upGrad

#RahoAmbitious

Full Stack Software Development

Course: Introduction to Web Development

Lecture On: CSS Layouts, CSS Combinators and CSS Pseudo Elements

upGrad

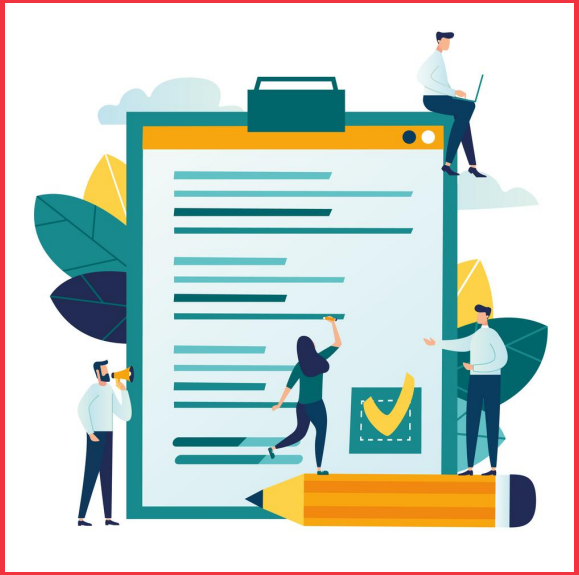
upGrad

In the last class, we discussed...

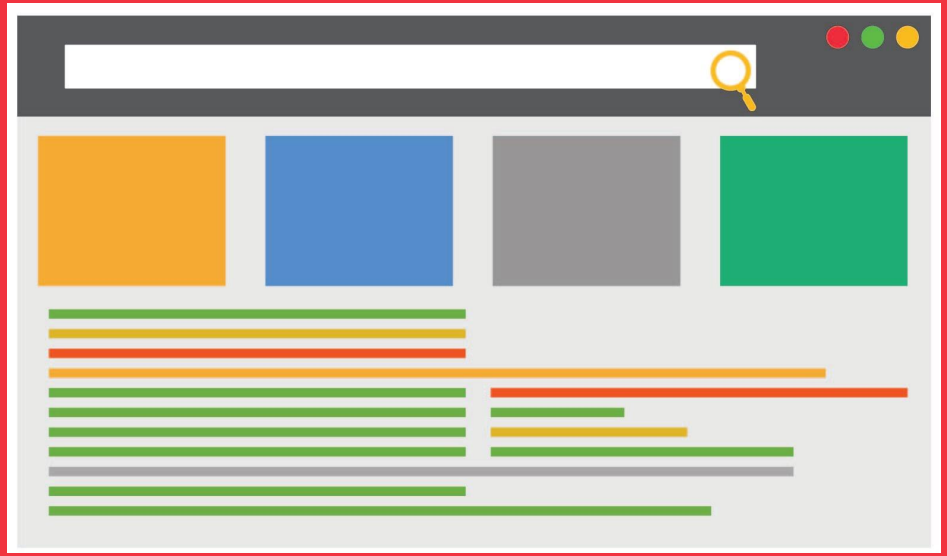
- CSS Box Model

Today's Agenda

- CSS Layouts
- CSS Combinators
- CSS Pseudo Elements

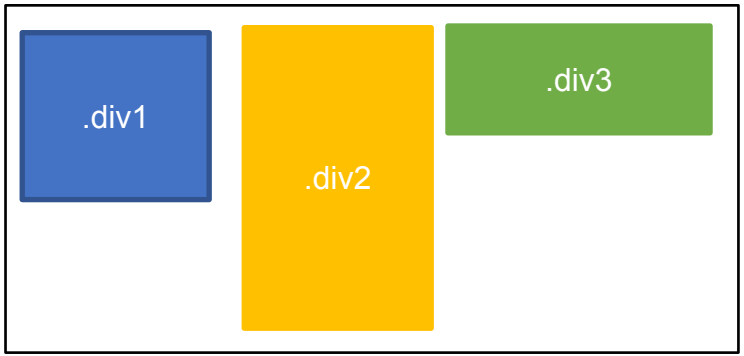


CSS Layouts

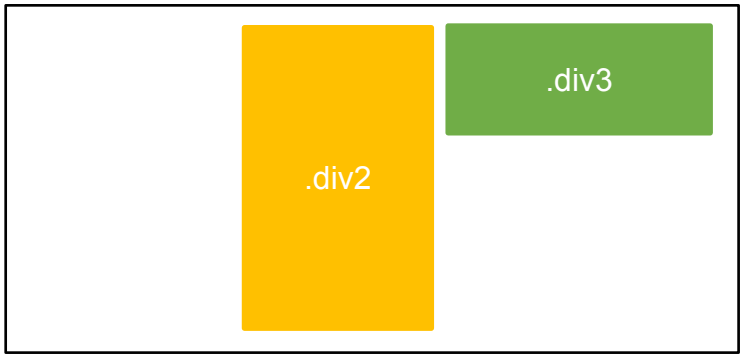


CSS visibility

The '**visibility**' property specifies whether or not an element should be visible on a webpage.



