

PART 3: WEBFLOW

# ***BUILDING YOUR FIRST WEB PAGE***

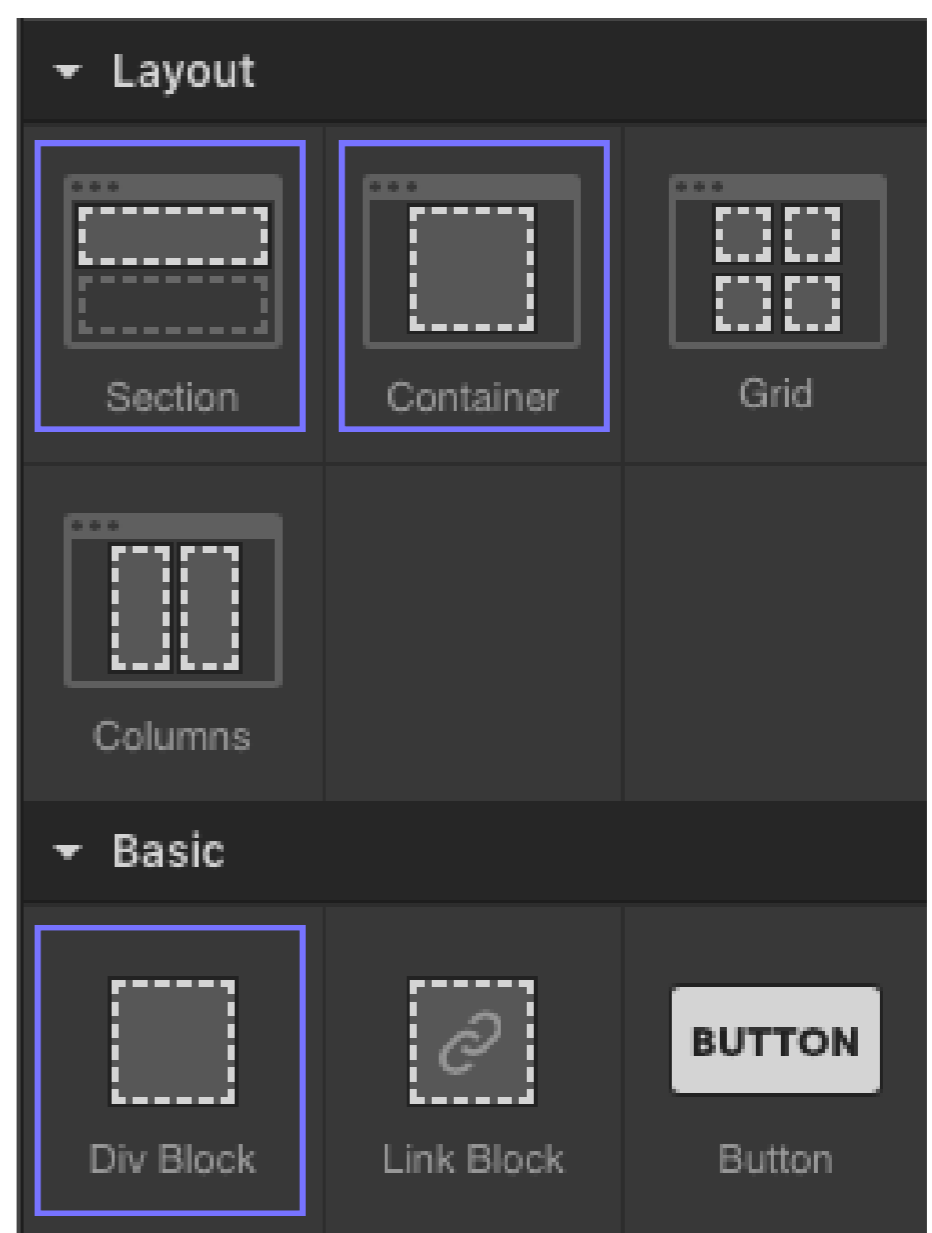
## Section, Container, Div Block

These are the building blocks of a page. We add a **Section**, then we add (nest) a **Container** inside it.

