Codecademy Posicionamiento CSS
Alonso Bobadilla – Ingeniería Civil Informática

A continuación se mostrarán algunas capturas de la guía seguida en este módulo

1. See it to believe it

2. Taking up space

3. Inline-block

Inline-block

Good work! If you didn't notice much of a difference, don't worry. Our `<div>` s were `block` elements by default; as we specify different display values, they'll start to move around.

As mentioned, any element that comes in as a block (say, a paragraph) will automatically take up the full width of the page, no matter how much or how little content you put in.

If we specify a display of `inline-block`, however, our blocks are still blocks, but will be able to sit next to each other on the same line.

Instrucciones

Foro de preguntas   Glosario

```
1  * {
2      border: 1px dashed blue;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 2px solid black;
9      border-radius: 5px;
10     /*Add your CSS here!*/
11     display: inline-block;
12 }
13
14 #one {
15     background-color: #FF0000;
16 }
17
18 #two {
19     background-color: #0000FF;
20 }
21
22 #three {
23     background-color: #FFD700;
24 }
```

✓ ¡Bien!   **Iniciar lección siguiente**

≡ 3. Inline-block

4. Inline

Did you see that? Your `<div>` s all moved onto the same line! You can already start to see how this type of positioning can be useful for navigation bars like the one at the top of the main Codecademy page (where you can click "Learn," "Teach," and so on).

The `inline-block` value allows you to put several block elements on the same line. The `inline` value places all your elements next to one another, but not as blocks: they don't keep their dimensions.

Instrucciones

Try it and see! Set all your `<div>` s to have a `display` value of `inline`.

Foro de preguntas   Glosario

```
1  * {
2      border: 1px dashed blue;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 2px solid black;
9      border-radius: 5px;
10     /*Add your CSS here!*/
11     display: inline;
12 }
13
14 #one {
15     background-color: #FF0000;
16 }
17
18 #two {
19     background-color: #0000FF;
20 }
21
22 #three {
23     background-color: #FFD700;
24 }
```

✓ ¡Bien!   **Iniciar lección siguiente**

≡ 4. Inline

5. None!

CSS Positioning

index.html | stylesheet.css

suited for HTML elements that are blocks by default, such as headers and paragraphs.

Finally, we'll try out the display value `none`. As you might expect, this prevents the page from displaying the selected element. As you might *not* expect, this removes the selected element from the page *entirely*, including any children and any content. Poof! Gone! (But not gone forever —changing the display value away from `none` will bring everything back.)

Instrucciones

Give it a whirl! Set all your `<div>`s' `display` property to the `none` value.

```css
1  * {
2      border: 1px dashed blue;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 2px solid black;
9      border-radius: 5px;
10     /*Add your CSS here!*/
11     display: none;
12 }
13
14 #one {
15     background-color: #FF0000;
16 }
17
18 #two {
19     background-color: #0000FF;
20 }
21
22 #three {
23     background-color: #FFD700;
24 }
```

Pantalla completa

✓ ¡Felicitaciones, terminaste esta sección! | Siguiente: Margins, Borders, and Padding→

≡ 5. None!

7. Margin

CSS Positioning

index.html | stylesheet.css

Let's start with our margins. Adjusting our margins not only moves our element relative to other elements on the page, but also relative to the "walls" of the HTML document.

For instance, if we take an HTML element with a specific width (such as our `<div>` in the editor) and set its `margin` to `auto`, this tells the document to `auto`matically put equal left and right margins on our element, centering it on the page.

Instrucciones

Try it out! Set our div's `margin` property to `auto` to center our div on the page.

```css
1  * {
2      border: 1px dashed black;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 2px solid black;
9      border-radius: 5px;
10     background-color: #308014;
11     margin: auto;
12 }
```

Pantalla completa

✓ ¡Bien! | Iniciar lección siguiente

≡ 7. Margin

8. Margin top, right, bottom, left

CSS Positioning

index.html   stylesheet.css

You can also set an element's margins all at once: you just start from the top margin and go around clockwise (going from top to right to bottom to left). For instance,

```
margin: 1px 2px 3px 4px;
```

will set a top margin of 1 pixel, a right margin of 2, a bottom of 3, and a left of 4.
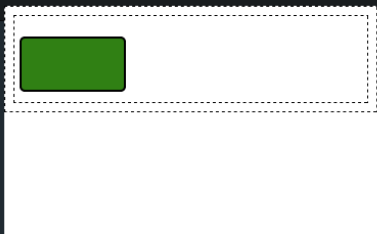
Instrucciones