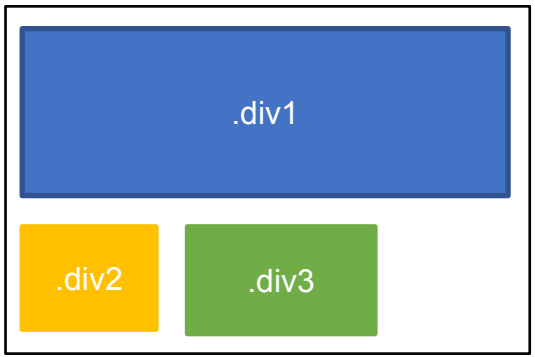

```
.div1 {
   visibility: visible;
}
```



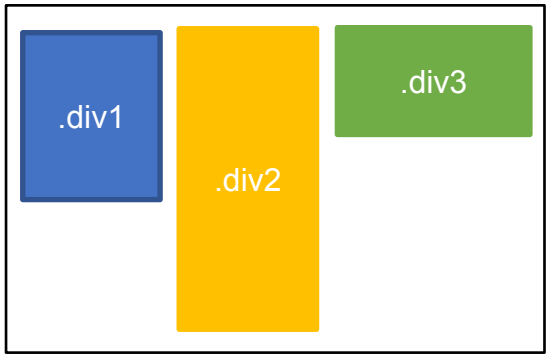

```
.div1 {
   visibility: hidden;
   /*This hides the element */
}
```

CSS display

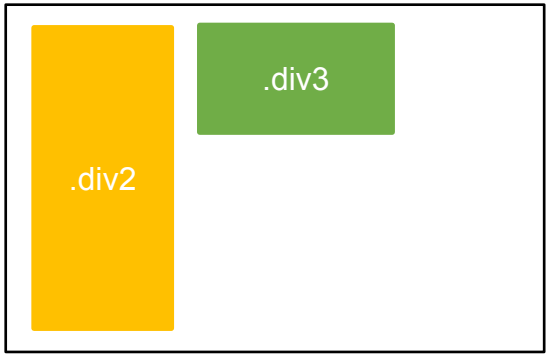
The '**display**' property specifies how an element should be displayed on a webpage.




```
.div1 {
   display: block;
}
/* This will display div1 as
block element */
```




```
.div1 {
   display: inline;
}
/* This will display div1
inline with other div */
```

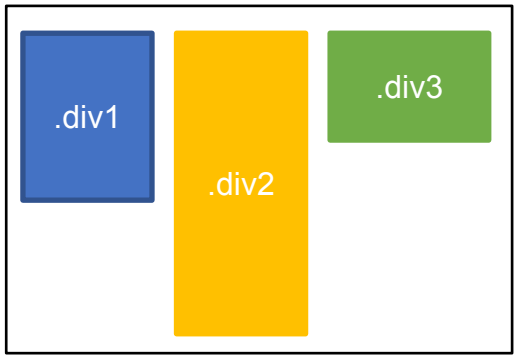



```
.div1 {
   display: none;
}
/* This will not display div1
*/
```

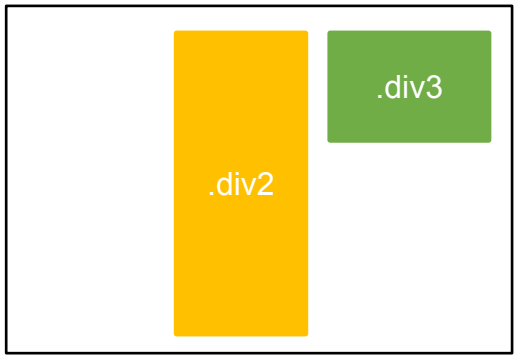
You can read this article for better understanding

CSS visibility vs display

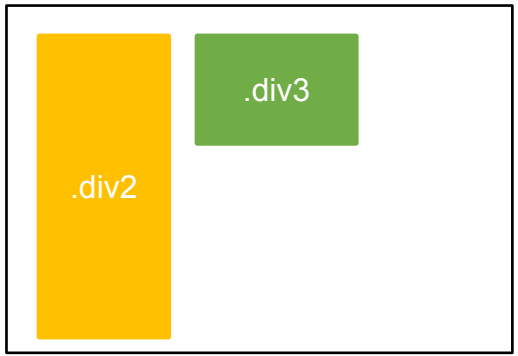
- The '**visibility: hidden;**' property only hides an element; it leaves the space that would be taken up by the element.
- The '**display: none;**' property not only hides an element, but also removes the space that would have been taken up by the element.