A container holds our content in the center. If we don't add a container then everything we put inside a section is going to be glued to the edges. Doesn't look good.

For everything else, we use Div Blocks.

In reality, sections and containers are div blocks too. If you check the code export, you will see `<div>` tags in places of sections and containers.



## CONTAINER

A container is super important for the layout of your page.

You can either add a default container from the elements panel or you can create your own. I advise creating your own since the default one isn't wide enough.

A container needs to have some maximum width so it's not stretched the edge to edge like a section and needs to be centered horizontally so it's not glued to the left edge.

**To create your custom container**, add a div block inside a section, and apply these two styles:

- 1. Max-width:** 1160px or any other value you fancy.
- 2. Margin:** Auto on both sides, left and right. There's a centering button in the corner that automatically applies auto margins.



**Common Mistake:** Often students forget to use a container with correct settings or completely forget to use it. This creates a lot of complications down the line. So don't forget to **USE THE CONTAINER!**

## Size Settings

In Webflow there are 6 different size settings for every element. Two of them are Width and Height.

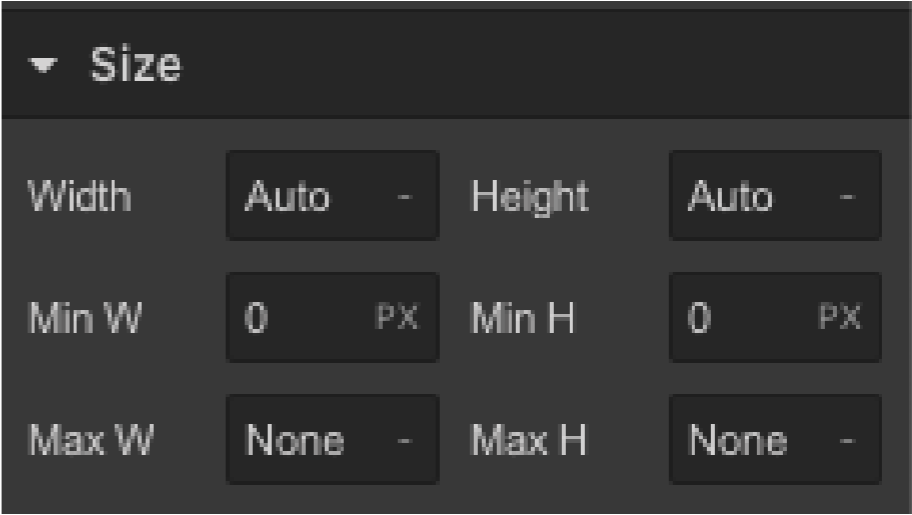
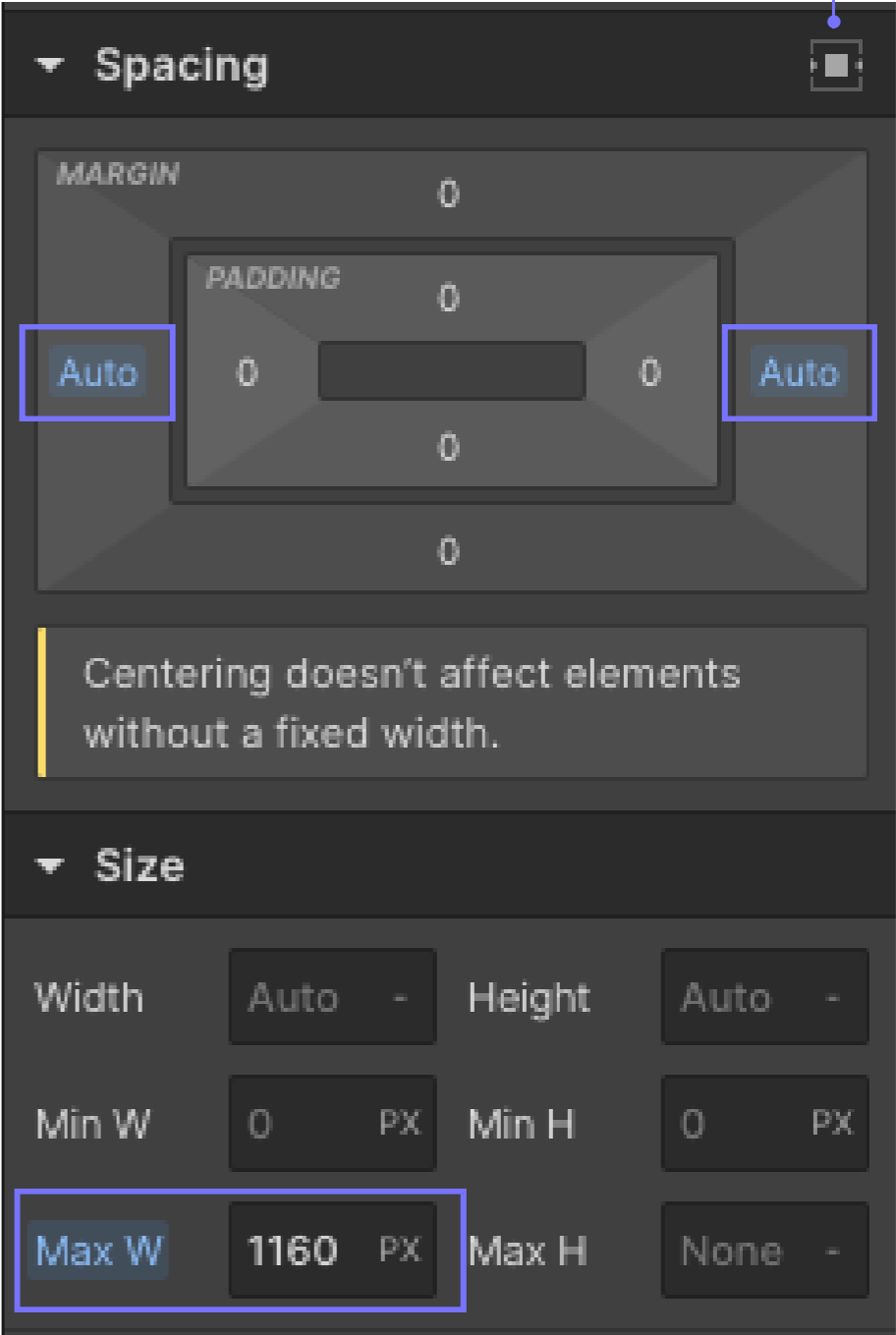
And we have four **Minimums** and **Maximums**:

- Min Width
- Max Width
- Min Height
- Max Height

Minimums and Maximums set floors and ceilings on the elements.

### Center

Centers an element horizontally by applying Auto margins on the sides. The element has to have a fixed or max-width.



## SIZE UNITS

**PX (Pixels)** - relative the screen resolution

**% (Percentage)** – relative to the parent element

**EM** - relative to the font size of the parent element

**VH** - relative to the viewport (browser) height. The viewport height is 100vh.

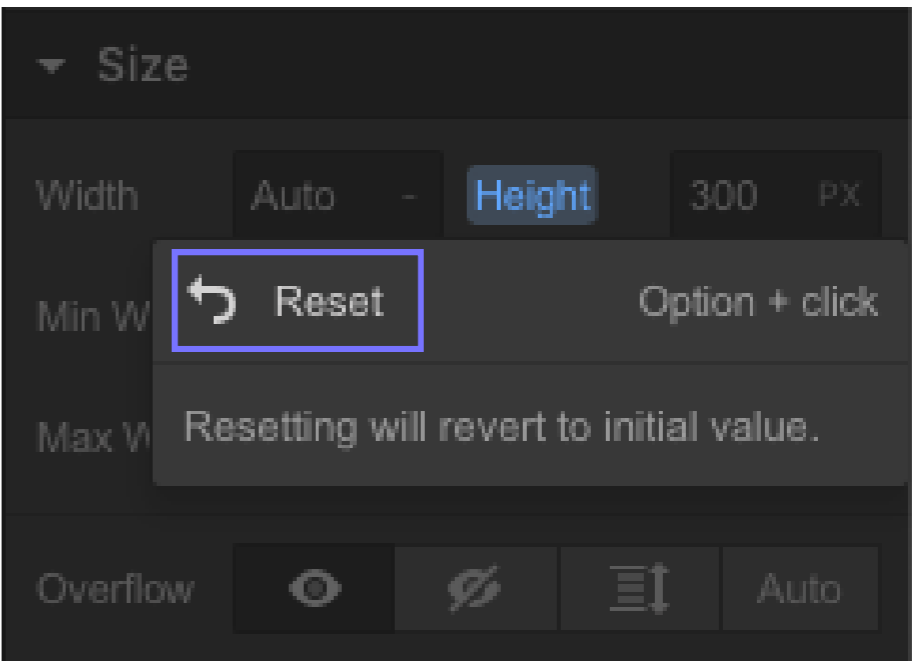
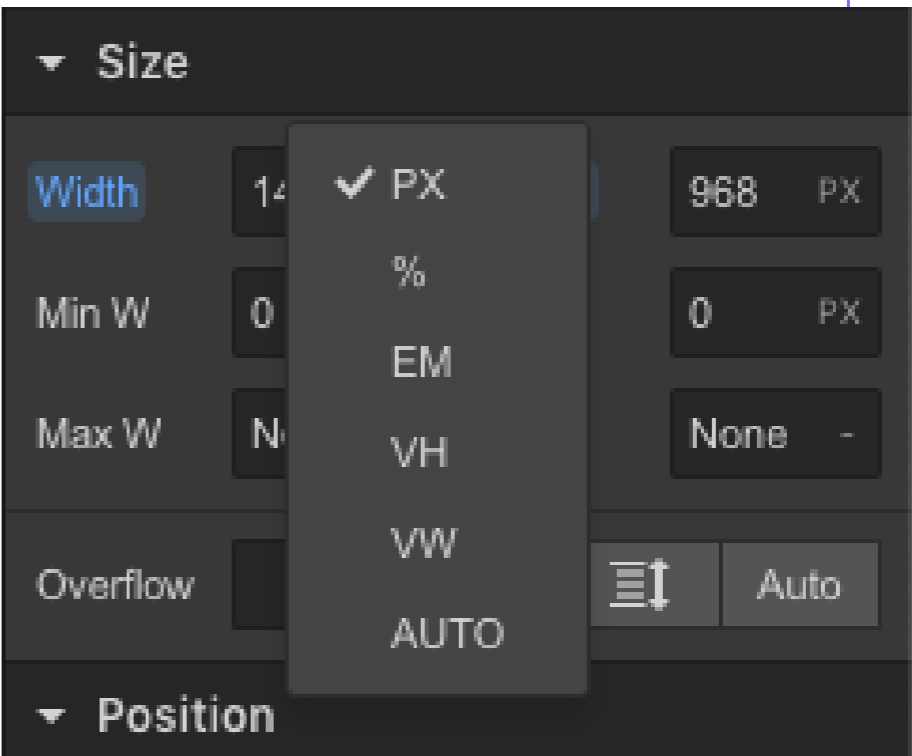
**VW** - relative to the viewport (browser) width. The viewport width is 100vw.

**AUTO** - the element will take the full width of its parent or shrink/grow to take the size of its content.



**TIP:** When returning any style to the default value, instead of manually putting that default value, **RESET** it.

This keeps the site’s source code cleaner and avoids potential bugs.



## Padding & Margin

Padding is spacing inside of an element. And Margin is spacing on the outside.

