5.3 Positioning elements

The position property

The CSS ***position*** property gives developers more control over where elements should appear in the browser. `position` has four possible values:

- `static` - ***Static positioning*** is the default positioning
- `relative` - ***Relative positioning*** positions the element relative to the element's default position
- `fixed` - ***Fixed positioning*** positions the element relative to the viewport in a fixed location
- `absolute` - ***Absolute positioning*** positions the element relative to the nearest positioned ancestor

PARTICIPATION ACTIVITY 5.3.1: Relative positioning.

1 2 3 4 5 2x speed

```
#content {
  border: solid 2px blue;
  color: blue;
  position: relative;


}
```


```
<div>Before</div>
<div id="content">Content</div>
<div>After</div>
```

Adding relative positioning to #content does not change the "Content" position until "left" and/or "top" properties are specified.

Captions

1. The "Content" displays in the default location.
2. Adding relative positioning to #content does not change the "Content" position until "left" and/or "top" properties are specified.
3. "left: -20px" moves the left edge 20 pixels left from the default location.
4. "left: 20px" moves the left edge 20 pixels to the right of the default location.
5. Negative values for "top" move the element up, and positive values move the element down.

Feedback?

PARTICIPATION ACTIVITY 5.3.2: Relative and static positioning.

1) Where is the image located relative to the image's default location?

```
<img src="world.jpg"
style="position:relative;
left:30px">
```

- (•) 30 pixels to the right
- ( ) 30 pixels to the left
- ( ) No change

Correct

`30px` is a positive value, so the image is 30 pixels to the right of the image's default location.

2) Where is the image located relative to the image's default location?

```
<img src="world.jpg"
style="position:relative;
top:-30px">
```

- (•) 30 pixels higher
- ( ) 30 pixels lower
- ( ) No change

Correct

`-30px` is a negative value, so the image is 30 pixels higher from the image's default location.

3) Where is the image located relative to the image's default location?

```
<img src="world.jpg"
style="position:static;
left:-20px; top:30px">
```

- ( ) 20 pixels to the right and 30 pixels higher
- ( ) 20 pixels to the left and 30 pixels lower
- (•) No change

Correct

Static positioning leaves the image at the default location.

Feedback?

Fixed positioning

Fixed positioning places the element at a fixed location in the viewport, and scrolling does not move the element. A ***viewport*** is the visible area of a webpage. The fixed element is detached from the normal flow of elements in the page and is layered on top of the page contents.

PARTICIPATION ACTIVITY 5.3.3: Fixed positioning.

1 2 3 4 2x speed

```
#content {
  border: solid 2px blue;
  color: blue;


}
```

Before
Content
After

```
<div>Before</div>
<div id="content">Content</div>
<div>After</div>
```

The "Content" displays in the default location.

Captions

1. The "Content" displays in the default location.
2. Adding fixed positioning to #content detaches the "Content" so the is layered on top of the underlying content.
3. "left: 60px" moves the 's left edge 60 pixels to the right of the browser's left edge.
4. "top: 50px" moves the 's top edge 50 pixels below the browser's top edge.

Feedback?

PARTICIPATION ACTIVITY 5.3.4: Fixed positioning.

Refer to the CSS below.

```
.special {
  position: fixed;
  left: 100px;
  top: 25px;
}
```

1) All elements using the "special" class are displayed 100 pixels from the browser's left edge and 25 pixels from the browser's top edge.

- (●) True
- ( ) False

Correct

Fixed positioning fixes the element's position in the browser.

2) All elements using the "special" class scroll with the page contents.

- ○ True
- ◉ False

Correct

Fixed-positioned elements remain in a fixed location and do not scroll.

3) The text "123" is displayed on top of "ABC".

```
<span
class="special">ABC</span>
<span
class="special">123</span>
```

- ◉ True
- ○ False

Correct

The content of both spans are rendered at the same fixed position in the browser.

Feedback?

Absolute positioning

Absolute positioning is similar to fixed positioning except:

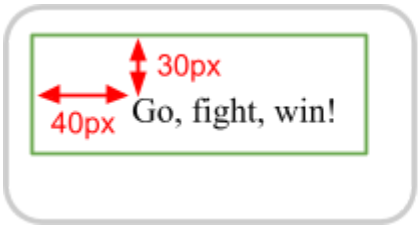
1. The position is based on the nearest positioned ancestor element that uses fixed, absolute, or relative positioning. If no positioned ancestor element exists, the element is positioned relative to the document body.
2. An absolute-positioned element scrolls with the document unless an ancestor element is using fixed positioning.