Remove our div's `margin: auto;` on the CSS tab. Using whichever method you like best, give it a top margin of 20px, a right margin of 50px, a bottom margin of 10px, and a left margin of 5px.

```
1  * {
2      border: 1px dashed black;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 2px solid black;
9      border-radius: 5px;
10     background-color: #308014;
11     margin: 20px 50px 10px 5px;
12 }
```

Pantalla completa

Foro de preguntas   Glosario

✓ ¡Bien!   Iniciar lección siguiente

☰   8. Margin top, right, bottom, left

9. Borders

CSS Positioning

index.html   stylesheet.css

Borders

Well done! You can see how fine-tuning your margins will help you place elements where you'd like them to be on the page.

We've worked with borders before, but it never hurts to have extra practice.

Instrucciones
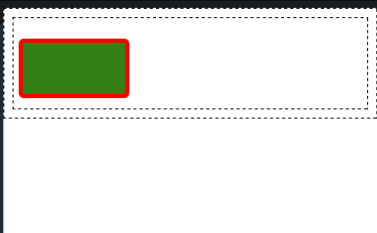
Change your div's border to 4 pixels wide and solid, with the hex color `#FF0000` .

```
1  * {
2      border: 1px dashed black;
3  }
4
5  div {
6      height: 50px;
7      width: 100px;
8      border: 4px solid #FF0000;
9      border-radius: 5px;
10     background-color: #308014;
11     margin: 20px 50px 10px 5px;
12 }
```

Pantalla completa

Foro de preguntas   Glosario

✓ ¡Bien!   Iniciar lección siguiente

☰   9. Borders

10. Padding

CSS Positioning

index.html   stylesheet.css

```
padding-right: /*some value*/
padding-bottom: /*some value*/
padding-left: /*some-value*/
```

Or select them all in one declaration, like this:

```
padding: value value value value;
```

You should also know that if you want your padding to be the same for all four sides, you can declare that value only once. `padding: 10px` will give your HTML element 10 pixels of padding on all sides.

Instrucciones

Go ahead and use your preferred method to give your div `padding` of `40px` on all sides.

```
 1 ⚠ * {
 2       border: 1px dashed black;
 3   }
 4
 5   div {
 6       height: 50px;
 7       width: 100px;
 8 ⚠     border: 4px solid #FF0000;
 9       border-radius: 5px;
10       background-color: #308014;
11       margin: 20px 50px 10px 5px;
12 ⚠     padding: 40px 40px 40px 40px;
13   }
```

Pantalla completa

Foro de preguntas   Glosario

✓ **¡Bien!**   **Iniciar lección siguiente**

☰   10. Padding

---

11. Negative values

CSS Positioning

index.html   stylesheet.css

between the element and its reference: for instance, if you have a `<div>` and you give it a `margin-left` of `20px`, it puts twenty pixels between the left margin of that `<div>` and the side of the screen. This effectively moves the `<div>` twenty pixels to the *right*.

If you want to move an element in the other direction, you can give CSS a *negative* value: `margin-left: -20px` will move the element twenty pixels to the *left*.

Instrucciones

Give your `<div>` a `margin-top` of `-20px` to see what happens.

```
 1 ⚠ * {
 2       border: 1px dashed black;
 3   }
 4
 5   div {
 6       height: 50px;
 7       width: 100px;
 8 ⚠     border: 4px solid #FF0000;
 9       border-radius: 5px;
10       background-color: #308014;
11       margin: -20px 50px 10px 5px;
12 ⚠     padding: 40px 40px 40px 40px;
13   }
```

Pantalla completa

Foro de preguntas   Glosario

✓ **¡Bien!**   **Iniciar lección siguiente**

☰   11. Negative values

12. Review

CSS Positioning

index.html   stylesheet.css

```
1   /*Add your CSS below!*/
2   div {
3       border: 1px solid black;
4       background-color: #CC0000;
5       margin: 10px 5px 5px 50px;
6       padding: 0px 30px 0px 10px;
7   }
8
```

Pantalla completa

Instrucciones

We've put a `<div>` for you to use on the HTML tab. On the CSS tab:

01. Give that div a border of 1px solid black.

02. Give it a background color of `#CC0000`.

03. Set its top margin to 10px, its right margin to 5px, its bottom margin to 5px, and its left margin to 50px.

04. Set its top padding to 0px, its right padding to 30px, its bottom padding to 0px, and its left padding to 10px.

❓ **¿Te trabaste?** ¡Te damos un

Foro de preguntas   Glosario

✅ **¡Felicitaciones, terminaste esta sección!**   **Siguiente: Floating→**

☰ 12. Review

13. To the right!

CSS Positioning

index.html   stylesheet.css

several elements all floating, they all know the others are there and don't land on top of each other.

