Subject Code:	CS 3	Computer Science 3
Module Code:	5.0	Introduction to Cascading Style Sheets
Lesson Code:	5.6	Float, Position, width, height, top, bottom, left, right
Time Frame:		30 minutes

TARGET

Time Allocation: 1 min.

After completing this module, you are expected to:

By the end of this learning guide module, the students should be able to learn and apply:

1. relative positioning
2. absolute positioning
3. containing blocks

HOOK

Time Allocation: 1 min.

The previous learning module covers the three values for CSS position, the **position:static; position: relative;** and **position: fixed;**. CSS position is sometimes considered an advanced CSS property because it is not as intuitive as font-size or margin, etc., since it changes the natural render flow of elements in a browser.

In this learning module, we are going to look at **position: absolute** and on how it will work with **position: relative** since they are perhaps the most versatile ones that will get you a lot of mileage once you feel confident with them.

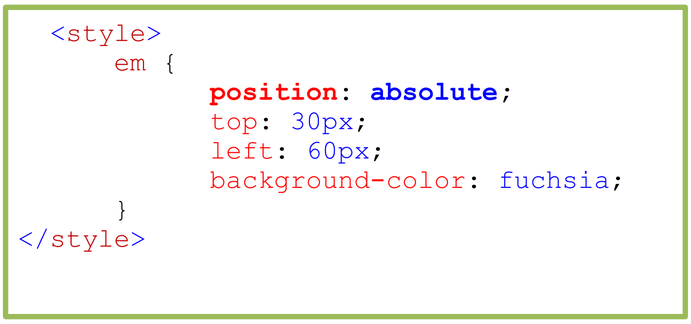
IGNITE

Time Allocation: 18 mins.

position: absolute;

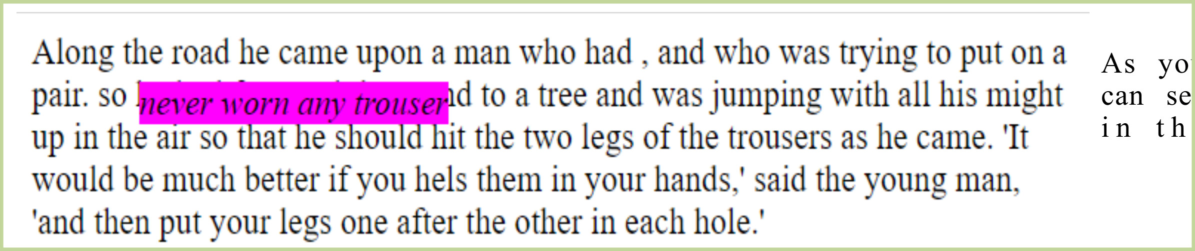
position: absolute; works a bit differently and is actually a more flexible method for accurately placing items on a page compared to relative positioning. Now that you have seen how relative positioning works in the previous lesson, let us take the same example as shown below, only this time we will change the value of the position to absolute.

The style sheet:



```
  <style>
      em {
            position: absolute;
            top: 30px;
            left: 60px;
            background-color: fuchsia;
      }
</style>
```

This will produce the following result:



Along the road he came upon a man who had , and who was trying to put on a pair. so never worn any trouserd to a tree and was jumping with all his might up in the air so that he should hit the two legs of the trousers as he came. 'It would be much better if you hels them in your hands,' said the young man, 'and then put your legs one after the other in each hole.'

As you can see in the example, the space once occupied by the em element is now closed up, as is the case for all absolutely positioned elements. In its new position, the element box overlaps the surrounding content. In the end, absolutely positioned elements have no influence whatsoever on the layout of surrounding elements.

The most significant difference here from position relative is the location of the positioned element. This time, the offset values position the em element 30 pixels down and 60 pixels to the right of the top-left corner of the browser window.

Containing blocks

The CSS2.1 Recommendation states, “The position and size of an element’s box(es) are sometimes calculated relative to a certain rectangle, called the containing block of the element.” It is critical to have an awareness of the containing block of the element you want to position. We sometimes refer to this as the positioning content.

The recommendation lays out a number of intricate rules for determining the containing block of an element, but it basically boils down to this.

- If the positioned element is not contained within another positioned element, then it will be placed relative to the initial containing block.
- If the element has an ancestor that has its position set to **relative, absolute, or fixed**, the element will be positioned relative to the edges of that element instead.