Normal Display of an Element




```
.div1 {
   visibility: hidden;
}
```




```
.div1 {
   display: none;
}
```

Poll 1 (15 Sec)

Which of the following statement(s) about the CSS properties '***visibility:hidden***' and '***display:none***' is/are true?
(Note: More than one option can be correct)

1. ***visibility:hidden*** means the element is invisible and space for the element is also not allocated.
2. ***visibility:hidden*** means the element is invisible. However, space for the element is allocated.
3. ***display:none*** means the element is invisible and space for the element is also not allocated.
4. ***display:none*** means the element is invisible and space for the element is allocated.

Poll 1 (Answer)

Which of the following statement(s) about the CSS properties '***visibility:hidden***' and '***display:none***' is/are true?
(Note: More than one option can be correct)

1. ***visibility:hidden*** means the element is invisible and space for the element is also not allocated.
2. ***visibility:hidden*** **means the element is invisible. However, space for the element is allocated.**
3. ***display:none*** **means the element is invisible and space for the element is also not allocated.**
4. ***display:none*** means the element is invisible and space for the element is allocated.

CSS Overflow

- The '**overflow**' property specifies what should happen when a content overflows its containers.

- This property has the following four values:

1. **visible** – This is the default value of this property. With this value, the overflowing content will not be clipped or hidden. In fact, the overflowing content renders outside of the container.
2. **hidden** – With this value, the overflowing content is hidden. The remaining content remains visible.
3. **scroll** – With this value, the overflow is clipped and a scrollbar is inserted inside the container. You can view the overflowing content by scrolling the container.
4. **auto** – This value is like the **visible value,** the only difference here is that the scrollbar is added only when the content overflows.

- Apart from the '**overflow**' property, you can also apply overflow individually to the vertical or the horizontal content of the container.
  The '**overflow-x**' property will apply *visible/hidden/scroll/auto* to only the horizontal overflowing content of the container, whereas the '**overflow-y**' property will behave in a similar fashion on the vertical overflowing content of the container.

Poll 2 (15 Sec)

Which of the following statements(s) about the CSS ‘overflow’ property is/are true?
(Note: More than one option can be correct)

1. The CSS ‘overflow’ property can have the values auto, none, visible and scroll.
2. The CSS ‘overflow’ property specifies the action to be taken if the content inside an element overflows.
3. The CSS ‘overflow’ property is present by default.
4. All of the above

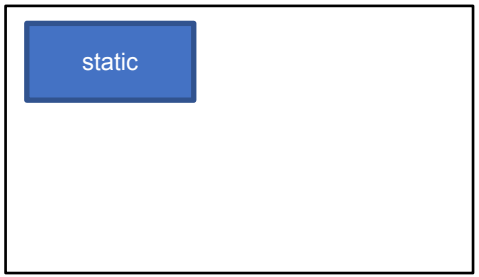
Poll 2 (Answer)

Which of the following statements(s) about the CSS ‘overflow’ property is/are true?
(Note: More than one option can be correct)

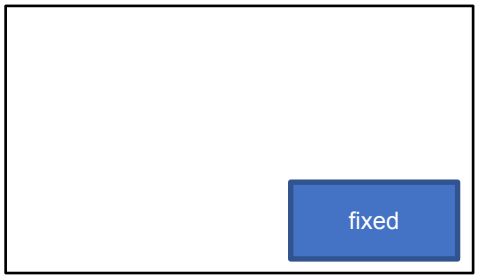
1. The CSS ‘overflow’ property can have the values auto, none, visible and scroll.
2. The CSS ‘overflow’ property specifies the action to be taken if the content inside an element overflows.
3. The CSS ‘overflow’ property is present by default.
4. **All of the above**

CSS Position – fixed, static and relative

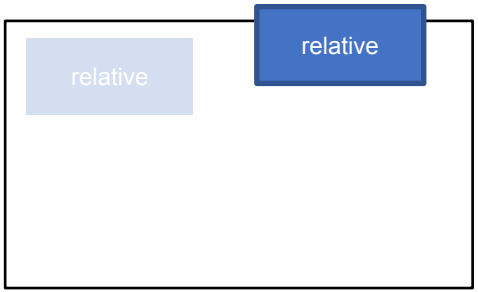
The '**position**' property specifies the positioning that needs to be applied to an element. It could be **fixed**, **static**, **relative**, **absolute** or **sticky**.



HTML elements are positioned statically by default. This means they are positioned according to the flow of the webpage and **cannot** be moved by the top, left, right or bottom properties.



A fixed element will remain fixed to the position that is assigned to it, even if the page is scrolled. You can position the element with respect to the viewport using top, right, bottom or left.

