1. Create an HTML document that embeds an external script file and contains an element in which to write

```
<script type = "text/javascript" src = "slice.js" > </script>
<div id = "panel" > </div>
```

slice.html

2. Create the JavaScript file with an "init" function to initialize a variable with the panel element object, and to initialize an array

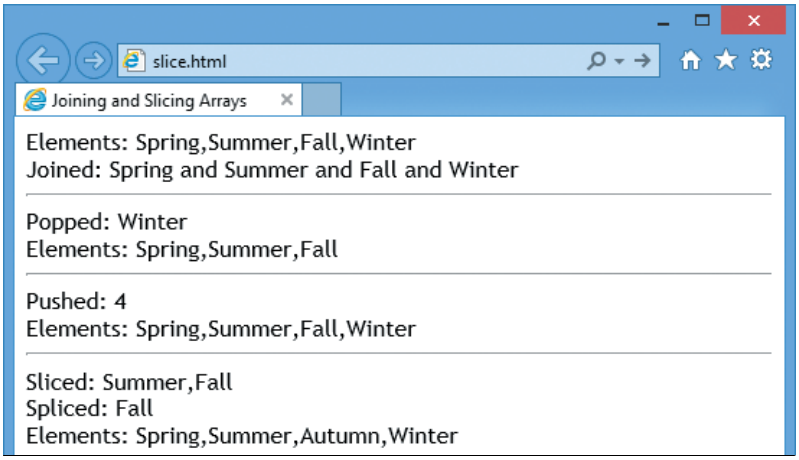
```
function init()
{
  var panel = document.getElementById( "panel" ) ;
  var seasons = [ "Spring" , "Summer" , "Fall" , "Winter" ] ;
}
document.addEventListener( "DOMContentLoaded", init, false ) ;
```

slice.js

3. In the function block, insert statements to manipulate the array elements and write the result of each task in the panel

```
panel.innerHTML = "Elements: " + seasons ;
panel.innerHTML += "<br>Joined: " + seasons.join( " and " ) ;
panel.innerHTML += "<hr>Popped: " + seasons.pop() ;
panel.innerHTML += "<br>Elements: " + seasons ;
panel.innerHTML +="<hr>Pushed: "+seasons.push("Winter");
panel.innerHTML += "<br>Elements: " + seasons ;
panel.innerHTML += "<hr>Sliced: " + seasons.slice( 1 , 3 ) ;
panel.innerHTML +=
          "<br>Spliced: " + seasons.splice( 2 , 1 , "Autumn") ;
panel.innerHTML += "<br>Elements: " + seasons ;
```

4. Save the script alongside the HTML document then open the page in a browser to see the changing array elements




**Hot tip**

Use the **splice()** method without a replacement value to delete a specified number of elements at a specified position and it will automatically renumber all remaining elements that follow in that array.

**Don't forget**

The **shift()** and **unshift()** methods work like **pop()** and **push()** but on the first element rather than the last. The **reverse()** and **sort()** methods are used in the next example listed overleaf.