This is the revised style rule:

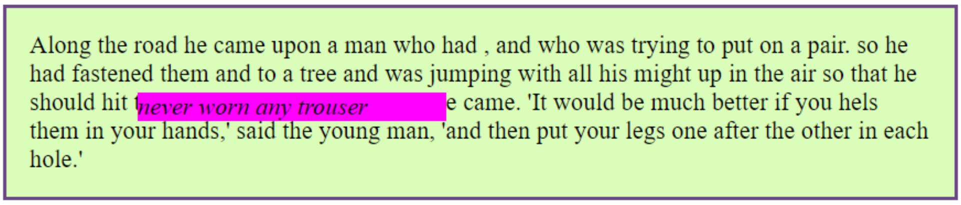
This time, let us add width and margin properties to the positioned **<em>** element.

```
<style>
        em {
                width: 200px;
                margin: 25px;
                position: absolute;
                top: 30px;
                left: 60px;
                background-color: fuchsia;
        }

        p {
                position: relative;
                padding: 15px;
                background-color: #DBFDBA;
                border: 2px solid #6C4788;

        }
</style>
```

This will produce the following result:



Along the road he came upon a man who had , and who was trying to put on a pair. so he had fastened them and to a tree and was jumping with all his might up in the air so that he should hit t*never worn any trouser*e came. 'It would be much better if you hels them in your hands,' said the young man, 'and then put your legs one after the other in each hole.'

Here we can see that:

- The offset values apply to the outer edges of the element box (from margin edge to margin edge), and
- Absolutely positioned elements always behave as block-level elements. For example, the margins on all sides are maintained, even though this is an inline element. It also permits a width to be set for the element.

It is important to keep in mind that once you have positioned an element, it becomes the new containing block for all the elements it contains.

Specifying position

Pixel measurement

The positive offset values push the positioned element box away from the specified edge and toward the center of the containing block. If there is no value provided for a side, it is set to auto, and the browser adds enough space to make the layout work.

In this example, pixel lengths are used for all four offset properties to place the positioned element at a particular spot in its containing element.

The source:

```
<div id="a"><div id="b"></div></div>
```

The style sheet:

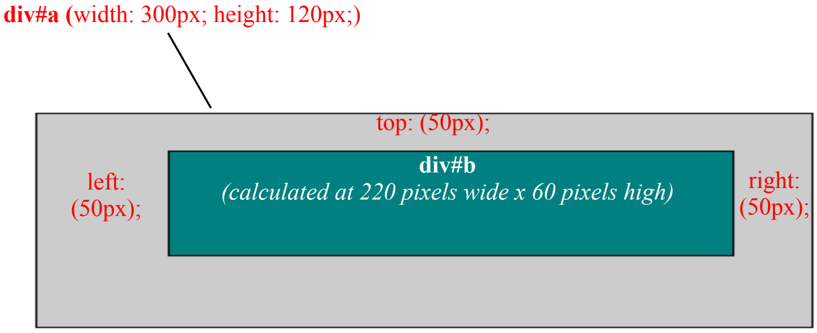
```
<style>

      div#a {position: relative; /* creates the containing block */
                   height: 120px;
                         width: 300px;
                   border: 1px solid;
                   background-color: #CCC;
      }

      div#b {position: absolute;
                   top: 20px;
                   right: 30px;
                   bottom: 40px;
                   left: 50px;
                   border: 1px solid;
                   background-color: teal;
      }

</style>
```

This will produce the following result:


div#a (width: 300px; height: 120px;)
top: (50px);
left: (50px);
div#b
(calculated at 220 pixels wide x 60 pixels high)
right: (50px);

Notice that by setting offsets on all four sides, we have indirectly set the dimensions of the positioned div#b (it fills the 220 x 60 pixel space that is left over within the containing block after the offset values are applied).

Percentage values

You can also specify positions with percentage values. In the first example, the image is positioned halfway down the left edge of the containing block. In the second example on the right, the img element is positioned so that it always appears in the bottom-right corner of the containing block.

