

selectAll()

Searches the page for all elements that matches the given [CSS selector string](#).

The selector string can be an ID, class, tag name, or a combination. `selectAll()` returns an array of [p5.Element](#) objects if it finds any matches and an empty array if none are found.

The second parameter, `container`, is optional. It specifies a container to search within. `container` can be CSS selector string, a [p5.Element](#) object, or an [HTMLElement](#) object.

[Reference](#)

DOM

[addClass\(\)](#)
[attribute\(\)](#)
[center\(\)](#)
[child\(\)](#)
[class\(\)](#)
[doubleClicked\(\)](#)
[dragLeave\(\)](#)
[dragOver\(\)](#)
[draggable\(\)](#)
[drop\(\)](#)
[elt](#)
[hasClass\(\)](#)
[height\(\)](#)
[hide\(\)](#)
[html\(\)](#)
[id\(\)](#)
[mouseClicked\(\)](#)
[mouseMoved\(\)](#)
[mouseOut\(\)](#)

Examples

1

▶

■

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

  // Create three buttons.
  createButton('1');
  createButton('2');
  createButton('3');

  // Select the buttons by their tag.
  let buttons = selectAll('button');

  // Position the buttons.
  for (let i = 0; i < 3; i += 1) {
    buttons[i].position(0, i * 30);
  }

  describe('Three buttons stacked vertically. The buttons are labeled, "1", "2", and "3".');
}
```

□

⟳

2

▶

■

```
function setup() {
  // Create three buttons and position them.
  let b1 = createButton('1');
  b1.position(0, 0);
  let b2 = createButton('2');
  b2.position(0, 30);
  let b3 = createButton('3');
  b3.position(0, 60);

  // Add a class attribute to each button.
  b1.class('btn');
  b2.class('btn btn-pink');
  b3.class('btn');

  // Select the buttons by their class.
  let buttons = selectAll('.btn');
  let pinkButtons = selectAll('.btn-pink');

  // Style the selected buttons.
  buttons.forEachsetFont);
  pinkButtons.forEachsetColor);

  describe('Three buttons stacked vertically. The buttons are labeled, "1", "2", and "3". Buttons "1" and "3" are gray. Button "2" is pink.');
}
```

□

⟳

Syntax

```
selectAll(selectors, [container])
```

□

Parameters

`selectors` String: CSS selector string of element to search for.
`container` String|[p5.Element](#)|[HTMLElement](#): CSS selector string, [p5.Element](#), or [HTMLElement](#) to search within.

Returns

[p5.Element\[\]](#): array of [p5.Element](#)s containing any elements found.

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

[addClass](#)

Adds a class to the element.

[attribute](#)

Adds an attribute to the element.

[center](#)

Centers the element either vertically, horizontally, or both.

[child](#)

Attaches the element as a child of another element.



Donate Today! Support p5.js and the Processing Foundation.

×

Resources

[Reference](#)

[Tutorials](#)

[Examples](#)

[Contribute](#)

[Community](#)

[About](#)

[Start Coding](#)

[Donate](#)

Information

[Download](#)

[Contact](#)

[Copyright](#)

[Privacy Policy](#)

[Terms of Use](#)

Socials

[GitHub](#) ↗

[Instagram](#) ↗

[X](#) ↗

[YouTube](#) ↗

[Discord](#) ↗

[Forum](#) ↗