

# createRadio()

Creates a radio button element.

The parameter is optional. If a string is passed, as in `let myRadio = createSelect(' food' )`, then each radio option will have "food" as its `name` parameter: `<input name="food">`. If an existing or `<span></span>` element is passed, as in `let myRadio = createSelect(container)`, it will become the radio button's parent element.

Radio buttons extend the `p5.Element` class with a few helpful methods for managing options:

- `myRadio.option(value, [label])` adds an option to the menu. The first parameter, `value`, is a string that sets the option's value and label. The second parameter, `label`, is optional. If provided, it sets the label displayed for the `value`. If an option with `value` already exists, its label is changed and its value is returned.
- `myRadio.value()` returns the currently-selected option's value.
- `myRadio.selected()` returns the currently-selected option.
- `myRadio.selected(value)` selects the given option and returns it as an `HTMLInputElement`.
- `myRadio.disable(shouldDisable)` Disables the radio button if `true` is passed, and enables it if `false` is passed.

## Examples

red

• yellow

• blue

▶

■

```
let style = document.createElement('style');
style.innerHTML = `
.p5-radio label {
  display: flex;
  align-items: center;
}
.p5-radio input {
  margin-right: 5px;
}
`;
document.head.appendChild(style);

let myRadio;

function setup() {
  createCanvas(100, 100);

  // Create a radio button element and place it
  // in the top-left corner.
  myRadio = createRadio();
  myRadio.position(0, 0);
  myRadio.class('p5-radio');
  myRadio.size(60);

  // Add a few color options.
  myRadio.option('red');
  myRadio.option('yellow');
  myRadio.option('blue');

  // Choose a default option.
  myRadio.selected('yellow');

  describe('A yellow square with three color options listed,
"red", "yellow", and "blue". The square changes color when the
user selects a new option.');
```

▶

■

```
let myRadio;

function setup() {
  createCanvas(100, 100);

  // Create a radio button element and place it
  // in the top-left corner.
  myRadio = createRadio();
  myRadio.position(0, 0);
  myRadio.size(50);

  // Add a few color options.
  // Color values are labeled with
  // emotions they evoke.
  myRadio.option('red', 'love');
  myRadio.option('yellow', 'joy');
  myRadio.option('blue', 'trust');

  // Choose a default option.
  myRadio.selected('yellow');

  describe('A yellow square with three options listed, "love",
"joy", and "trust". The square changes color when the user
selects a new option.');
```

▶

■

```
let myRadio;

function setup() {
  createCanvas(100, 100);

  // Create a radio button element and place it
  // in the top-left corner.
  myRadio = createRadio();
  myRadio.position(0, 0);
  myRadio.class('p5-radio');
  myRadio.size(50);

  // Add a few color options.
  myRadio.option('red');
  myRadio.option('yellow');
  myRadio.option('blue');

  // Choose a default option.
  myRadio.selected('yellow');

  // Create a button and place it beneath the canvas.
  let btn = createButton('disable');
  btn.position(0, 100);

  // Call disableRadio() when btn is pressed.
  btn.mousePressed(disableRadio);

  describe('A yellow square with three options listed, "red",
"yellow", and "blue". The square changes color when the user
selects a new option. A "disable" button beneath the canvas
disables the color options when pressed.');
```

## Syntax

- `createRadio([containerElement])`
- `createRadio([name])`
- `createRadio()`

## Parameters

`containerElement` Object: container HTML Element, either a `<div></div>` or `<span></span>`.  
`name` String: name parameter assigned to each option's `<input></input>` element.

## Returns

`p5.Element`: new `p5.Element` object.

This page is generated from the comments in `src/dom/dom.js` . Please feel free to edit it and submit a pull request!

## Related References

<code>addClass</code> Adds a class to the element.	<code>attribute</code> Adds an attribute to the element.	<code>center</code> Centers the element either vertically, horizontally, or both.	<code>child</code> Attaches the element as a child of another element.
---	---	--	---