- re: Regular expressions

Example using math module:

python

```
import math
print(math.sqrt(16))  # Output: 4.0
```

Summary

- Modules encapsulate related Python code and definitions.
- Created by saving .py files.
- Imported in scripts to reuse code.
- Python standard library offers numerous built-in modules for commonly needed operations.
- Modular programming facilitates scalability, maintainability, and reusability.

Lists in Python

A list is a versatile, ordered, and mutable data structure in Python that can store a collection of items. Lists can contain elements of different data types including numbers, strings, and even other lists.

Creating Lists

Using square brackets with comma-separated values:

python

```python
fruits = ["apple", "banana", "cherry"]
numbers = [1, 2, 3, 4, 5]
mixed = [1, "two", 3.0, True]
```

Using the list() constructor:

python

```python
numbers = list([1, 2, 3])
chars = list("Hello")  # Converts a string to a list of characters
```

Characteristics of Lists

- Ordered: Items have a defined order and can be accessed by their index (starting from 0).
- Mutable: Items can be changed, added, or removed.
- Duplicates Allowed: Lists can contain duplicate elements.
- Indexed: Items can be accessed or modified with their index.

Accessing List Elements

Access by index:

python

```python
print(fruits[0])  # Output: apple
print(fruits[-1]) # Output: cherry (last element)
```

Modifying elements:

python

```python
fruits[1] = "blueberry"
```

Slicing lists:

python

```python
print(fruits[1:3])  # Output: ['blueberry', 'cherry']
```

Common List Methods

- append(item): Add item to the end
- insert(index, item): Insert item at specified index
- extend(iterable): Add all items from another iterable
- remove(item): Remove first occurrence of item
- pop(index): Remove and return item at index (default last)
- clear(): Remove all items
- index(item): Return index of first occurrence
- count(item): Count occurrences of item

- sort(): Sort the list
- reverse(): Reverse the list in place

Nested Lists

A list can contain other lists as elements, allowing multi-dimensional data structures like matrices.

Example:

python

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