Figure 5.3.1: Cheer is absolute positioned inside a positioned ancestor (left) and relative to the document body (right).

```
#container {
  border: solid 2px green;
  position: relative;
  height: 60px;
  width: 150px;
}

#cheer {
  position: absolute;
  left: 40px;
  top: 30px;
}
```

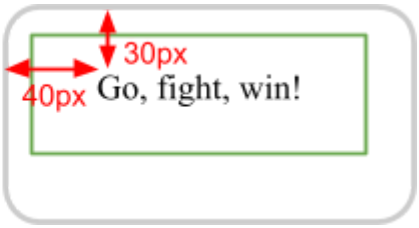
```
<div id="container">
  <div id="cheer">Go, fight,
win!</div>
</div>
```




```
#container {
  border: solid 2px green;
  /* No positioning */
  height: 60px;
  width: 150px;
}

#cheer {
  position: absolute;
  left: 40px;
  top: 30px;
}
```

```
<div id="container">
  <div id="cheer">Go, fight,
win!</div>
</div>
```



Feedback?

PARTICIPATION ACTIVITY 5.3.5: Absolute positioning.

Refer to the CSS below.

```
.special {
  position: absolute;
  left: 100px;
  top: 25px;
}
```

1) The `<span>` is displayed 100 pixels from the browser's left edge and 25 pixels from the browser's top edge.

```
<body>
  <span
class="special">Special</span>
</body>
```

- (●) True
- ( ) False

Correct

The absolute-positioned element is positioned relative to the document body.

2) Elements using the "special" class that do not have a positioned

Correct

Absolute-positioned elements that do not have a positioned ancestor are positioned relative to the

ancestor will scroll with the page contents.

- True
- False

document body, and elements positioned relative to the document body scroll with the page contents.

3) If the "container" class uses fixed positioning, the `<span>` will not scroll with the page contents.

```
<div class="container">
  <span
class="special">Special</span>
</div>
```

- True
- False

Correct

Fixed elements do not scroll, so elements positioned relative to a fixed element do not scroll.

4) If the "container" class uses static positioning, the `<span>` is positioned relative to the `<div>`.

```
<body>
  <div class="container">
    <span
class="special">Special</span>
  </div>
</body>
```

- True
- False

Correct

Since the `<div>` uses static positioning, the `<div>` is not a positioned ancestor. So the `<span>` is positioned relative to the document body.

Feedback?

z-index property

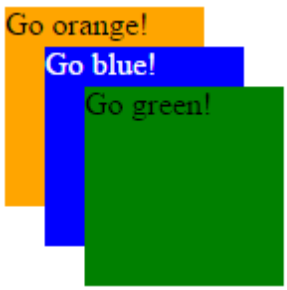
When a relative, absolute, or fixed element is placed on top of another positioned element, the element that is specified last in the HTML is placed on top. However, the CSS ***z-index*** property is used to specify a relative distance that orders the appearance of elements. Elements with higher `z-index` values are placed on top of elements with lower `z-index` values.

On the left side of the figure below, the browser renders the square elements in the order the elements appear in the HTML: The orange square is rendered first, and the green square is rendered last. The right side of the figure shows how the ordering changes using the `z-index` property: The orange square has the largest `z-index` and therefore appears on top.

Figure 5.3.2: No z-index is used on the left, but z-index changes the rendered order on the right.