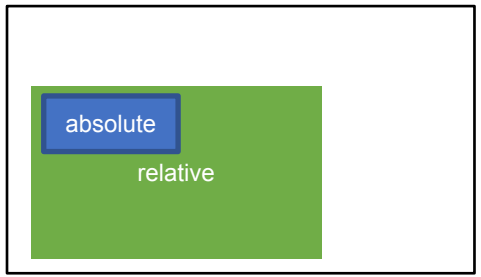


The '*position: relative*' property will position the element relative to its original position. Using the property top, right, left or bottom, you can move the element **away from its original position**.

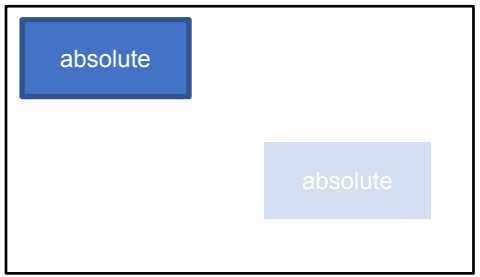
***You can find example code here*

CSS Position – absolute

- For '*position: absolute*', the element can be positioned freely on the space occupied by its **positioned ancestor/parent element**.
- By the positioned ancestor/parent element, we mean that the parent element should be assigned the property as '*position: relative*'.
- If the element with the '*position: absolute*' does not have an immediate parent element having '*position: relative*', then it will check the parent of its parent element for the relative position. It keeps searching all the way up to the top root element.
- If none of the parents have '*position: relative*', then in that case, the element having '*position: absolute*' considers the viewport to be its parent element and positions itself to the top left of the screen, which it considers as (0,0) or the starting point.



Case (a): When the element has a parent with '*position: relative*'.



Case (b): When the element has no parents with '*position: relative*'.

CSS Position – sticky

- An element having '*position: sticky*' toggles between position type '*relative*' and '*fixed*' based on the scrolling position of the user.
- It is positioned as '*relative*' until a given offset position is reached in the viewport and then it behaves as '*fixed*'.

You can read more about it here

Poll 3 (15 Sec)

The default value of CSS position property is:

1. absolute
2. fixed
3. relative
4. static

Poll 3 (Answer)

The default value of CSS position property is:

1. absolute
2. fixed
3. relative
4. **static**

upGrad

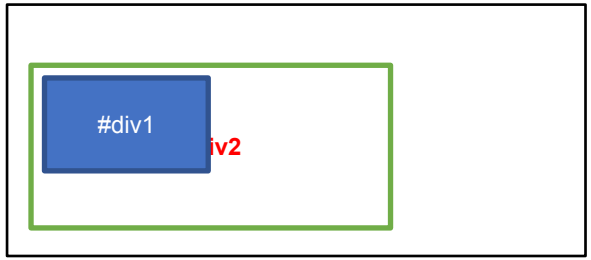
CSS Position – z-index

- The '**z-index**' property specifies the stack order of an element.
- z-index can take positive as well as negative values.
- The **default** value of z-index is: ***auto***
- Basically, it helps the element to be placed in front of or behind other elements.
- ***z-index*** can be applied to all the positioned elements except ***position: static***

CSS Layout

upGrad

CSS Position – z-index



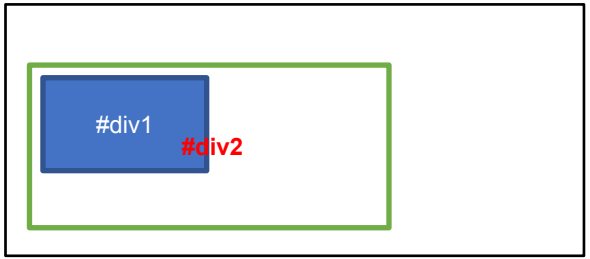

```
<body>
   <div id="div1">#div1</div>
   <div id="div2">#div2</div>
</body>
```

```
#div1 {
  background-color: blue;
  width: 70px;
  height: 70px;
  position: absolute;
  top: 0;
  left: 0;
  z-index: 90;
}
#div2 {
   border: 1px solid green;
   width: 100px;
   height: 100px;
   position: absolute;
   top: 0;
   left: 0;
   z-index: 80;
}
```

CSS Position – z-index

```css
#div1 {
  background-color: blue;
  width: 70px;
  height: 70px;
  position: absolute;
  top: 0;
  left: 0;
  z-index: -1;
}
#div2 {
   border: 1px solid green;
   width: 100px;
   height: 100px;
   position: absolute;
   top: 0;
   left: 0;
   z-index: 0;
}
```

```html
<body>
   <div id="div1">#div1</div>
   <div id="div2">#div2</div>
</body>
```



Poll 4 (15 Sec)

Which of the following statement(s) about z-index is/are true?
(Note: Multiple options can be correct)

