

Chapter 11: Dictionaries

A dictionary represents a collection of keys and values. See [MSDN Dictionary\(Tkey, TValue\) Class](#).

Section 11.1: Create a dictionary filled with values

```
Dim extensions As New Dictionary(Of String, String) _  
    from { { "txt", "notepad" },  
          { "bmp", "paint" },  
          { "doc", "winword" } }
```

This creates a dictionary and immediately fills it with three KeyValuePairs.

You can also add new values later on by using the Add method:

```
extensions.Add("png", "paint")
```

Note that the key (the first parameter) needs to be unique in the dictionary, otherwise an Exception will be thrown.

Section 11.2: Loop through a dictionary and print all entries

Each pair in the dictionary is an instance of KeyValuePair with the same type parameters as the Dictionary. When you loop through the dictionary with **For Each**, each iteration will give you one of the Key-Value Pairs stored in the dictionary.

```
For Each kvp As KeyValuePair(Of String, String) In currentDictionary  
    Console.WriteLine("{0}: {1}", kvp.Key, kvp.Value)  
Next
```

Section 11.3: Checking for key already in dictionary - data reduction

The ContainsKey method is the way to know if a key already exists in the Dictionary.

This comes in handy for data reduction. In the sample below, each time we encounter a new word, we add it as a key in the dictionary, else we increment the counter for this specific word.

```
Dim dic As IDictionary(Of String, Integer) = New Dictionary(Of String, Integer)  
  
Dim words As String() = Split(<big text source>, " ", -1, CompareMethod.Binary)  
  
For Each str As String In words  
    If dic.ContainsKey(str) Then  
        dic(str) += 1  
    Else  
        dic.Add(str, 1)  
    End If  
Next
```

XML reduction example : getting all the child nodes names and occurrence in a branch of an XML document

```
Dim nodes As IDictionary(Of String, Integer) = New Dictionary(Of String, Integer)  
Dim xmlsrc = New XmlDocument()  
xmlsrc.LoadXml(<any text stream source>)
```

```

For Each xn As XmlNode In xmlsrc.FirstChild.ChildNodes 'selects the proper parent
    If nodes.ContainsKey(xn.Name) Then
        nodes(xn.Name) += 1
    Else
        nodes.Add(xn.Name, 1)
    End If
Next

```

Section 11.4: Getting a dictionary value

You can get the value of an entry in the dictionary using the 'Item' property:

```

Dim extensions As New Dictionary(Of String, String) From {
    { "txt", "notepad" },
    { "bmp", "paint" },
    { "doc", "winword" }
}

Dim program As String = extensions.Item("txt") 'will be "notepad"

' alternative syntax as Item is the default property (a.k.a. indexer)
Dim program As String = extensions("txt") 'will be "notepad"

' other alternative syntax using the (rare)
' dictionary member access operator (a.k.a. bang operator)
Dim program As String = extensions!txt 'will be "notepad"

```

If the key is not present in the dictionary, a `KeyNotFoundException` will be thrown.