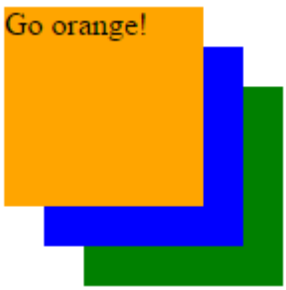
```
div {
  width: 100px;
  height: 100px;
  position: absolute;
}
#orange {
  background-color: orange;
  left: 10px;
  top: 10px;
}
#blue {
  background-color: blue;
  color: white;
  left: 30px;
  top: 30px;
}
#green {
  background-color: green;
  left: 50px;
  top: 50px;
}
```

```
<div id="orange">Go orange!
</div>
<div id="blue">Go blue!
</div>
<div id="green">Go green!
</div>
```




```
div {
  width: 100px;
  height: 100px;
  position: absolute;
}
#orange {
  background-color: orange;
  z-index: 3;
  left: 10px;
  top: 10px;
}
#blue {
  background-color: blue;
  color: white;
  z-index: 2;
  left: 30px;
  top: 30px;
}
#green {
  background-color: green;
  z-index: 1;
  left: 50px;
  top: 50px;
}
```

```
<div id="orange">Go orange!
</div>
<div id="blue">Go blue!
</div>
<div id="green">Go green!
</div>
```



Feedback?

PARTICIPATION ACTIVITY 5.3.6: z-index.

Refer to the figure above.

1) In the example on the right, what `z-index`

Correct

value would make the green square appear on top of the orange and blue squares?

- 1
- 2
- 4

Giving the green square a `z-index` greater than the orange and blue squares' `z-index` makes the green square appear on top.

2) If all three squares are given the same `z-index` value of 5, which square appears on top?

- orange
- blue
- green

Correct

When all `z-index` values are equal, the browser renders each square in the order the squares appear in the HTML, as shown on the left side of the figure.

Feedback?

PARTICIPATION ACTIVITY 5.3.7: Positioning practice.

The webpage below displays the iconic "I ♥ NY" logo. Use the `position` and `z-index` properties to make the webpage render like the expected webpage:

1. Use relative positioning in the `img` CSS rule to place the t-shirt image 10 pixels further to the right of the image's default location.
2. Use absolute positioning in the `.first`, `.heart`, and `.last` CSS rules to place "I", "♥" and "NY" in the correct configuration on top of the t-shirt.

HTML CSS

```
<span class="first words">I</span>
<span class="heart">&hearts;</span>
<span class="last words">NY</span>

<div>
   <img src="https://resources.zybooks.com/WebProgramming/tshirtv
</div>
```

Render webpage Reset code

Your webpage **Expected webpage**

▼ View solution

Explain

```
--- START FILE: HTML ---

<span class="first words">I</span>
<span class="heart">&hearts;</span>
<span class="last words">NY</span>

<div>
   <img
src="https://resources.zybooks.com/WebProgramming/tshirtv1.pn
alt="White t-shirt">
```

```
</div>


--- END FILE: HTML ---

--- START FILE: CSS ---

img {
  position: relative;
  left: 10px;
  z-index: 0;
}

.words {
   font-family: impact;
   font-size: 60px;
}

.first {
   position: absolute;
   left: 130px;
   top: 60px;
   z-index: 2;
}

.heart {
   color: red;
   font-size: 80px;
   position: absolute;
   left: 150px;
   top: 50px;
   z-index: 2;
}

.last {
  position: absolute;
  left: 130px;
  top: 115px;
  z-index: 2;
}


--- END FILE: CSS ---
```

Feedback?

CHALLENGE ACTIVITY 5.3.1: Positioning elements.

530096.4000608.qx3zqy7

Jump to level 1

1 2 3 4

Using the z-index property, order the images so the bike image appears on top of the lake image, and the lake image appears on top of the food image.

SHOW EXPECTED

CSS HTML

```
4  }
5
6  #bike {
7     left: 10px;
8     top: 10px;
9     z-index:3;
10 }
11 #food {
12    left: 30px;
13    top: 30px;
14    z-index:1;
15 }
16 #lake {
17    left: 50px;
18    top: 50px;
19    z-index:2;
```

1 2 3 4

Check Next **Done**. Click any level to practice more. Completion is preserved.

For the element with id bike, is the z-index an integer?

Yours `Yes`

For the element with id food, is the z-index an integer?

Yours `Yes`

For the element with id lake, is the z-index an integer?

Yours `Yes`

Is bike above lake?

Yours `Yes`

Is lake above food?

Yours `Yes`

Your webpage

View your last submission

```
#bike {
   left: 10px;
   top: 10px;
   z-index:2;
}
#food {
   left: 30px;
   top: 30px;
   z-index:3;
}
#lake {
   left: 50px;
   top: 50px;
   z-index:1;
```

Feedback?

Exploring further:

- CSS Layout - The position Property from W3Schools

How was this section? Provide section feedback