The popular conteners and data stricter in programing language:

Python:

1. Lists: A list is a mutable collection of items of any data type.

my\_list = [1, 2, 3, 'four', 5.0]

1. Tuples: A tuple is an immutable collection of items of any data type.

my\_tuple = (1, 2, 3, 'four', 5.0)

1. Dictionaries: A dictionary is a collection of key-value pairs, where each key maps to a value.

my\_dict = {'name': 'John', 'age': 30, 'city': 'New York'}

1. Sets: A set is an unordered collection of unique items of any data type.

my\_set = {1, 2, 3, 4, 5}

R:

1. Vectors: A vector is a one-dimensional array of values of the same data type.

my\_vector <- c(1, 2, 3, 4, 5)

1. Lists: A list is a collection of objects of any data type.

my\_list <- list(1, "two", TRUE)

1. Data frames: A data frame is a two-dimensional table-like structure with rows and columns.

my\_df <- data.frame(name=c("John", "Mary", "Peter"), age=c(30, 25, 40))

C++:

1. Arrays: An array is a collection of elements of the same data type.

int my\_array[5] = {1, 2, 3, 4, 5};

1. Vectors: A vector is a collection of elements of the same data type that can change in size dynamically.

std::vector<int> my\_vector = {1, 2, 3, 4, 5};

1. Maps: A map is a collection of key-value pairs where each key maps to a value.

std::map<std::string, int> my\_map = {{"John", 30}, {"Mary", 25}, {"Peter", 40}};

C:

1. Arrays: An array is a collection of elements of the same data type.

int my\_array[5] = {1, 2, 3, 4, 5};

1. Structures: A structure is a collection of variables of different data types, grouped together under a single name.

struct person {

char name[50];

int age;

char city[50];

};

Java:

1. Arrays: An array is a collection of elements of the same data type.

int[] my\_array = {1, 2, 3, 4, 5};

1. ArrayLists: An ArrayList is a dynamic collection of elements of any data type.

ArrayList<String> my\_list = new ArrayList<String>();

my\_list.add("John");

my\_list.add("Mary");

my\_list.add("Peter");

1. Maps: A Map is a collection of key-value pairs where each key maps to a value.

Map<String, Integer> my\_map = new HashMap<String, Integer>();

my\_map.put("John", 30);

my\_map.put("Mary", 25);

my\_map.put("Peter", 40);

JavaScript:

1. Arrays: An array is a collection of elements of any data type.

let my\_array = [1, 2, 3, 'four', 5.0];

1. Objects: An object is a collection of key-value pairs where each key maps to a value.

let my\_obj = {name: "John", age: 30, city: "New York"};

1. Sets: A Set is a collection of unique elements of any data type.

let my\_set = new Set([1, 2, 3, 4, 5]);

C#:

1. Arrays: An array is a collection of elements of the same data type.

int[] my\_array = {1, 2, 3, 4, 5};

1. Lists: A List is a dynamic collection of elements of any data type.

List<string> my\_list = new List<string>();

my\_list.Add("John");

my\_list.Add("Mary");

my\_list.Add("Peter");

1. Dictionaries: A Dictionary is a collection of key-value pairs where each key maps to a value.

Dictionary<string, int> my\_dict = new Dictionary<string, int>();

my\_dict.Add("John", 30);

my\_dict.Add("Mary", 25);

my\_dict.Add("Peter", 40);

PHP:

1. Arrays: An array isa collection of elements of any data type.

$my\_array = array(1, 2, 3, 'four', 5.0);

1. Associative arrays: An associative array is a collection of key-value pairs where each key maps to a value.

scheme

$my\_assoc\_array = array('name' => 'John', 'age' => 30, 'city' => 'New York');

1. Objects: An object is an instance of a class that contains properties and methods.

php

class Person {

public $name;

public $age;

public $city;

function \_\_construct($name, $age, $city) {

$this->name = $name;

$this->age = $age;

$this->city = $city;

}

}

$my\_obj = new Person('John', 30, 'New York');