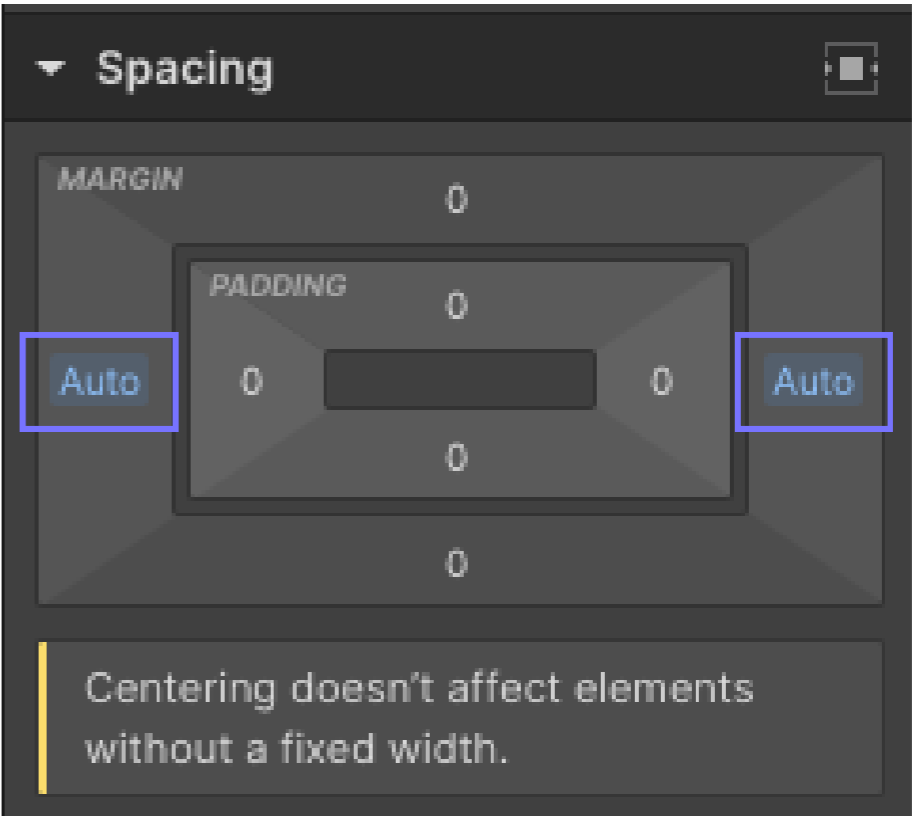
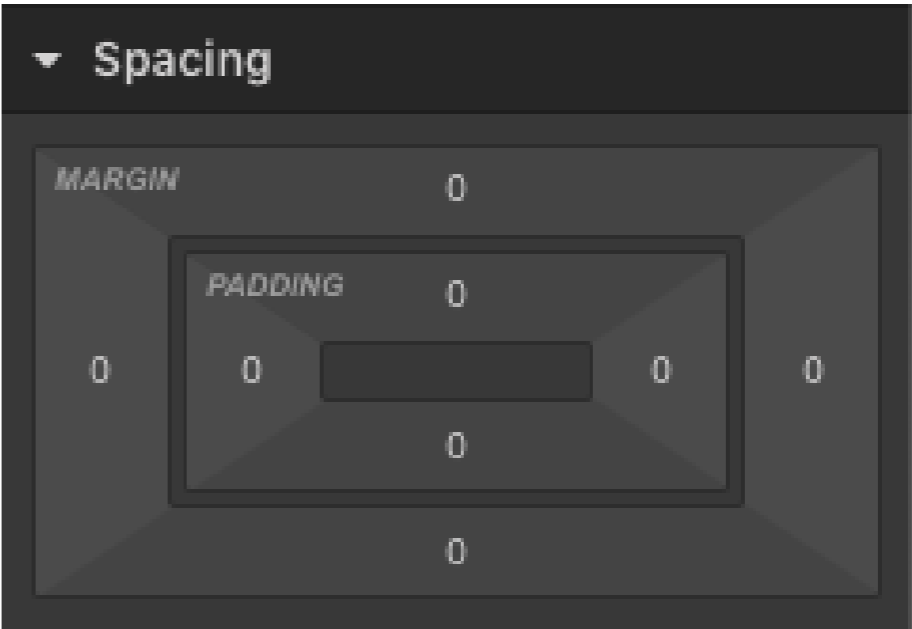
### CENTERING WITH AUTO MARGIN

By applying Auto margins on both sides of an element, you can center it horizontally within its container.

**Note**, the element needs to have a width or max width.



**Common Mistake:** If an element is as wide as the parent, then you’re not going to see centering take effect. Because technically, the element is already centered.