You can think of the HTML page as sort of like a sea, and floating elements as boats on it: all the boats have positions on the sea, and they all see and steer clear of each other.

(Some of the positioning methods we'll learn in upcoming sections *can* accidentally drop elements on top of each other.)

```
1   div {
2       height: 300px;
3       width: 100px;
4       border: 2px solid black;
5       border-radius: 5px;
6       background-color: #308014;
7       /*Add your CSS here!*/
8       float: right;
9   }
```

Pantalla completa

Instrucciones

Let's get started. Set your div's `float` property to `right` !

Foro de preguntas   Glosario

✅ **¡Bien!**   **Iniciar lección siguiente**

☰ 13. To the right!

14. To the left!

CSS Positioning

| index.html | stylesheet.css |

```
1  div {
2      height: 300px;
3      width: 100px;
4      border: 2px solid black;
5      border-radius: 5px;
6      background-color: #308014;
7      /*Add your CSS here!*/
8      float: left;
9  }
```

To the left!

Good! As you saw, your div moved over to the right side of the page.

Instrucciones

Move it back by changing your div's `float` from `right` to `left`!

Pantalla completa

Foro de preguntas   Glosario

✓ ¡Bien!   **Iniciar lección siguiente**

☰   14. To the left!

15. Float for two

CSS Positioning

| index.html | stylesheet.css |

```
1  div {
2      height: 300px;
3      width: 300px;
4      border: 2px solid black;
5      border-radius: 5px;
6      background-color: #308014;
7      /*Add your CSS here!*/
8      float: right;
9  }
10
11 p {
12     font-family: Verdana, sans-serif;
13     font-size: 20px;
14     width: 280px;
15     /*Add your CSS here!*/
16     float: left;
17 }
```

Float for two

As you may have already guessed, we can use floated elements to naturally divide our pages into different sections. Try it!

Instrucciones

Set your `<div>` to float to the right and your `<p>` to float to the left.

❓ **¿Te trabaste?** ¡Te damos un

Pantalla completa

Foro de preguntas   Glosario

✓ ¡Bien!   **Iniciar lección siguiente**

☰   15. Float for two

16. Clearing elements

CSS Positioning

index.html   stylesheet.css

If you tell an element to `clear: left`, it will immediately move below any floating elements on the left side of the page; it can also clear elements on the `right`. If you tell it to `clear: both`, it will get out of the way of elements floating on the left *and* right!

The syntax is what you're used to:

```
element {
    clear: /*right, left, or both*/
}
```

Instrucciones

Tell the div with the ID `#footer` to `clear` `both`.

```
 9  }
10
11  .left {
12      height: 300px;
13      width: 150px;
14      background-color: #A7DBD8;
15      float: left;
16      margin-bottom: 10px;
17  }
18
19  .right {
20      height: 300px;
21      width: 450px;
22      background-color: #E0E4CC;
23      float: right;
24      margin-bottom: 10px;
25  }
26
27  #footer {
28      height: 50px;
29      background-color: #69D2E7;
30      /*Add your CSS here!*/
31      clear: both;
32  }
```

Pantalla completa

✔ **¡Felicitaciones, terminaste esta sección!**   Siguiente: Absolute, Relative, and Fixed Positioning→ ⊗

Foro de preguntas   Glosario

☰   16. Clearing elements

---

18. Absolute positioning

CSS Positioning

index.html   stylesheet.css

`position: absolute` gets positioned relative to `<html>`.

To show you how this works, we've set the `#outer` div to have absolute positioning. This means that when you position the `#inner` div, it will be relative to `#outer`. (If `#outer` had the default positioning of `static`, then `#inner` would get positioned relative to the entire HTML document.)

Instrucciones

Try it out: set `#inner`'s `position` to `absolute` and give it a `margin-left` of `20px`.

❓ **¿Te trabaste?** ¡Te damos un

```
 1  div {
 2      height: 100px;
 3      width: 100px;
 4      border-radius: 5px;
 5      border: 2px solid black;
 6  }
 7
 8  #inner {
 9      height: 75px;
10      width: 75px;
11      background-color: #547980;
12      /*Add your CSS here!*/
13      position: absolute;
14      margin-left: 20px;
15
16  }
17
18  #outer {
19      height: 1500px;
20      width: 150px;
21      background-color: #45ADA8;
22      position: absolute;
23      margin-left: 100px;
24  }
```

Pantalla completa

✔ **¡Bien!**   Iniciar lección siguiente ⊗

Foro de preguntas   Glosario

☰   18. Absolute positioning

19. Relative positioning

Relative positioning is more straightforward: it tells the element to move relative to where it would have landed if it just had the default `static` positioning.

