

JavaScript Radio Button

Summary: in this tutorial, you will learn how to use JavaScript to check which radio button in a radio group is checked.

Introduction to the JavaScript Radio Button

Radio buttons allow you to select only one of a predefined set of mutually exclusive options. To create a radio button, you use the `<input>` element with the type `radio`. A group of radio buttons is called a radio group.

To form a radio group, you use a common name for all the radio buttons. For example:

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

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

<body>
  <p>Select your size:</p>
  <div>
    <input type="radio" name="size" value="XS" id="xs">
    <label for="xs">XS</label>
  </div>
  <div>
    <input type="radio" name="size" value="S" id="s">
    <label for="s">S</label>
  </div>
  <div>
    <input type="radio" name="size" value="M" id="m">
```

```

        <label for="m">M</label>
    </div>
    <div>
        <input type="radio" name="size" value="L" id="l">
        <label for="l">L</label>
    </div>
    <div>
        <input type="radio" name="size" value="XL" id="xl">
        <label for="xl">XL</label>
    </div>
    <div>
        <input type="radio" name="size" value="XXL" id="xxl">
        <label for="xxl">XXL</label>
    </div>
</body>
</html>

```

In this example, all the radio buttons have the same name size but different values. Because of this, you can only select one radio button at a time.

To find the selected radio button, you follow these steps:

- Select all radio buttons by using a DOM method such as `querySelectorAll()` method.
- Get the `checked` property of the radio button. If the `checked` property is `true`, the radio button is checked; otherwise, it is unchecked.

To know which radio button is checked, you use the `value` attribute. For example:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Radio Button</title>
</head>
<body>
    <p>Select your size:</p>
    <div>

```

```
<input type="radio" name="size" value="XS" id="xs">
<label for="xs">XS</label>
</div>
<div>
  <input type="radio" name="size" value="S" id="s">
  <label for="s">S</label>
</div>
<div>
  <input type="radio" name="size" value="M" id="m">
  <label for="m">M</label>
</div>
<div>
  <input type="radio" name="size" value="L" id="l">
  <label for="l">L</label>
</div>
<div>
  <input type="radio" name="size" value="XL" id="xl">
  <label for="xl">XL</label>
</div>
<div>
  <input type="radio" name="size" value="XXL" id="xxl">
  <label for="xxl">XXL</label>
</div>
<p>
  <button id="btn">Show Selected Value</button>
</p>

<p id="output"></p>

<script>
  const btn = document.querySelector('#btn');
  const radioButtons = document.querySelectorAll('input[name="size"]');
  btn.addEventListener("click", () => {
    let selectedSize;
    for (const radioButton of radioButtons) {
      if (radioButton.checked) {
        selectedSize = radioButton.value;
        break;
      }
    }
  })
}
```

```
        // show the output:
        output.innerText = selectedSize ? `You selected ${selectedSize}` : `You haven't s
    });
</script>
</body>
</html>
```

Output

Select your size:

- ☐ XS
- ☐ S
- ☐ M
- ☐ L
- ☐ XL
- ☐ XXL

Show Selected Value

How it works.

First, select the button with `#btn` id, output element with the `#output` id, and all the radio buttons with the name `size` :

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

const radioButtons = document.querySelectorAll('input[name="size"]');
```

Second, register a click event listener on the button element:

```
btn.addEventListener('click', () => {
});
```

Third, iterate over the radio buttons and get the value of the selected radio button:

```
let selectedSize;
for (const radioButton of radioButtons) {
  if (radioButton.checked) {
    selectedSize = radioButton.value;
    break;
  }
}
```

If a radio button is checked, its `checked` property is `true`. Then, we assign the `value` of the selected radio button to the `selectedSize` variable.

Since only one radio button in a radio group can be checked at a time, the loop is terminated immediately by the `break` statement.

Finally, set the message for the output element:

```
output.innerText = selectedSize ? `You selected ${selectedSize}` : `You haven't selected any`
```

Radio button's change event

When you check or uncheck a radio button, it fires the change event. To listen to the change event, you use the `addEventListener()` method like this:

```
radioButton.addEventListener('change',function(e){

});
```

Inside the change event handler, you can access the `this` keyword to access the radio button. To check if the radio button is checked, you can use the `checked` property:

```
if(this.checked) {
  //
```

```
}
```

To get the value of the checked button, you use the value property:

```
if(this.checked) {  
    console.log(this.value);  
}
```

It'll look like this:

```
radioButton.addEventListener('change', function (e) {  
    if (this.checked) {  
        console.log(this.value);  
    }  
});
```

The following example dynamically generates a radio group and shows the selected value when a radio button is checked:

```
<!DOCTYPE html>  
<html lang="en">  
  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>JavaScript Radio Button</title>  
  </head>  
  
  <body>  
    <p>Select your size:</p>  
    <div id="group">  
      </div>  
  
    <p id="output"></p>  
  
    <script>  
      const sizes = ['XS', 'S', 'M', 'L', 'XL', 'XXL'];
```

```
// generate the radio groups
const group = document.querySelector("#group");
group.innerHTML = sizes.map((size) => `<div>
    <input type="radio" name="size" value="${size}" id="${size}">
    <label for="${size}">${size}</label>
</div>`).join(' ');

// add an event listener for the change event
const radioButtons = document.querySelectorAll('input[name="size"]');
for(const radioButton of radioButtons){
    radioButton.addEventListener('change', showSelected);
}

function showSelected(e) {
    console.log(e);
    if (this.checked) {
        document.querySelector('#output').innerText = `You selected ${this.value}`;
    }
}
</script>
</body>

</html>
```

Output

Select your size:

- ☐ XS
- ☐ S
- ☐ M
- ☐ L
- ☐ XL
- ☐ XXL

How it works.

First, define an array of strings that hold the sizes. In practice, you may get these values from a database in the back-end or from the result of an API call:

```
const sizes = ['XS', 'S', 'M', 'L', 'XL', 'XXL'];
```

Second, generate the radio groups from the elements of the sizes array:

```
const group = document.querySelector('#group');
group.innerHTML = sizes
  .map(
    (size) => `<div>
      <input type="radio" name="size" value="${size}" id="${size}">
      <label for="${size}">${size}</label>
    </div>`
  )
  .join(' ');
```

In this code:

- 1) Select the element with id #group.
- 2) Generate a radio group using the map() method with template literals; each array element corresponds to a radio button HTML.
- 3) Join radio button HTML strings into an HTML string using the join() method.
- 4) Assign the HTML to the innerHTML of the output element.

Third, select all the radio buttons with the name size and add the change event listener:

```
const radioButtons = document.querySelectorAll('input[name="size"]');
for (const radioButton of radioButtons) {
  radioButton.addEventListener('change', showSelected);
}
```


Finally, define the change event handler:

```
function showSelected(e) {  
  if (this.checked) {  
    document.querySelector('#output').innerText = `You selected ${this.value}`;  
  }  
}
```

Summary

- Use the `<input>` element with the type `radio` to create a radio button.
- Assign a name to multiple radio buttons to form a radio group. Only one radio button in the group can be selected.
- If the radio button is selected, its `checked` property is `true`.