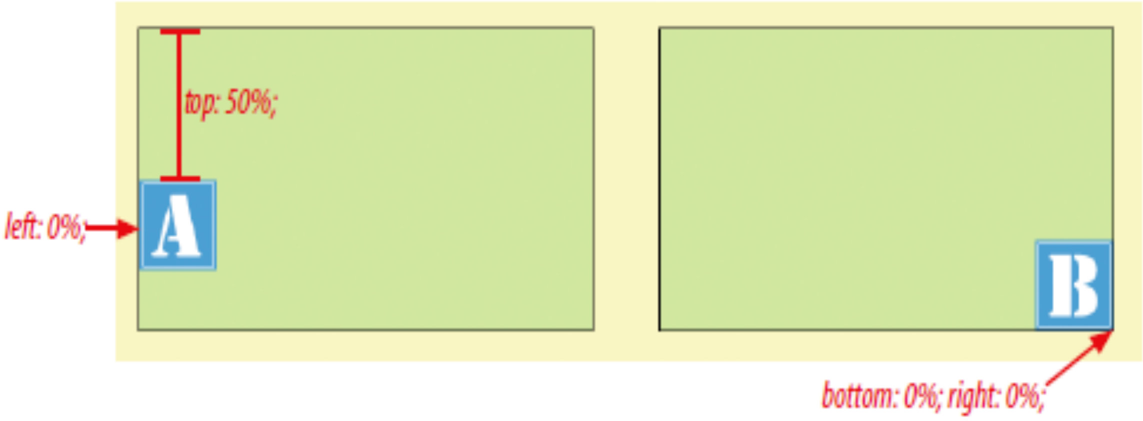
The source:

```
<img id="A" src="A.jpg">
<img id="B" src="B.jpg">
```

The style sheet:

```
img#A {
      position: absolute;
      top: 50%;
      left: 0%; /* the % symbol could be omitted for a 0 value */
}
img#B {
      position: absolute;
      bottom: 0%; /* the % symbol could be omitted for a 0 value */
      right: 0%; /* the % symbol could be omitted for a 0 value */
}
```

This will produce the following result:



Although the examples here specify both a vertical and horizontal offset, it is common to provide just one offset for a positioned element, for example to move it left or right into a margin using either left or right properties.

NAVIGATE

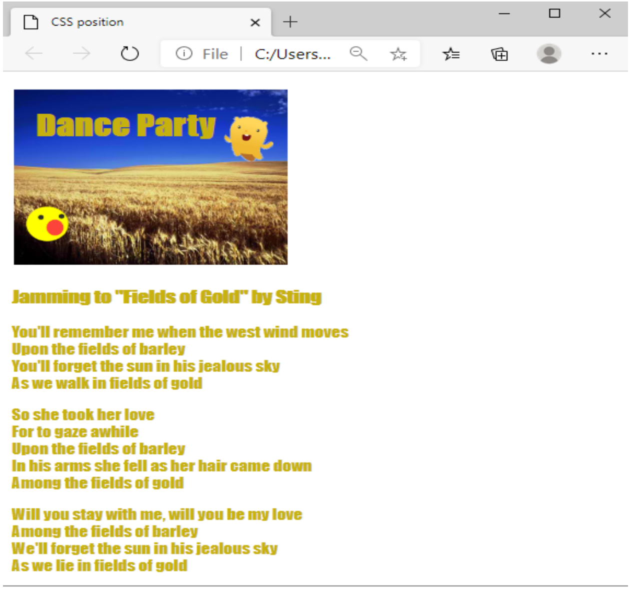
Time Allocation: 10 mins.

It is your time to apply what you have learned.

The goal of this activity is to use the CSS position and z-index property to create a webpage that looks like the one below.

- Create a blank web page with just html, head, title, body, and a style tag.
- Copy and modify the given source code and create the styling sheet.

Here is what it looks like:



Jamming to "Fields of Gold" by Sting

You'll remember me when the west wind moves
Upon the fields of barley
You'll forget the sun in his jealous sky
As we walk in fields of gold

So she took her love
For to gaze awhile
Upon the fields of barley
In his arms she fell as her hair came down
Among the fields of gold

Will you stay with me, will you be my love
Among the fields of barley
We'll forget the sun in his jealous sky
As we lie in fields of gold

Download pictures needed using the link below:

fields of grain (*https://www.kasandbox.org/programming-images/landscapes/fields-of-grain.png*)
Hopper-jumping (*https://www.kasandbox.org/programming-images/avatars/Hopper-jumping.gif*)
Winston (*https://www.kasandbox.org/programming-images/creatures/Winston.png*)

The Source:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>CSS position</title>