If you give an element relative positioning and tell it to have a `margin-top` of `10px`, it doesn't move down ten pixels from any particular thing—it moves down ten pixels from where it *otherwise would have been.*

Instrucciones

Give it a try: change `#inner`'s `position` to `relative` and give it a `margin-left` of `200px`.

Foro de preguntas   Glosario

```
1  div {
2      height: 100px;
3      width: 100px;
4      border-radius: 5px;
5      border: 2px solid black;
6  }
7
8  #inner {
9      height: 75px;
10     width: 75px;
11     background-color: #547980;
12     /*Add your CSS here!*/
13     position: relative;
14     margin-left: 200px;
15
16  }
17
18  #outer {
19     height: 1500px;
20     width: 150px;
21     background-color: #45ADA8;
22     position: absolute;
23     margin-left: 100px;
24  }
```

Pantalla completa

✓ ¡Bien!   Iniciar lección siguiente

☰ 19. Relative positioning

20. Fixed positioning

Fixed positioning

Perfect! See? This positioning stuff's not so hard.

Finally, **fixed** positioning anchors an element to the browser window—you can think of it as gluing the element to the screen. If you scroll up and down, the fixed element stays put even as other elements scroll past.

Instrucciones

Set `#inner`'s position to `fixed`, then scroll up and down a bit. It stays put even as `#outer` moves out of the frame!

Foro de preguntas   Glosario

```
1  div {
2      height: 100px;
3      width: 100px;
4      border-radius: 5px;
5      border: 2px solid black;
6  }
7
8  #inner {
9      height: 75px;
10     width: 75px;
11     background-color: #547980;
12     /*Add your CSS here!*/
13     position: fixed;
14     margin-left: 200px;
15
16  }
17
18  #outer {
19     height: 1500px;
20     width: 150px;
21     background-color: #45ADA8;
22     position: absolute;
23     margin-left: 100px;
24  }
```

Pantalla completa

✓ ¡Felicitaciones, terminaste esta sección!   Siguiente: Review→

☰ 20. Fixed positioning

22. Navigation bar, where are you?

CSS Positioning

Check out the website we've started in the Result tab. Do you recognize it? It's the demo we showed you in the first CSS lesson!

It doesn't look quite the same, though. This is because much of the crucial display and positioning CSS we used has been removed. Your job? Add it back in!

Instrucciones

There's a navigation bar here somewhere, but it's lost due to display problems! On the CSS tab, give `#navbar` a position of fixed and a top margin of -10px.

```
10      padding-bottom: 25px;
11      color: #acd1b2;
12  }
13
14  img {
15      height: 170px;
16      width: 170px;
17      box-shadow: rgba(0,0,0,0.2) 10px 10px;
18
19  }
20
21  #navbar {
22      /*Add your CSS here!*/
23      position: fixed;
24      margin-top: -10px;
25      left:50%;
26      margin-left:-254px;
27  }
28
29  #header {
30      position: relative;
31      top: -10px;
32      background-color: #3c4543;
33      border-top-left-radius: 15px;
```

✅ **¡Bien!** **Iniciar lección siguiente**

Foro de preguntas   Glosario

≡ 22. Navigation bar, where are you?

23. Displaying it properly

CSS Positioning

Displaying it properly

Good work! The navigation bar is all stacked up, however, instead of being laid out horizontally.

Instrucciones

Fix this by:

01. Setting `li`'s display value to `inline`;

02. Giving it 5 pixels of padding all around.

```
39      position: fixed;
40      margin: -10px;
41  }
42
43  li {
44      /*Add your CSS here!*/
45      display: inline;
46      padding: 5px 5px 5px 5px;
47      border: 2px solid #000000;
48      font-family: Futura, Tahoma, sans-serif;
49      color: #ffffff;
50      border-radius: 5px 5px;
51      background-color: #cc0323
52  }
53
54  #left{
55      /*Add your CSS here!*/
56
57  }
58
59  p {
60      font-family: Tahoma;
61      font-size: 1em;
62  }
```

✅ **¡Bien!** **Iniciar lección siguiente**

Foro de preguntas   Glosario

≡ 23. Displaying it properly

24. Floating right along

25. You've done it!

Curso finalizado