

Lists in Python

Lists are one of the most commonly used data types in Python. They are mutable, ordered collections of items that can be of different types.

Creating Lists

You can create a list by enclosing items in square brackets `[]`.

```
# Empty list
empty_list = []

# List of integers
int_list = [1, 2, 3, 4, 5]

# List of strings
str_list = ["apple", "banana", "cherry"]

# Mixed type list
mixed_list = [1, "hello", 3.14, True]

print(empty_list)
print(int_list)
print(str_list)
print(mixed_list)
```

Accessing List Elements

You can access elements in a list by their index. Python uses zero-based indexing.

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

# Accessing the first element
print(fruits[0]) # Output: apple

# Accessing the last element
print(fruits[-1]) # Output: cherry

# Accessing a slice of the list
print(fruits[1:3]) # Output: ['banana', 'cherry']
```

Modifying Lists

Adding Elements

You can add elements to a list using methods like `append()`, `insert()`, and `extend()`.

```

fruits = ["apple", "banana"]

# Append adds an element at the end
fruits.append("cherry")
print(fruits) # Output: ['apple', 'banana', 'cherry']

# Insert adds an element at a specific position
fruits.insert(1, "orange")
print(fruits) # Output: ['apple', 'orange', 'banana', 'cherry']

# Extend adds multiple elements at the end
fruits.extend(["mango", "grape"])
print(fruits) # Output: ['apple', 'orange', 'banana', 'cherry', 'mango', 'grape']

```

Removing Elements

You can remove elements from a list using methods like `remove()`, `pop()`, and `clear()`.

```

fruits = ["apple", "banana", "cherry", "banana"]

# Remove removes the first occurrence of the specified value
fruits.remove("banana")
print(fruits) # Output: ['apple', 'cherry', 'banana']

# Pop removes and returns the element at the specified position
popped_fruit = fruits.pop(1)
print(popped_fruit) # Output: cherry
print(fruits) # Output: ['apple', 'banana']

# Clear removes all elements from the list
fruits.clear()
print(fruits) # Output: []

```

Common List Methods

1. `index()`

Returns the index of the first occurrence of a specified value.

```

fruits = ["apple", "banana", "cherry"]
print(fruits.index("banana")) # Output: 1

```

2. `count()`

Returns the number of occurrences of a specified value.

```
fruits = ["apple", "banana", "cherry", "banana"]
print(fruits.count("banana")) # Output: 2
```

3. sort()

Sorts the list in ascending order.

```
numbers = [3, 1, 4, 1, 5, 9]
numbers.sort()
print(numbers) # Output: [1, 1, 3, 4, 5, 9]
```

4. reverse()

Reverses the order of the list.

```
numbers = [1, 2, 3, 4, 5]
numbers.reverse()
print(numbers) # Output: [5, 4, 3, 2, 1]
```

List Comprehensions

List comprehensions provide a concise way to create lists.

```
# Creating a list of squares
squares = [x**2 for x in range(10)]
print(squares) # Output: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Nested Lists

Lists can contain other lists as elements, creating nested lists.

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(nested_list) # Output: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
# Accessing elements in a nested list
print(nested_list[0][1]) # Output: 2
```

Stay Updated

Be sure to [visit this repository](#) to stay updated with new examples and enhancements!

License

This project is protected under the MIT License.

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with by Panagiotis Moschos (<https://github.com/pmoschos>)