JavaScript Checkbox

Summary: in this tutorial, you will learn how to use JavaScript to test if a checkbox is checked, get the values of selected checkboxes, and select/unselect all checkboxes.

Creating an HTML checkbox

To create a checkbox, you use the `<input>` element with the `type` of `checkbox` as follows:

```
<input type="checkbox" id="accept"> Accept
```

It's a good practice to always associate a checkbox with a label to improve usability and accessibility. By doing this, clicking the label also checks or unchecks the checkbox.

```
<label for="accept">
   <input type="checkbox" id="accept" name="accept" value="yes">  Accept
</label>
```

Or:

```
<input type="checkbox" id="accept" name="accept" value="yes">
<label for="accept"> Accept </label>
```

Note that the for attribute's value of the label must match the id of the checkbox.

The following works but is bad practice so you should avoid it:

```
<input type="checkbox" id="accept" name="accept" value="yes"> Accept
```

Checking if a checkbox is checked

A checkbox has two states: checked and unchecked.

To get the state of a checkbox, you follow these steps:

- First, select the checkbox using a DOM method such as `getElementById()` or `querySelector()` .
- Then, access the `checked` property of the checkbox element. If its `checked` property is `true` , then the checkbox is checked; otherwise, it is not.

See the following example:

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkbox</title>
</head>
<body>
    <label for="accept">
        <input type="checkbox" id="accept" name="accept" value="yes"> Accept
    </label>

    <script>
        const cb = document.querySelector('#accept');
        console.log(cb.checked); // false
    </script>

</body>

</html>
```

In this example:

First, create the HTML checkbox element:

```
<label for="accept">
   <input type="checkbox" id="accept" name="accept" value="yes"> Accept
</label>
```

Second, select the checkbox with id `#accept` and examine the `checked` property:

```
const cb = document.querySelector('#accept');
console.log(cb.checked);
```

Because the checkbox is unchecked, you'll see `false` in the console:

```
false
```

Consider another example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkbox</title>
</head>

<body>
    <label for="accept">
        <input type="checkbox" id="accept" name="accept" value="yes"> Accept
    </label>

    <script>
        const checked = document.querySelector('#accept:checked') !== null;
        console.log(checked); // false
    </script>

</body>
</html>
```

In this example, the selector `#accept:checked` selects the element with the id `#accept` and has the attribute `checked`. For example, it matches the following element:

```
<input type="checkbox" id="accept" checked> Accept
```

But not this one:

```
<input type="checkbox" id="accept"> Accept
```

Therefore, if the checkbox element with the id `#accept` is checked, the `document.querySelector()` will return it. On the console window, you'll see the value `false` because the checkbox is unchecked:

```
false
```

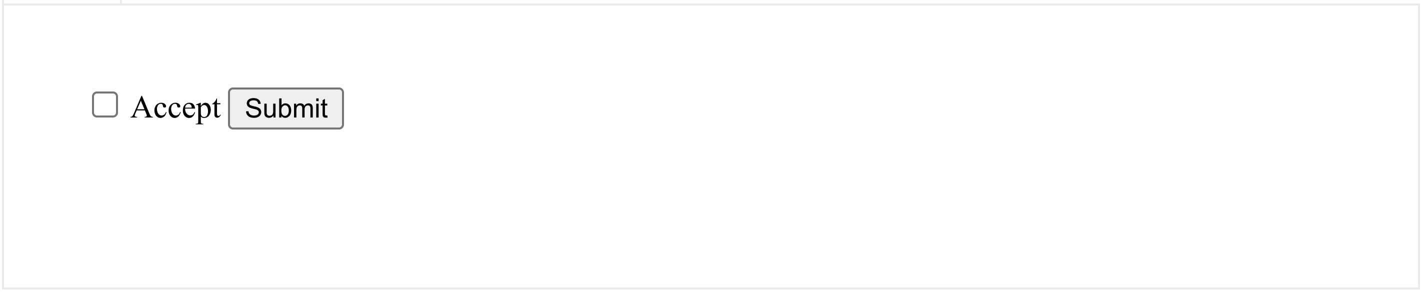
Getting checkbox values

The following page shows a checkbox and a button. When you click the button, you'll see the checkbox's value on the console window:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkbox</title>
</head>

<body>
    <label for="accept">
        <input type="checkbox" id="accept" name="accept"> Accept
    </label>

    <button id="btn">Submit</button>
    <script>
        const cb = document.querySelector('#accept');
        const btn = document.querySelector('#btn');
        btn.onclick = () => {
           alert(cb.value);
        };
    </script>
</body>

</html>
```

When you get the `value` attribute of a checkbox, you always get the `'on'` string whether the checkbox is checked or not.

Output



Getting values of multiple selected checkboxes

The following page shows three checkboxes. If you select one or more checkboxes and click the button, it'll show the values of the selected checkbox:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkboxes</title>
</head>

<body>
    <p>Select your favorite colors:</p>
    <label for="c1"> <input type="checkbox" name="color" value="red" id="c1">Red</label>
    <label for="c2"><input type="checkbox" name="color" value="green" id="c2"> Green</label>
    <label for="c3"><input type="checkbox" name="color" value="blue" id="c3">Blue</label>
    <p>
        <button id="btn">Get Selected Colors</button>
    </p>

    <script>
        const btn = document.querySelector('#btn');
        btn.addEventListener('click', (event) => {
            let checkboxes = document.querySelectorAll('input[name="color"]:checked');
            let values = [];
            checkboxes.forEach((checkbox) => {
                values.push(checkbox.value);
            });
            alert(values);
        });
    </script>

</body>

</html>
```

How it works.

In the HTML, we create three checkbox elements with the same name (color) but distinct values:

```
<label for="c1"><input type="checkbox" name="color" value="red" id="c1">Red</label>
<label for="c2"><input type="checkbox" name="color" value="green" id="c2">Green</label>
<label for="c3"><input type="checkbox" name="color" value="blue" id="c3">Blue</label>
```

In the JavaScript:

First, add the click event handler to the button:

```
const btn = document.querySelector('#btn');
btn.addEventListener('click', (event) => {
    // ...
});
```

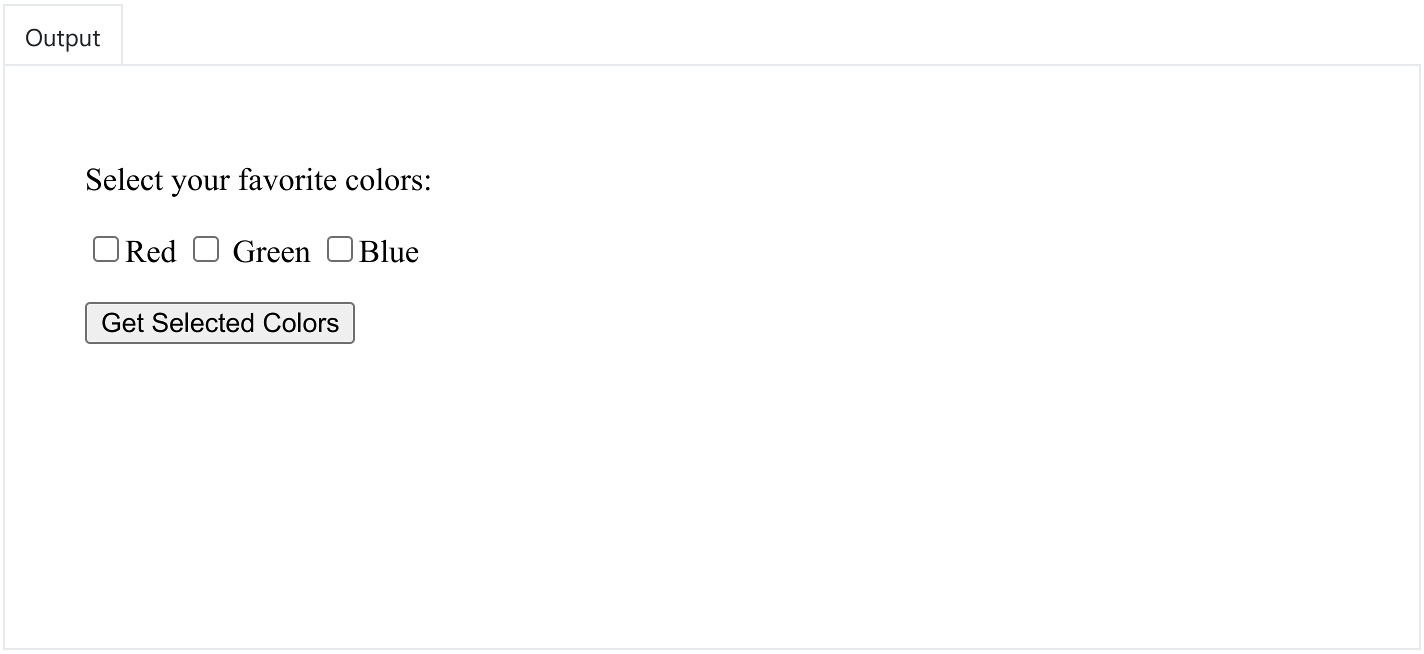
Second, select the selected checkboxes using the `document.querySelectorAll()` method inside the click event handler:

```
let checkboxes = document.querySelectorAll('input[name="color"]:checked');
```