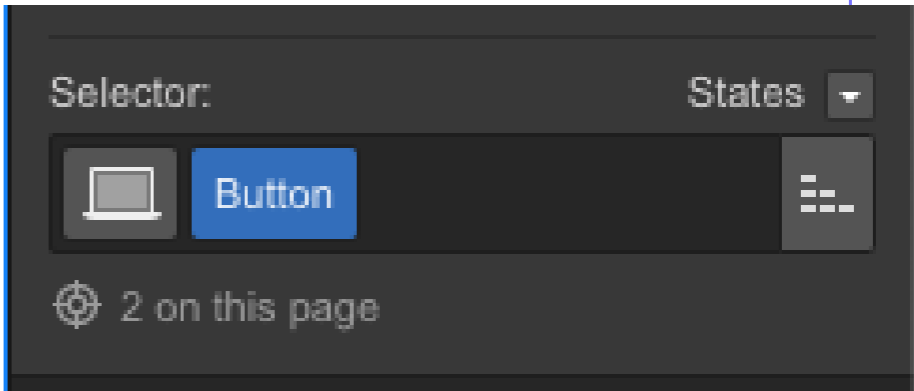


# CSS Classes

Classes save styling information that you can apply to as many elements as you want throughout your project.

The moment you begin styling an element, a class is **automatically created** and applied to the selected element. All styling adjustments made on this element are saved in this class.

You can rename a class and you can also create a class manually before adding styles by typing in a class name in the **selector**.

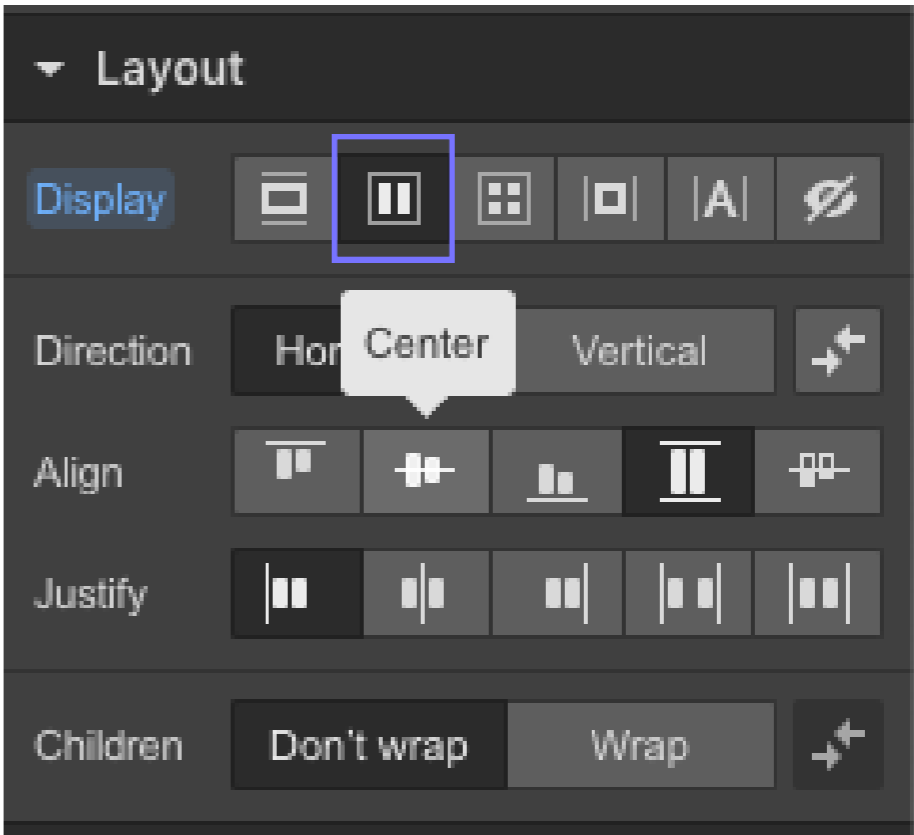


# Flexbox

Flexbox is a way to arrange and distribute content inside that element.


Flex settings affect only the **children**, not the element itself.

By default, flex children will always try to fit on a single line. You can change that by wrapping the children in the settings.