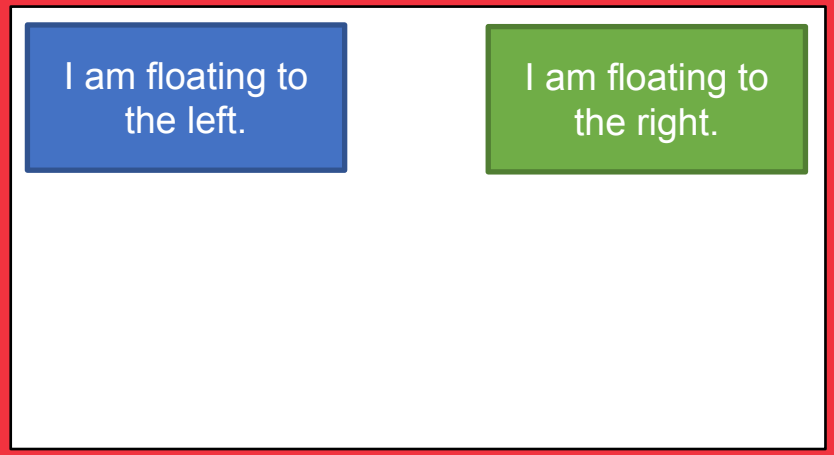
1. The default value of z-index is -1.
2. The default value of z-index is 0.
3. Elements with higher z-indices are stacked above elements with lower z-indices.
4. ‘z-index: auto’ sets the order of the stack equal to the z-indices of the parent elements.

Poll 4 (Answer)

Which of the following statement(s) about z-index is/are true?
(Note: Multiple options can be correct)

1. The default value of z-index is -1.
2. **The default value of z-index is 0.**
3. **Elements with higher z-indices are stacked above elements with lower z-indices.**
4. **‘z-index: auto’ sets the order of the stack equal to the z-indices of the parent elements.**

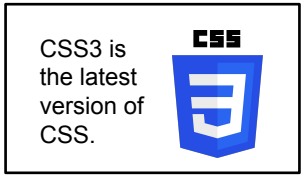
Float and Clear



Float

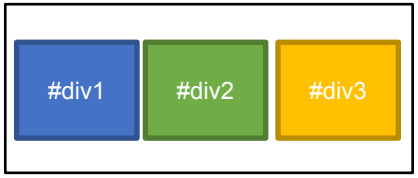
- The CSS '**float**' property specifies how an HTML element should float in its container. It takes the value ***left***, ***right***, ***none*** or ***inherit***.
- It basically helps to position the element inside a container. For example, applying '*float: left*' to an image will make it float to the left of the container.
- Float is typically used to position images and text next to each other. Similarly, it is used to place other elements, like , <section>, etc., next to each other, as if in a row.


```
img {
   float: left;
}
```




```
img {
   float: right;
}
```


```
img {
   float: none;
}
```



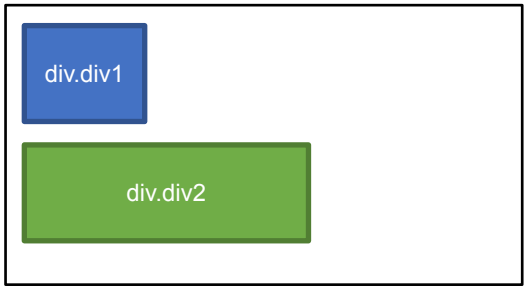

```
div {
   float: left;
}
```

Clear

- The CSS '**clear**' property specifies which elements can float next to the cleared elements.

- The '**clear**' property can take the following values:
  1. **none** – It allows floating elements on both sides of the element. This is the default value.
  2. **left** – It does not allow floating elements on the left side. However, floating elements are allowed to the right.
  3. **right** – It does not allow floating elements on the right side. However, floating elements are allowed to the left.
  4. **both** – It does not allow floating elements on either side of the element.
  5. **Inherit** – It inherits the properties of the parent element.

- It is commonly used right after float. If an element (say, element A) has been floated to the left, the next element (say, element B) should have '*clear: left*'. In this way, element A will stay floated to left, but element B will appear on the next line.