Third, push the values of the selected checkboxes to an array:

```
let values = [];
checkboxes.forEach((checkbox) => {
    values.push(checkbox.value);
});

alert(values);
```

Demo:



Select your favorite colors:

Check / Uncheck all checkboxes

The following page has three checkboxes and a button. When you click the button, if the checkboxes are checked, they will be unchecked and vise versa:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Check/uncheck checkboxes</title>
</head>
<body>
    <p>
        <button id="btn">Check / Uncheck All</button>
    </p>
    <label for="c1"><input type="checkbox" name="color" value="red" id="c1"> Red</label>
    <label for="c2"><input type="checkbox" name="color" value="green" id="c2"> Green</label>
    <label for="c3"> <input type="checkbox" name="color" value="blue" id="c3">Blue</label>

    <script>
        function check(checked = true) {
            const checkboxes = document.querySelectorAll('input[name="color"]');
            checkboxes.forEach((checkbox) => {
                checkbox.checked = checked;
            });
        }

        function checkAll() {
            select();
            this.onclick = uncheckAll;
        }

        function uncheckAll() {
            select(false);
            this.onclick = checkAll;
        }

        const btn = document.querySelector('#btn');
        btn.onclick = checkAll;
    </script>
```

```
</body>
</html>
```

How it works:

First, define the `check()` function that checks or unchecks all checkboxes with the name `"color"` :

```
function check(checked = true) {
  const checkboxes = document.querySelectorAll('input[name="color"]');
  checkboxes.forEach((checkbox) => {
    checkbox.checked = checked;
  });
}
```

When you click the button, it checked all the checkboxes. And. If you click the button again, it should uncheck all the checkboxes. To do this switch, you need to reassign the click event handler whenever the click event fires.

Second, select the `#btn` button and assign the `checkAll()` function to the `onclick` property of the button:

```
const btn = document.querySelector('#btn');
btn.onclick = checkAll;
```

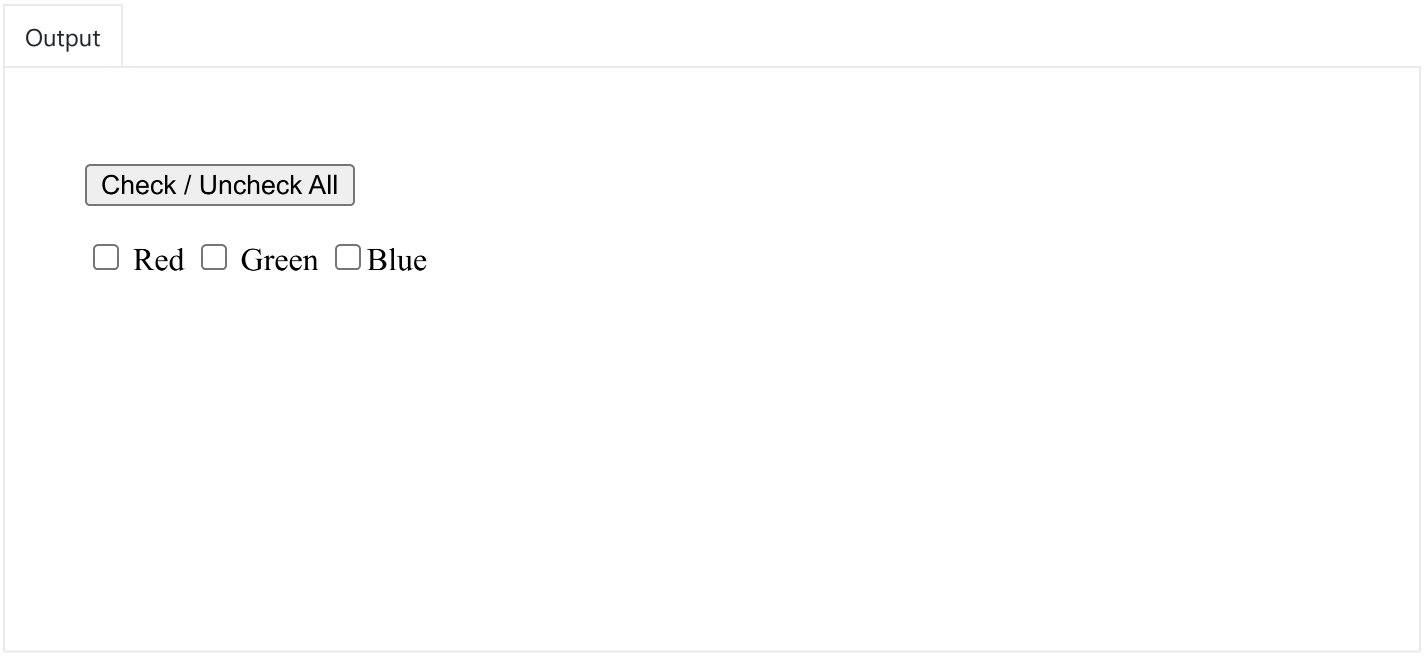
Third, define the `checkAll()` function that checks all the checkboxes:

```
function checkAll() {
  check();
  this.onclick = uncheckAll;
}
```

Finally, define the `uncheckAll()` function that unchecks all the checkboxes:

```
function uncheckAll() {
  check(false);
  this.onclick = checkAll;
}
```

Demo:



Creating checkboxes dynamically

The following example shows how to create checkboxes dynamically using JavaScript:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkboxes</title>
</head>
<body>
    <div id="root"></div>
    <script>
        const colors = ["Red","Green","Blue"];
        colors.forEach((color)=>{
            //  generate id
            const id = `color-${color}`;
            // create a label
            const label = document.createElement('label');
            label.setAttribute("for", id);

            // create a checkbox
            const checkbox = document.createElement('input');
            checkbox.type = "checkbox";
            checkbox.name = "color";
            checkbox.value = color;
            checkbox.id = id;

            // place the checkbox inside a label
            label.appendChild(checkbox);
            // create text node
            label.appendChild(document.createTextNode(color));
            // add the label to the root
            document.querySelector("#root").appendChild(label);
        });
    </script>
</body>
</html>
```

Output:

```
<div class="output-cont"><div class="output">
<iframe height="250px" src="https://www.javascripttutorial.net/sample/dom/checkbox/checkbox-
</div></div>
```

How it works.

First, define an array that consists of three strings. In practice, you may have the array that comes from the result of an API call:

```
const colors = ["Red","Green","Blue"];
```

Second, iterate over the array elements and:

1) Generate a unique id for each checkbox:

```
const id = `color-${color}`;
```

2) Create a label and assign the id to the for attribute:

```
const label = document.createElement('label');
label.setAttribute("for", id);
```

3) Create a checkbox:

```
const checkbox = document.createElement('input');
checkbox.type = "checkbox";
checkbox.name = "color";
checkbox.value = color;
checkbox.id = id;
```

4) Place the checkbox inside the label:

```
label.appendChild(checkbox);
```

5) Create a text node and append it to the label:

```
label.appendChild(document.createTextNode(color));
```

6) Add the label to the root element:

```
 document.querySelector("#root").appendChild(label);
```

The following example also dynamically generates checkboxes like the above example:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Checkboxes</title>
</head>
<body>
    <div id="root"></div>
    <script>
        const colors = ["Red","Green","Blue"];
        const html = colors.map(color => `<label for="color-${color}">
                <input type="checkbox" name="color" id="color-${color}" value="${color}"> ${c
            </label>`
        ).join(' ');
       document.querySelector("#root").innerHTML = html;
    </script>
</body>
</html>
```

In this example:

- First, generate a label and checkbox element using the Array `map()` method and template literals.
- Second, join the HTML strings into a single HTML using the String `join()` method.
- Third, append the HTML to the `#root` element.

Summary

- Use the `<input>` element with the type `checkbox` to create a checkbox element.
- Place a checkbox inside a label element to improve the usablity and accessibility.
- Use `checkbox.checked` property or `:check` selector to test if a checkbox is checked.
- Get the `value` attribute to get the value of a checkbox.