   </head>
    <body>

         <h1>Dance Party</h1>

         <img src="https://www.kasandbox.org/programming-images/
         landscapes/fields-of-grain.png" id="landscape">
         <img src="https://www.kasandbox.org/programming-images/avatars/
         Hopper-jumping.gif" id="hopper">
         <img src="https://www.kasandbox.org/programming-images/
         creatures/Winston.png" id="winston">

         <div id="song">

             <h3>Jamming to "Fields of Gold" by Sting</h3>
             <p>You'll remember me when the west wind moves<br>
               Upon the fields of barley<br>
               You'll forget the sun in his jealous sky<br>
               As we walk in fields of gold
             </p>

             <p>So she took her love<br>
               For to gaze awhile<br>
               Upon the fields of barley<br>
               In his arms she fell as her hair came down<br>
               Among the fields of gold</p>
             <p>Will you stay with me, will you be my love<br>
               Among the fields of barley<br>
               We'll forget the sun in his jealous sky<br>
               As we lie in fields of gold</p>

         </div>

  </body>
</html>
```

Here are recommended steps for making this happen:

1. Write CSS rule to set the whole document to **"fantasy"** font face with font color of **rgb(199, 177, 14).**
2. Create an id named **landscape** that is 20 pixels from the top, 10 pixels from the left, 250 pixels width. Add a z-index property with value 1 and use an absolute position.
3. Create an id named **winston** that is 150 pixels from the top, 20 pixels from the left, 40 pixels width and height. Add a z-index property with value 2 and use an absolute position.
4. Create an id named **hopper** that is 30 pixels from the top, 200 pixels from the left, 50 pixels width, with z-index value of 3 and an absolute position.
5. Write CSS rule to position the heading: set the **"Dance Party"** to fixed positioning, 30 pixels from the top, and 200 pixels from the left.
6. Create an id named **song** with 220 pixels from the top and set the position to relative.

Note: This is a **graded activity**. Save your file as **Section_FullName_Position.html** and send your final solution to ________________________.

This activity will be graded using the following rubric:

REQUIREMENTS	POINTS		Total
Styles:			
The whole document used **"fantasy"** font face with font color rgb **(199, 177, 14).**	Complied – 1 pt Incomplete – 0 pt		
The id named **landscape** has 20 pixels from the top, 10 pixels from the left, 250 pixels width, a z-index property with value 1 and an absolute position.	Complied – 2 pt Incomplete – 0 pt		
The id named **winston** has 150 pixels from the top, 20 pixels from the left, 40 pixels width and height, a z-index property with value 2 and with an absolute position.	Complied – 2 pt Incomplete – 0 pt		
The id named **hopper** has 30 pixels from the top, 200 pixels from the left, 50 pixels width, z-index value of 3 and an absolute position.	Complied – 2 pt Incomplete – 0 pt		
The position of the heading **"Dance Party"** is fixed with 30 pixels from the top, and 200 pixels from the left.	Complied – 2 pt Incomplete – 0 pt		

The id named **song** has 220 pixels from the top and set the position to relative.	Complied – 1 pt Incomplete – 0 pt		
Total Points	**10**		

KNOT

Time Allocation: 1 min.

In summary,

- **position: static** is the default for an HTML element. The element position on the browser when rendered is in the order on how it was constructed in the HTML document. Most often you will not need to set this on an element.
- The element is positioned according to the normal flow of the document, and then offset *relative to itself* based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were static.
- **position: absolute;** works a bit differently and is actually more flexible method for accurately placing items a page relative positioning.
- "The position and size of an element's box(es) are sometimes calculated relative to a certain rectangle, called the containing block of the element." It is critical to have an awareness of the containing block of the element you want to position. We sometimes refer to this as the positioning content.
- You can also specify positions with pixel measurements and percentage values.

REFERENCES

Jennifer Niederst Robbins (2012). Floating and Positioning. In Jennifer Niederst Robbins (4th Edition), Learning Web Design A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (341-371). Canada, O'Reilly Media, Inc.

CSS Positioning (n.d.). Retrieve from https://codepen.io/jorgecardoso/post/exercises-css-positioning

Jacob Jenkov (2019, June 03) CSS Position. Retrieve from http://tutorials.jenkov.com/css/position.html

Prepared by:	Reviewed by:
LEO R. MACALALAD SST-III PSHS-CALABARZON	ALINE TERESA MENDOZA SST V PSHS- Main