no content.

syntax

- `X.clear()`

# LIST CAN BE NESTED

- `x = ['a', ['bb', ['ccc', 'ddd'], 'ee', 'ff'], 'g', ['hh', 'ii'], 'j']`



# CONTINUE

- List use to share same copy in memory when we use = operator
- `b = colors` ## Does not copy the list