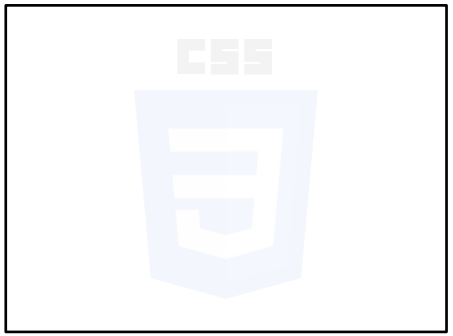

```
.div1 {
   float: left;
}
.div2 {
   clear: left;
}
```

Opacity

- The '**opacity**' property specifies the transparency of an element.
- Its value ranges from 0.0 to 1.0.

```
img {
   opacity: 1;
}
```

```
img {
   opacity: 0.5;
}
```



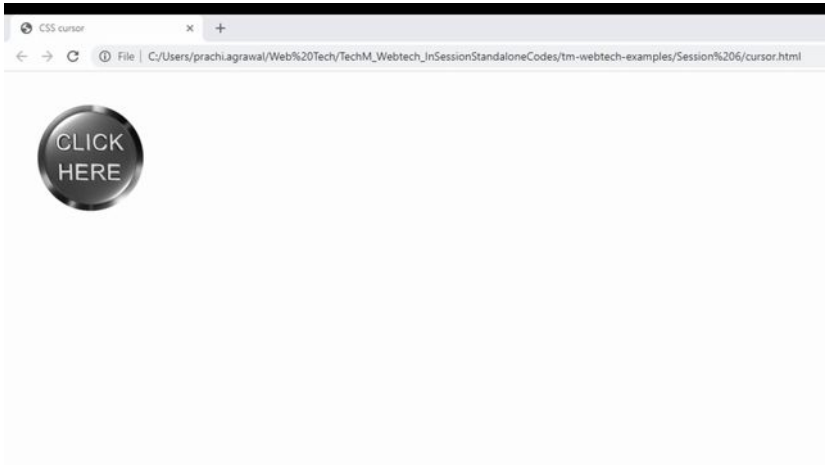
```
img {
   opacity: 0.1;
}
```

CSS Cursor

CSS Cursor

upGrad

You can read more about cursor here.




```
<!DOCTYPE html>
<html>
    <head>
        <title>CSS cursor</title>
        <style>
            .image {
                cursor: pointer;
            }
        </style>
    </head>
    <body>
        <img class="image"
src="https://cdn.pixabay.com/photo/2016/06/29/16/53/click-here-b
utton-1487274_960_720.png" width="200px" height="200px"/>
    </body>
</html>
```

Poll 5 (15 Sec)

Which of the following is a way of positioning an element in CSS?
(Note: More than one option may be correct.)

1. relative
2. static
3. fixed
4. absolute

Poll 5 (Answer)

Which of the following is a way of positioning an element in CSS?
(Note: More than one option may be correct.)

1. **relative**
2. **static**
3. **fixed**
4. **absolute**

Poll 6 (15 Sec)

Which of the following is true about the opacity property of CSS?
(Note: More than one option may be correct.)

1. It takes a value between 0.0 and 1.0.
2. A lower value means more transparency of elements.
3. A lower value means lower transparency of elements.
4. None of the above

Poll 6 (Answer)

Which of the following is true about the opacity property of CSS?
(Note: More than one option may be correct.)

1. **It takes a value between 0.0 and 1.0.**
2. **A lower value means more transparency of elements.**
3. A lower value means lower transparency of elements.
4. None of the above

CSS Combinators

```
<div id='div1'>
    <p>I am a child of 'div1'</p>
</div>
<div>I am a sibling of 'div1'</div>
```

CSS Combinators

upGrad

- A combinator explains the relationship between selectors.
- A CSS selector can contain more than one selector. Combinators help in showing the relationships among multiple selectors.
- The following are the four types of combinators:
    - **Descendant selector** (*space)*
    - **Child selector** (>)
    - **Adjacent sibling selector** (+)
    - **General sibling selector** (~)

CSS Combinators

upGrad

Descendant Selector

- A descendant selector matches all the elements that are descendants of the specified element.
- A descendant selector is signified by the *space* between two selectors.
- The descendant can be at any level

```
div p {
    color: red;
}
```

```
<div>
    <p>This is paragraph 1.</p>
    <p>This is paragraph 2.</p>
    <div>
      <p>This is paragraph 3.</p>
    </div>
</div>
<p>This is paragraph 4.</p>
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

CSS Combinators

upGrad

Child Selector

- A child selector matches all the elements that are the direct children of the specified element. Elements that are not the direct children are not selected.
- A child selector is signified by > between two selectors.

```
div > p {
    color: red;
}
```

```
<div>
    <p>This is paragraph 1.</p>
    <div><p>This is paragraph 2.</p></div>
    <p>This is paragraph 3.</p>
</div>
<p>This is paragraph 4.</p>
<div>
    <p>This is paragraph 5.</p>
</div>
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

This is paragraph 5.

CSS Combinators

upGrad

Adjacent Sibling

- An adjacent sibling selector selects only those elements that are exactly adjacent to the specified element. Adjacent means that the specified element and the adjacent element have the **same parent** and the adjacent element **immediately follows** the specified element.