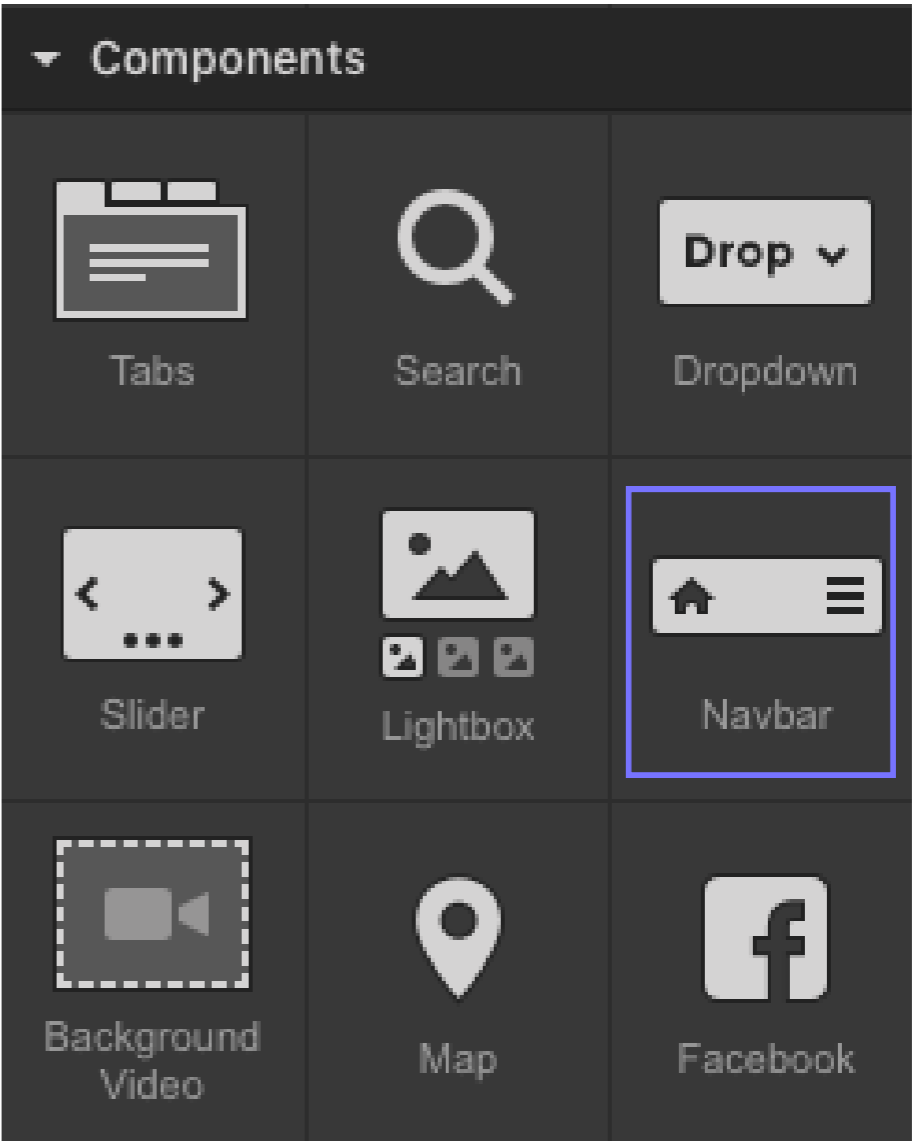


# Navbar

Navbar is a premade Webflow component. It already comes with useful elements inside and a dropdown menu for the mobile screens.

**TIP:** The content inside a navbar is nested in the Container. The Max-width of this container most likely is going to be different than the custom *'container'* class that you've created. This will show **navbar misaligned** with the rest of your page content.

To fix this, just apply the same *'container'* class to the navbar container.



