sandeepsuryaprasad / **python_tutorials** Private

Code Issues Pull requests Actions Projects Security Ins

master python_tutorials / 2_container_objects / **_lists.py** / <> Jump to

Go to file ···

Sandeep Suryaprasad testing Latest commit 242aef0 on 19 Dec 2021 **History**

0 contributors

199 lines (154 sloc) | 7.27 KB

Raw Blame

```
from itertools import zip_longest

# PYTHON LISTS
"""
0. Objects that hold reference to the other objects are called containers
1. Lists are Mutable
2. Elements in the Lists are Ordered
3. Lists can hold duplicate elements
4. Lists can be indexed by integers starting zero
5. Lists are heterogeneous in nature. (They can point to any kind of objects)
"""

# Creating an List
my_list = []
my_list = [1, 2, 3, 4, 5]
my_list = list()    # Using list constructor
my_list = list('helloworld')
my_list = list([1, 2, 3, 4, 5])

names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'micr

print(names)    # Prints the items of the List
print(len(names))       # Prints the Length of the List. Index starts from Ze
print(names[0])     # Prints the item present in the 0th index of the List.


# Adding elements to the List
```

```
names.append('gmail')   # Adding element to the list
names.insert(3, 'watsapp')      # Inserts the item at 3rd index.

# Extends the exisitng list with the items of the new list
names.extend(['netflix', 'walmart', 'kroger'])

a = ["apple", "google", "yahoo"]
b = ["gmail", "flipkart", "facebook"]
# Merging two different lists
c = a + b
c = [*a, *b]

print('gmail' in names)     # Prints True if the item is present in the list

# Removing Items from the List
names.remove('kroger')  # Removes the item 'kroger' from the List
names.pop()     # By default this will remove the last item in the List
# pop method returns the item that it has removed from the List
names.pop(3)    # Removes the item in the 3rd index of the List

del names[0]    # Deletes 0th item in the list
# del names[3:6]  # Deletes 3rd, 4th and 5th items in the list
# del names[::2]  # Deletes alternate items in the list
# del names   # Deletes the reference to the list "names"
# del names       # Deletes the entire list

# Making copy of the list (Shallow Copy!!!)
a = [1, 2, 3, 4, 5]
b = a.copy()
  # OR
b = a[:]

# 6_Sorting List's
names.sort()    # Sorts the List in Alphabetical Order
# sort method modifies the list inplace.
names.sort(reverse=True)    # Sorts the List in Decending Order

sorted(names)   # Sorts the List in Alphabetical Order and returns a new list
# sorted method does not alter the existing list.

sorted(names, reverse=True)     # Sorts the List in Decending Order

names.index('google')       # Returns the index of the item in the List

print('yahoo' in names)    # Returns True if the item present in the List
```

```

# Iterating through the List (pythonic approach)
for item in names:
    print(item)

# Prints the item and its corresponding index in the list (Pythonic approach)
for index, item in enumerate(names):    # enumerate returns a tuple of index,
    print(index, item)

# Using range function (not preferred method)
for index in range(0, len(names)):
    print(names[index])

# Printing Index and Item using range function (not preferred method)
for index in range(0, len(names)):
    print(index, names[index])

# Printing alternate items of the list (Pythonic approach)
for name in names[::2]:
    print(name)

# Printing alternate items of the list using range function (not preferred me
for index in range(0, len(names), 2):
    print(names[index])

# Iterating over a part of the list
for item in names[:4]:
    print(item)

for index, item in enumerate(names, start=1):   # Index starts from 1
    print(index, item)

# ===========================================================================
# Iterating over multiple lists simultaniously
cities = ['Tokyo', 'Delhi', 'Shanghai', 'Sao Paulo', 'Mumbai']
population = ['38,001,000', '25,703,168', '23,740,778', '21,066,245', '21,042

# Iterating through multiple list Non-Pythonic approach
for i in range(len(cities)):
    print(cities[i], population[i])

# Iterating through multiple list using zip function
for city, population in zip(cities, population):
    print(city, population)

```

```python
# Iterating through multiple list with un-equal lengths using zip function
a = [1, 2, 3]
b = ['v', 'w', 'x', 'y', 'z']

for i in zip(a, b):
    print(i)    # Prints (1, 'v'), (2, 'w'), (3, 'x')
    # zip function stops at the shortest list

for i in zip_longest(a, b):
    print(i)    # Prints (1, 'v'), (2, 'w'), (3, 'x'), (None, y), (None, z)

for i in zip_longest(a, b, fillvalue='NA'):
    print(i)    # Prints (1, 'v'), (2, 'w'), (3, 'x'), ('NA', y), ('NA', z)

a = [1, 2, 3]
b = ['x', 'y', 'z']
c = ['alpha', 'beta', 'gamma']

for i in zip(a, b, c):
    print(i)    # Prints (1, 'x', 'alpha'), (2, 'y', 'beta'), (3, 'y', 'gamma
# ===========================================================================

files = ['youtube.txt', 'amazon.pdf', 'facebook.pdf', 'google.pdf', 'apple.do
for file in files:
    if file.endswith('pdf'):
        print(file)

# ===========OR==========
for file in files:
    if file[-3:] == 'pdf':
        print(file)

filenames = ['youtube.txt', 'amazon.pdf', 'facebook.pdf', 'google.py', 'apple
# Multiple conditions in startswith and endswith function
for filename in filenames:
    if filename.endswith(('txt', 'pdf')):   # filename either endswith txt or
        # startswith and endswith can take tuple as an argument
        print(filename)

# Converting Lists to String
print('-'.join(names))  # Prints yahoo-netflix-microsoft-instagram-google-gma
print('|'.join(names))  # Prints yahoo|netflix|microsoft|instagram|google|gma
print(','.join(names))  # Prints yahoo,netflix,microsoft,instagram,google,gma

# Slicing List's
# names[start:stop:step]
names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'micr
#       [   0         1         2       3           4           5           6
#       [  -7        -6        -5      -4          -3          -2          -1
print(names[2:5])   # Prints all the items from 2nd index upto but not includ
print(names[:4])    # Prints all items from 0th index and upto 4th index, but
print(names[2:])    # Prints all items from 2nd index till the end of the Lis

# Expression inside square brackets
print(names[1 + 3])  # Prints 4th item of the list
print(names[1 - 3])  # Prints 5th item of the list

# Slicing using negative indexing
print(names[-1])    # Prints the last index item of the list
print(names[-7])    # Prints the 0th index item of the list
print(names[-4:-2])     # Prints ['amazon', 'facebook']
print(names[-6:5])      # prints ['google', 'yahoo', 'amazon', 'facebook', 'i
print(names[1:-1])      # prints ['google', 'yahoo', 'amazon', 'facebook', 'i
print(names[:-1])   # Prints ['apple', 'google', 'yahoo', 'amazon', 'facebook

print(names[:])     # Prints the entire list
print(names[::2])   # Prints alternate items in the list
print(names[::-1])  # Prints the items in the list in reverse order

print(names[::2])   # Prints alternate items in the list
print(names[2:7:2])
print(names[-1:2:-1])
print(names[::-1])      # Prints the list in Reverse order

names[:2] = ['unknown', 'Unknown']  # Replacing Multiple items in the list
print(names)

# Print the extension of each file name in the list
files = ['youtube.txt', 'yahoo.pdf', 'microsoft.doc', 'apple.xls', 'amazon.xm
for file in files:
    print(file[-3:])
# ===========================================================================
```