- A adjacent sibling selector is signified by + between two selectors.

```
div + p {
   color: red;
}
```

```
<p>This is paragraph 1.</p>
<div>
   <p>This is paragraph 2.</p>
</div>
<p>This is paragraph 3.</p>
<p>This is paragraph 4.</p>
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

CSS Combinators

upGrad

General Sibling Selector

- A general sibling selector selects all the specified elements that are after the selected element.
- A general sibling selector is signified by ~ between two selectors.

```
div ~ p {
    color: red;
}
```

```
<p>This is paragraph 1.</p>
<div>
   <p>This is paragraph 2.</p>
</div>
<p>This is paragraph 3.</p>
<p>This is paragraph 4.</p>
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

Poll 7 (15 Sec)

Which of the following is not a CSS combinator?

1. <
2. >
3. ~
4. +

Poll 7 (Answer)

Which of the following is not a CSS combinator?

1. **<**
2. >
3. ~
4. +

upGrad

CSS Pseudo-Class

CSS Pseudo-Classes

upGrad

- A pseudo-class is used to define a state of an element.
- A pseudo-class can be accessed using the ‘:’ combinator.
- You can use the *hover* state for all the elements. Similarly, you can use other pseudo-classes to select/filter a few elements from the entire web page and apply styles specifically to them.

Syntax of a pseudo class:

```
selector:pseudo-class {
   property: value;
}
```

Poll 8 (15 Sec)

For what purpose is a CSS pseudo-class used?

1. To select all the elements that are visited
2. To select all the elements that are active
3. To define a special state of an element
4. To explain relationship between the selectors.

Poll 8 (Answer)

For what purpose is a CSS pseudo-class used?

1. To select all the elements that are visited
2. To select all the elements that are active
3. **To define a special state of an element**
4. To explain relationship between the selectors.

Poll 9 (15 Sec)

Which of the following is the correct syntax of a CSS pseudo-class?

1. selector {property:value;}
2. pseudo-class {property:value;}
3. selector:pseudo-class {property:value;}
4. pseudo-class:selector {property:value;}

Poll 9 (Answer)

Which of the following is the correct syntax of a CSS pseudo-class?

1. selector {property:value;}
2. pseudo-class {property:value;}
3. **selector:pseudo-class {property:value;}**
4. pseudo-class:selector {property:value;}

upGrad

CSS Input Pseudo Classes

CSS Pseudo-Classes

upGrad

- Links have the following four different states:
  - **link -** for an unvisited link
  - **visited -** when the link has been clicked once
  - **hover -** when a user hovers over the link
  - **active -** the moment a link has been clicked

- You can style links and states using properties such as color, font-family, font-size and background.

- ***IMPORTANT: 'a:hover'** must come after '**a:link**' and '**a:visited**'. '**a:active**' must come after '**a:hover**'.*

Let's try out the examples in our code.

***You can find the reference HTML code here and the CSS code here.*

CSS Input Pseudo-Classes

upGrad

:link

:link is a pseudo class that is used to represent the unvisited links.

Remember that the :link pseudo class can be used for the tags that consists of the href attribute.

```
<a href="https://www.upgrad.com/"
target="_blank">Welcome to upGrad</a>
```

```
a:link {
  background-color: yellow;
}
```

Welcome to upGrad

CSS Input Pseudo-Classes

upGrad

:visited

:visited is a pseudo class that is used to represent the already visited links.

```
<a href="https://www.upgrad.com/"
target="_blank">Welcome to upGrad</a>
```

```
a:visited {
  background-color: pink;
}
```

Welcome to upGrad

CSS Input Pseudo-Classes

upGrad

:active

:active is a pseudo class that is used to represent an active element.

An element is active when it is being clicked by a user.

```
<p> When the paragraph is clicked, the paragraph becomes active and the
background color of the paragraph becomes green as well as the text color
becomes white. </p>
```

```
p:active {
   background-color:green;
   color:white;
}
```

When the link is clicked, the paragraph becomes active and the background color of the paragraph becomes green

CSS Input Pseudo-Classes

upGrad

:hover

:hover is a pseudo class that is used to represent an element over which the mouse is hovered.

```
<div>
  <p>
  When the cursor is hovered over the div, the background color changes to red.
  </p>
</div>
```

```
div {
        border: 1px solid black;
        background-color: green;
        cursor: pointer;
        width: 200px;
        height: 200px;
}
div:hover {
        background-color:red;
}
```

CSS Input Pseudo-Classes

upGrad

:focus

:focus is a pseudo class that is used to exert a focus effect on the selected element.

```
<form>
   <label for="firstname">First Name:</label>
   <input id="firstname" type="text" name="firstname">
   <label for="age">Age:</label>
   <input id="age" type="number" name="age">
</form>
```

```
input:focus {
   background-color: yellow;
}
```

