



Java HashMap

[< Previous](#)[Next >](#)

Java HashMap

In the [ArrayList](#) chapter, you learned that Arrays store items as an ordered collection, and you have to access them with an index number (`int` type). A `HashMap` however, store items in "**key/value**" pairs, and you can access them by an index of another type (e.g. a `String`).

One object is used as a key (index) to another object (value). It can store different types: `String` keys and `Integer` values, or the same type, like: `String` keys and `String` values:

Example

Create a `HashMap` object called **capitalCities** that will store `String` **keys** and `String` **values**:

```
import java.util.HashMap; // import the HashMap class

HashMap<String, String> capitalCities = new HashMap<String, String>();
```

The `HashMap` class has many useful methods. For example, to add items to it, use the `put()` method:

Example

```
// Import the HashMap class
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {
        // Create a HashMap object called capitalCities
        HashMap<String, String> capitalCities = new HashMap<String, String>();

        // Add keys and values (Country, City)
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
        System.out.println(capitalCities);
    }
}
```

Try it Yourself »

Access an Item

To access a value in the `HashMap`, use the `get()` method and refer to its key:

Example

```
capitalCities.get("England");
```

Remove an Item

To remove an item, use the `remove()` method and refer to the key:

Example

```
capitalCities.remove("England");
```

Try it Yourself »

To remove all items, use the `clear()` method:

Example

```
capitalCities.clear();
```

Try it Yourself »

ADVERTISEMENT

An advertisement for the Samsung Galaxy S21. It features a blue triangle on the left, followed by an image of the phone in orange and purple. To the right of the image, the text reads: "SAMSUNG Galaxy S21 | S21+ 5G", "Get it on Canada's fastest national 5G network.*", and "*As ranked by PCMag. Available in select major cities. bell.ca/5G." Below the text is a grey progress bar with a play button icon on the right.

SAMSUNG Galaxy S21 | S21+ 5G
Get it on Canada's fastest national 5G network.*
*As ranked by PCMag. Available in select major cities. bell.ca/5G.

HashMap Size



Example

```
capitalCities.size();
```

[Try it Yourself »](#)

Loop Through a HashMap

Loop through the items of a `HashMap` with a **for-each** loop.

Note: Use the `keySet()` method if you only want the keys, and use the `values()` method if you only want the values:

Example

```
// Print keys
for (String i : capitalCities.keySet()) {
    System.out.println(i);
}
```

[Try it Yourself »](#)

Example

```
// Print values
for (String i : capitalCities.values()) {
    System.out.println(i);
}
```

Example

```
// Print keys and values
for (String i : capitalCities.keySet()) {
    System.out.println("key: " + i + " value: " + capitalCities.get(i));
}
```

Try it Yourself »

Other Types

Keys and values in a HashMap are actually objects. In the examples above, we used objects of type "String". Remember that a String in Java is an object (not a primitive type). To use other types, such as int, you must specify an equivalent wrapper class: **Integer**. For other primitive types, use: **Boolean** for boolean, **Character** for char, **Double** for double, etc:

ADVERTISEMENT

Example

Create a **HashMap** object called **people** that will store **String keys** and **Integer values**:

```
// Import the HashMap class
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {

        // Create a HashMap object called people
        HashMap<String, Integer> people = new HashMap<String, Integer>();
    }
}
```



HTML

CSS

MORE ▼

EXERCISES ▼



```
"Steve", 30);  
"Angie", 33);  
  
    i : people.keySet()) {  
t.println("key: " + i + " value: " + people.get(i));
```

[Next >](#)

ADVERTISEMENT

[HTML](#)[CSS](#)[MORE ▼](#)[EXERCISES ▼](#)

COLOR PICKER



LIKE US



Get certified
by completing
a course today!



Get started

CODE GAME



Play Game

Certificates

HTML
CSS
JavaScript
Front End
Python
SQL
And more

[HTML](#)[CSS](#)[MORE ▼](#)[EXERCISES ▼](#)

ADVERTISEMENT

ADVERTISEMENT

[REPORT ERROR](#)[FORUM](#)[ABOUT](#)[SHOP](#)

Top Tutorials

[HTML Tutorial](#)[CSS Tutorial](#)[JavaScript Tutorial](#)[How To Tutorial](#)

[HTML](#)[CSS](#)[MORE ▼](#)[EXERCISES ▼](#)[W3.CSS Tutorial](#)[Bootstrap Tutorial](#)[PHP Tutorial](#)[Java Tutorial](#)[C++ Tutorial](#)[jQuery Tutorial](#)

Top References

[HTML Reference](#)[CSS Reference](#)[JavaScript Reference](#)[SQL Reference](#)[Python Reference](#)[W3.CSS Reference](#)[Bootstrap Reference](#)[PHP Reference](#)[HTML Colors](#)[Java Reference](#)[Angular Reference](#)[jQuery Reference](#)

Top Examples

[HTML Examples](#)[CSS Examples](#)[JavaScript Examples](#)[How To Examples](#)[SQL Examples](#)[Python Examples](#)[W3.CSS Examples](#)[Bootstrap Examples](#)[PHP Examples](#)[Java Examples](#)[XML Examples](#)[jQuery Examples](#)

Web Courses

[HTML Course](#)[CSS Course](#)[JavaScript Course](#)[Front End Course](#)[SQL Course](#)[Python Course](#)[PHP Course](#)[jQuery Course](#)[Java Course](#)[C++ Course](#)[C# Course](#)



HTML

CSS

MORE ▼

EXERCISES ▼

[Get Certified »](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2021 by Refsnes Data. All Rights Reserved.

W3Schools is Powered by W3.CSS.