# Hover State

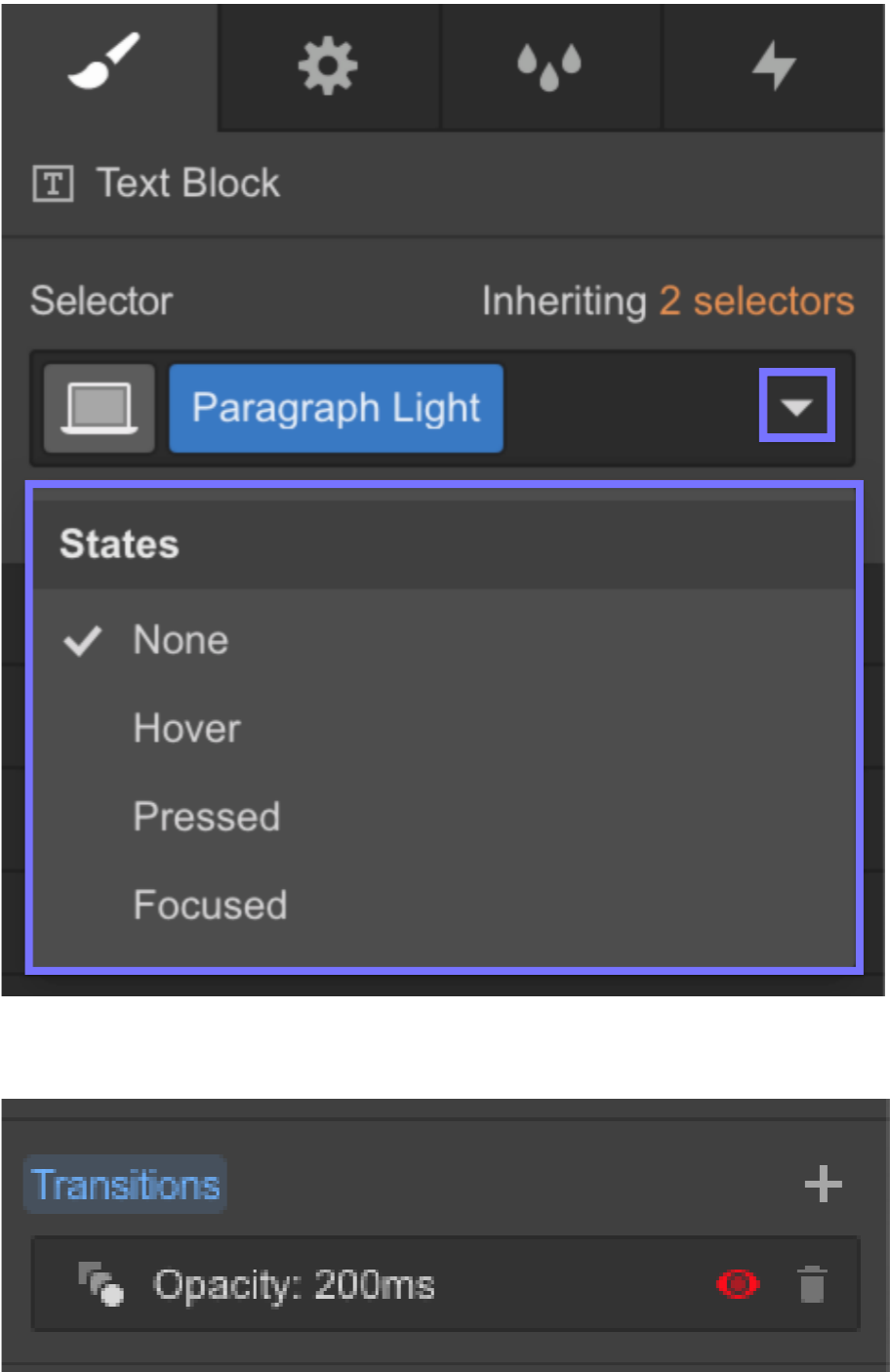
The hover state shows when you hover your mouse pointer on an element.

Changing styles on the hover state will show an element with those styles applied when the mouse pointer enters the boundaries of the element.

## TRANSITIONS

Transitions help create a smooth animation between different states of an element.

A common use case is to create a smooth transition for hover states on elements so that they don't abruptly change on hover.



# HTML Tags

Tags give us an easy way to control the default styling for a particular element type. For example, you can define the default styles for all paragraphs by editing the **all paragraphs tag**.

Adding a class with styles will override these default styles.

Tags are labeled **pink**, differentiating them from blue classes and green states.

