Resizing with CSS

There are times when it would be really nice if we could resize a piece of content on a page without having to use Javascript.

That’s where the resize property comes in handy. Using the property in an element tell the browser what direction the element can be resized on.

The most basic example of a resizable element looks like this:

```css
.container {
  height: 300px;
  width: 300px;
  resize: both;
  overflow: auto;
}
```

The possible values for resize are:

none : The element offers no user-controllable method for resizing it. This is the default value if it’s not present

`both` : The element displays a mechanism for allowing the user to resize it, which may be resized both horizontally and vertically.

`horizontal` : The element displays a mechanism for allowing the user to resize it in the horizontal direction.

`vertical` : The element displays a mechanism for allowing the user to resize it in the vertical direction.

`block` : The element displays a mechanism for allowing the user to resize it in the block direction (either horizontally or vertically, depending on the `writing-mode` and `direction` value).

`inline` The element displays a mechanism for allowing the user to resize it in the inline direction (either horizontally or vertically, depending on the `writing-mode` and `direction` value).

A few examples of where resize would work well

- Creating demos for container queries
- Using textarea boxes for data entry or as the UI for an editor

Controlling minimum and maximum sizes

By default resizing is unconstrained, meaning that there is no minimum or maximum size.

To establish constraints on the container set min-width/min-height to control the minimum dimensions and/or max-width/max-height to control the maximum dimensions.

This will prevent the resizable box from becoming too small/big.

A constrained exaple of a resizable box looks like this:

```css
.container {
  height: 300px;
  width: 300px;
  resize: both;
  overflow: auto;

  min-width: 100px;
  max-width: 500px;

  min-height: 100px;
  max-height: 500px;
}
```

A full working example of a constrained resizable element: