

CSS Display Types

[Introduction](#)

[Block and inline HTML tags](#)

[The difference between inline and block elements](#)

[Editing inline elements](#)

[Using inline-block in Divi](#)

[Action Items](#)

Introduction

Welcome to the lesson about the different display types in CSS. Every element on a web page is a rectangular box. The display property in CSS determines just how that rectangular box behaves.

There are various values of the display property. In this lesson, I will be covering the display inline, block, and inline-block.

Block and inline HTML tags

So first, we have to realize that some HTML tags like `<div>`, `<p>`, `` take the full width of space and each starts with a new line, whereas other HTML tags like ``, `` or `<a>` don't need a new line and can be placed side by side. This is because of the different display behaviors: block or inline.

Block

```
<div>
<h1>
<p>
<ul>
```

Inline

```
<span> <a> <img>
```

Let's see the difference with a short example.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Website</title>
5   <link rel="stylesheet" href="style.css">
6 </head>
7
8 <body>
9 <div id="main-content">
10
11   <p>Paragraph</p>
12   <p>Paragraph</p>
13   <p>Paragraph</p>
14
15   <a href="#">Link</a>
16   <a href="#">Link</a>
17   <a href="#">Link</a>
18
19   <!--
20   <p>Lorem ipsum dolor sit amet proin potenti tortor pharetra
21   finibus. Sample <a href="#">Link</a> torquent sollicitudin dapibus
22   vehicula. Nullam himenaeos taciti mollis montes sit fermentum.</p>
23   -->
24 </div>
25 </body>
26 </html>

```

Paragraph

Paragraph

Paragraph

[Link Link Link](#)

```
<div id="main-content">
```

```
<p>Paragraph</p>
```

```
<p>Paragraph</p>
```

```
<p>Paragraph</p>
```

```
<a href="#">Link</a>
```

```
<a href="#">Link</a>
```

```
<a href="#">Link</a>
```

```
</div>
```

Without any CSS, I will create an HTML template with `<p>` and `<a>` tags. I'm sure you can see the difference. Each `<p>` tag starts with a new line even if there is enough space for them to fit next to each other. Links, however, are displayed side by side.

The difference between inline and block elements

Block-level elements take full-width (100 percent width) by default, while inline elements take only as much space as they need. Each block element is displayed in a new line - and inline elements are displayed side by side. For a block element, width and height properties can be set, whereas inline elements don't accept width or height and top and bottom margin.

And block elements can contain other block or inline elements and an inline element can be a parent of other inline elements, but not block elements.

Block

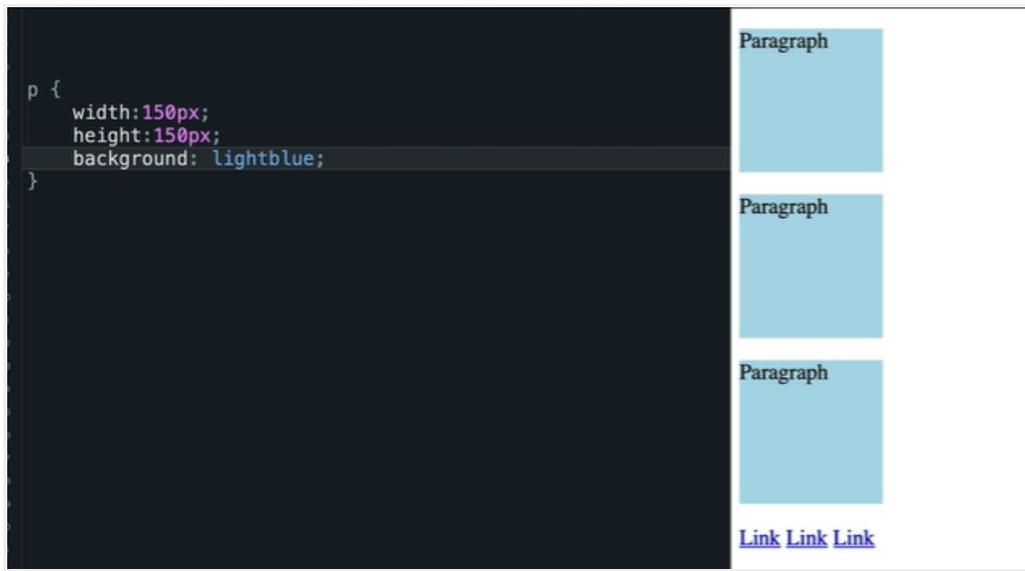
- ✔ Take full-width (100% width) by default
- ✔ Each gets displayed in a new line
- ✔ Width and height properties can be set
- ✔ Can contain other block or inline elements

Inline

- ✔ Take only as much space as they need
- ✔ Displayed side by side
- ✔ Don't accept width or height, or top-bottom margin
- ✔ Can be a parent of other inline elements only

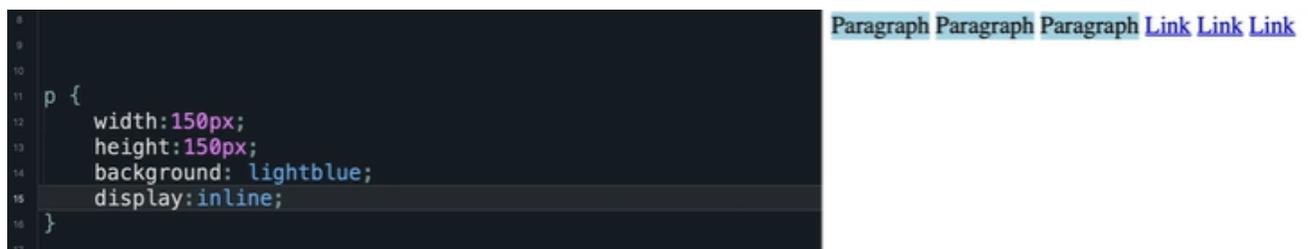
Okay, so now since we know those block elements can have width and height set, let's try changing the width of these <p> tags. Let's say "p" width: 150 pixels and height: 150 pixels, and let's change the background color so we can see it better.

```
p {width: 150px; height: 150px; background: lightblue;}
```



So now the width is set to 150 pixels but they still are displayed in a new line. What if I change the display type to inline?

```
p {display: inline}
```



My "p" tags will be displayed side by side but my width and height are not taking effect. That is when `display: inline-block` comes useful.



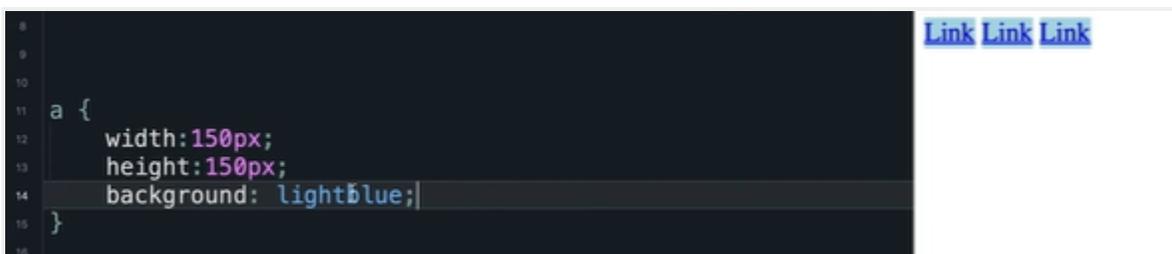
Whenever we want to place two block elements side by side and apply the height, width, or top and bottom padding and margin, we can set the display to inline-block. And it also works the other way around, so when we have an inline element and we want to keep it in line but we want to control the sizing better, we also can change the display to inline-block as well.

Let me show you. Let's hide these paragraphs for a moment, and let me show you how those links look if we change the styling a little bit.

Editing inline elements

So for an "a" element - if they are in line, the width and height cannot be set.

```
a {width: 150px; height: 150px; background: lightblue;}
```



And let's try changing the margin by adding another declaration: `margin-top: 30px;` Nothing happens - margin-top and bottom doesn't work on inline elements. But what about margin-right? Let's add: `margin-right: 50px;` Margin right does take effect.

```

8
9
10
11 a {
12   width:150px;
13   height:150px;
14   background: lightblue;
15   margin-top:30px;
16   margin-right:50px;
17 }
18

```

And let's remove that margin for a moment, let's try adding a top padding, so `padding-top: 30px;` and that seems to work. And `padding-right: 50px;` and that works too.

```

8
9
10
11 a {
12   width:150px;
13   height:150px;
14   background: lightblue;
15   margin-top:30px;
16   padding-top:30px;
17   padding-right:50px
18 }
19

```

Now let's add a bottom padding as well: `padding-bottom: 30px;` - it does apply but I will show you how such a link will behave inside a paragraph.

So let's hide these links, and I am adding comments, for example

`<!--Link-->` to hide the HTML. And we have a link inside a paragraph, and you see that padding is applied but it doesn't affect the other elements, it's still inline.

```

<p>Lorem ipsum dolor sit amet proin potenti tortor pharetra finibus. Sample <a href="#">Link</a> torquent sollicitudin dapibus vehicula. Nullam himenaeos taciti mollis montes sit fermentum.</p>

```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Website</title>
5   <link rel="stylesheet" href="style.css">
6 </head>
7
8 <body>
9 <div id="main-content">
10
11   <!--<p>Paragraph</p>
12   <p>Paragraph</p>
13   <p>Paragraph</p-->
14
15   <!--<a href="#">Link</a>
16   <a href="#">Link</a>
17   <a href="#">Link</a-->
18
19   <p>Lorem ipsum dolor sit amet proin potenti tortor pharetra
20   finibus. Sample <a href="#">Link</a> torquent sollicitudin dapibus
21   vehicula. Nullam himenaeos taciti mollis montes sit fermentum.</p>
22
23 </div>
24
25 </body>
26
27 </html>
28

```

Lorem ipsum dolor sit amet proin potenti tortor pharetra finibus.
 Sample [Link](#) torquent sollicitudin dapibus vehicula. Nullam
 himenaeos taciti mollis montes sit fermentum.

So to fix that I would also add `display: inline-block` to the declarations affecting the “a” elements, and that will make my element accept my width and height.

```

8
9
10
11 a {
12   width: 150px;
13   height: 150px;
14   background: lightblue;
15   margin-top: 30px;
16   padding-top: 30px;
17   padding-bottom: 30px;
18   padding-right: 50px;
19   display: inline-block;
20 }
21
22

```

Lorem ipsum dolor sit amet proin potenti tortor pharetra finibus.
 Sample [Link](#) torquent sollicitudin dapibus vehicula.
 Nullam himenaeos taciti mollis montes sit fermentum.

But if width and height weren't set, it would at least consider the padding and margin and it would affect the other elements on the page.

```

8
9
10
11 a {
12
13   background: lightblue;
14   padding-top: 30px;
15   padding-bottom: 30px;
16   padding-right: 50px;
17   display: inline-block;
18 }
19

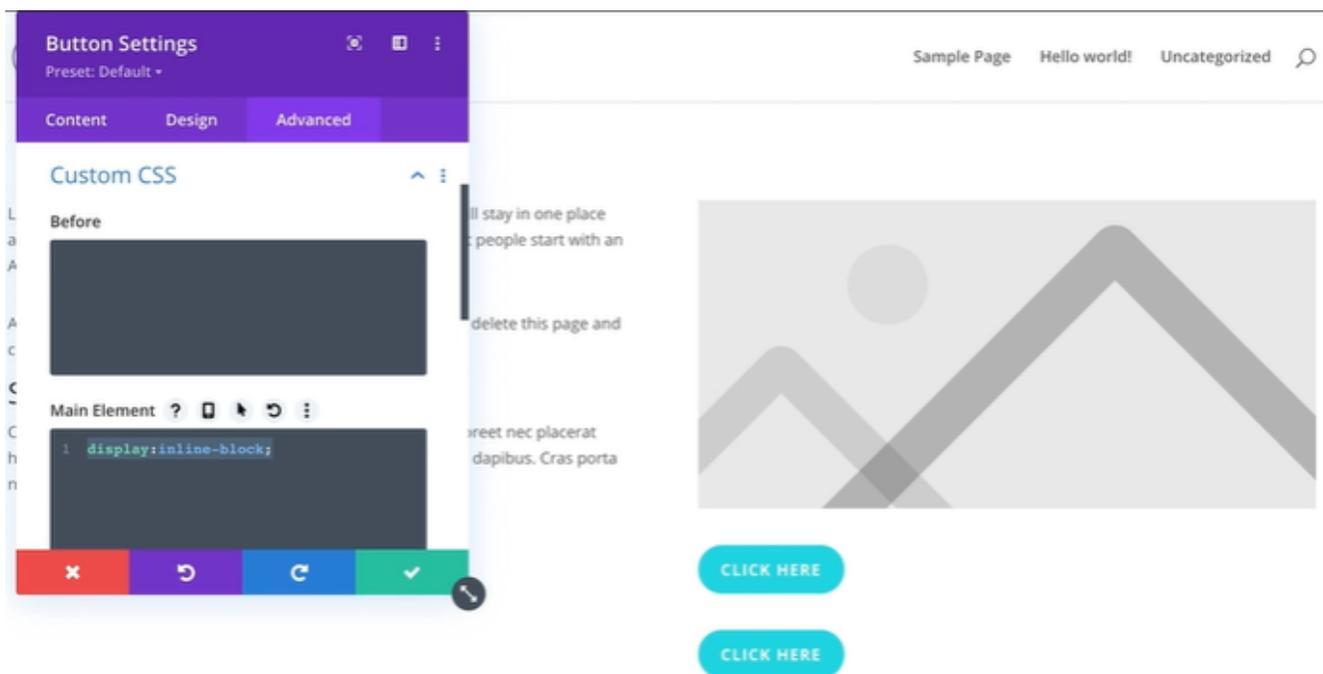
```

Lorem ipsum dolor sit amet proin potenti tortor pharetra finibus.
 Sample [Link](#) torquent sollicitudin dapibus vehicula. Nullam
 himenaeos taciti mollis montes sit fermentum.

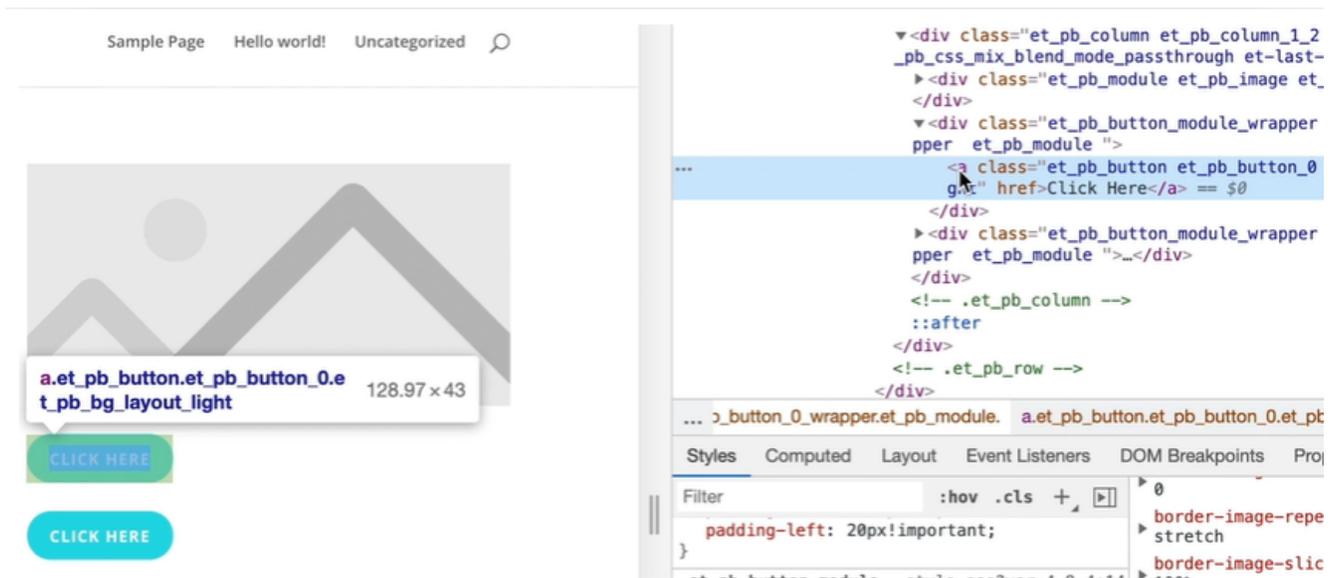
So display inline-block is very useful, and the most common use case for Divi is, for example, placing two button modules (or any modules really) side-by-side.

Using inline-block in Divi

So let me show you. I have a simple layout here and let's try duplicating that Button module and let's try setting the display property to display inline-block so that they are next to each other, okay?



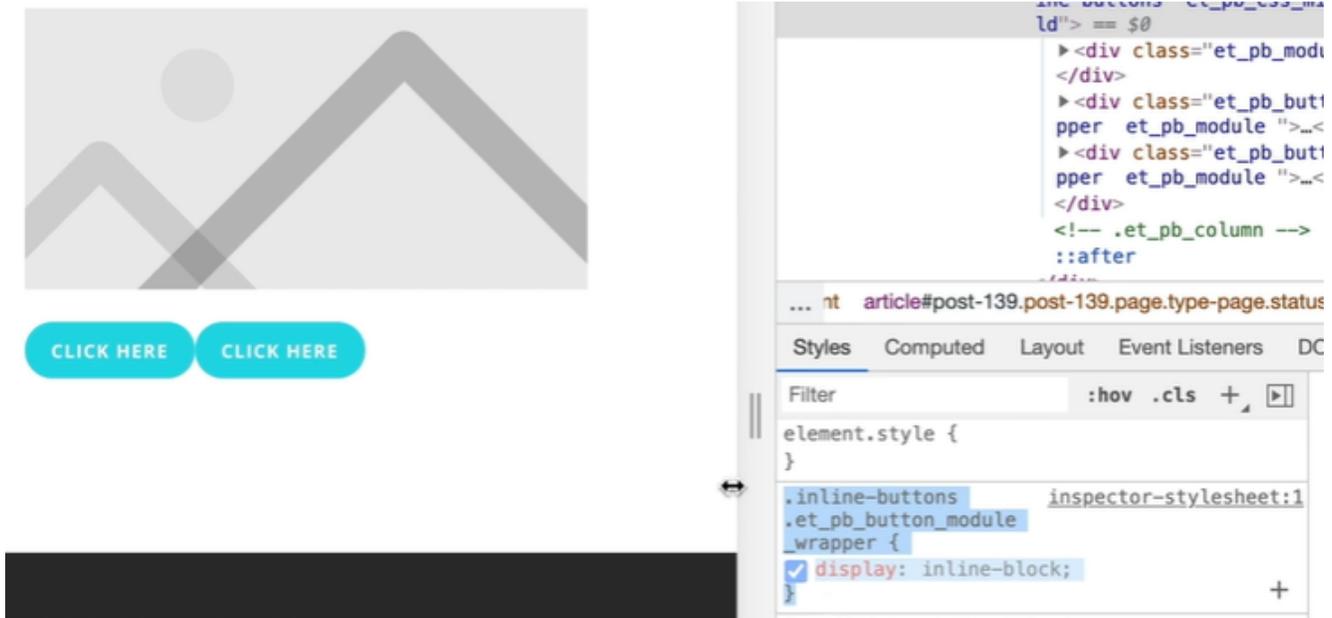
I will show you why this won't work here and how to fix this. So in the first button's settings -> Advanced -> Custom CSS for the main element, we want to add `display:inline-block;` And if I even apply that to extend that styling to the second button, we won't see any change, so let me save that and let's see how that looks in our browser inspector.



The button module is a specific module that really has an additional div wrapper. So the styles we've added for the display is for the a-link, which is inside the div here. That's why setting display inline-block for an "a" element doesn't do anything, because this div is still displayed as block by default, and it's taking the full width. So we have to target that outside div, that parent div, but it doesn't have any custom CSS class we could target.

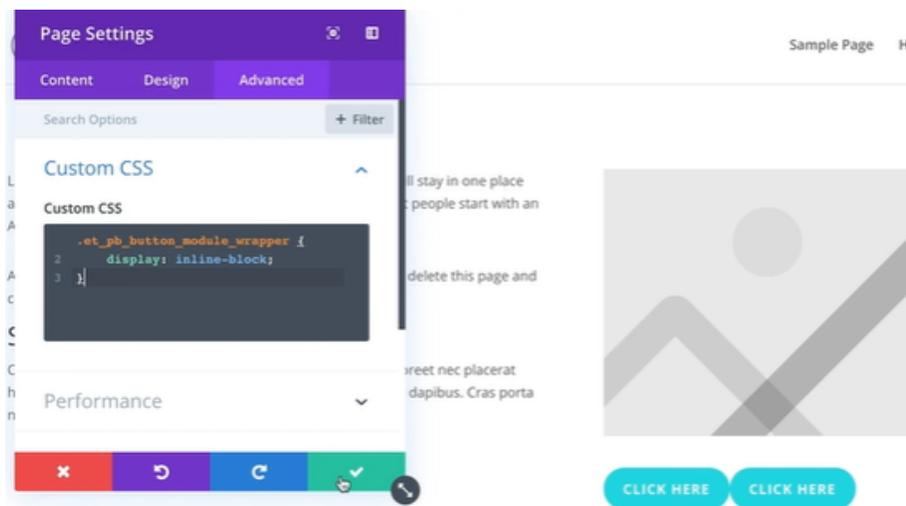
If we add a CSS class to the module, it would be added to the "a" element, so we can work around it and add a custom class to our column.

Let me show you: here in the row settings -> Second Column -> Advanced and let's say we'll add a custom class "inline-buttons". Okay, let's save that. And now if I refresh here, my column has a class of "inline-buttons" and now I can target the wrapper div which has a class "et_pb_button_module_wrapper".

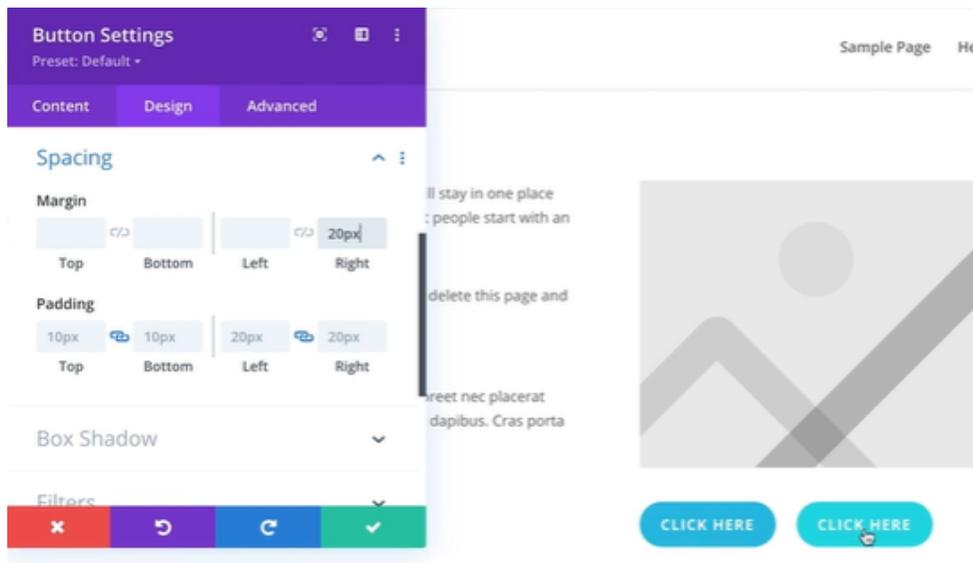


```
.inline-buttons .et_pb_button_module_wrapper {display: inline-block;}
```

So I'm targeting all button module wrappers inside the "inline-buttons" CSS class. Now I could copy that CSS and add it to my page's custom CSS for example.



And I can change the margin in the spacing section margin right at 20 pixels for example.



And I have beautiful side-by-side buttons just by understanding the HTML structure and the display property.

So I hope that was helpful.

Action Items

- Use the technique explained in the lesson to add two Button modules next to each other in Divi.