**filter() method**

* The filter() method creates a new array filled with elements that pass a test provided by a function.
* The filter() method does not execute the function for empty elements.
* The filter() method does not change the original array.

**Syntax**

array.filter(function(currentValue, index, arr), thisValue)

**Parameters**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| **function()** | Required. A function to run for each array element. |
| **currentValue** | Required. The value of the current element. |
| **index** | Optional. The index of the current element. |
| **arr** | Optional. The array of the current element. |
| **thisValue** | Optional. Default undefined A value passed to the function as its this value. |

**Return Value**

|  |  |
| --- | --- |
| **Type** | **Description** |
| **Array** | An array of elements that pass the test. An empty array if no elements pass the test. |

**Example 1:**

Return the values in ages[] that are over a specific number:

<p><input type="number" id="ageToCheck" value="30"></p>  
  
<button onclick="myFunction()">Try it</button>  
  
<p id="demo"></p>  
  
<script>  
const ages = [32, 33, 12, 40];  
  
function checkAge(age) {  
  return age > document.getElementById("ageToCheck").value;  
}  
  
function myFunction() {  
  document.getElementById("demo").innerHTML = ages.filter(checkAge);  
}  
</script>

**Output:**

If I enter 11, output should be 32, 33,12,40

If I enter 16, output should be 32, 33,40

**Example 2:**

Get every element in the array that has a value of 18 or more:

<script>

const ages = [32, 33, 16, 40];

document.getElementById("demo").innerHTML = ages.filter(checkAdult);

function checkAdult(age) {

return age >= 18;

}

</script>

**Output:**

32,33,40