CSS Input Pseudo-Classes

upGrad

:checked

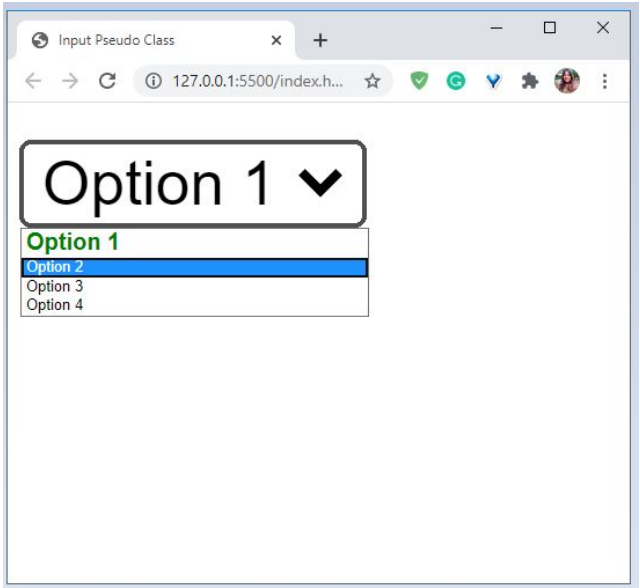
:checked is a pseudo class that is used to determine whether the element is checked or not.

Remember that the :checked pseudo class can be used only in the following cases:

- radio type in <input>
- checkbox type in <input>
- option in <select> element

```
<select>
  <option value="1">Option 1</option>
  <option value="2">Option 2</option>
  <option value="3">Option 3</option>
  <option value="4">Option 4</option>
</select>
```

```
option:checked {
   color: green;
   font-weight: bolder;
   font-size: 20px;
}
```



CSS Input Pseudo-Classes

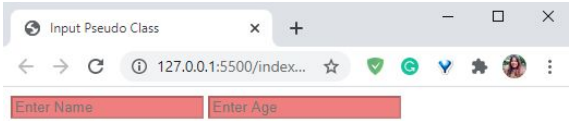
upGrad

:disabled

:disabled is a pseudo class that is used to represent a disabled element.

An element is said to be disabled when a user cannot activate (click, type or select) or accept focus.

```
<input type="text" name="name" placeholder="Enter Name" disabled>
<input type="text" name="age" placeholder="Enter Age" disabled>
```




```
input:disabled {
   background-color: lightcoral;
}
```

CSS Input Pseudo-Classes

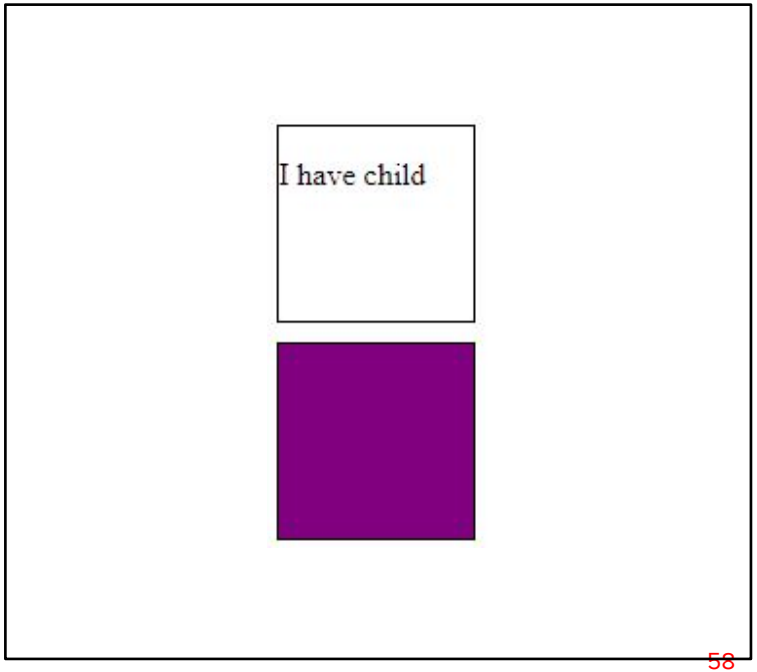
upGrad

:empty

:empty is a pseudo class that is used to represent elements with no children.

```
<div class="is-empty">
  <p>I have child</p>
</div>
<div class="is-empty"></div>
```

```
.is-empty {
   border: 1px solid black;
   height: 100px;
   width: 100px;
   margin: 10px;
}
.is-empty:empty {
   background-color: purple;
}
```



Poll 10 (15 Sec)

Which of the following is not a pseudo-class for the CSS anchor tag?

1. visited
2. unvisited
3. hover
4. active

Poll 10 (Answer)

Which of the following is not a pseudo-class for the CSS anchor tag?

1. visited
2. **unvisited**
3. hover
4. active

CSS Pseudo-Classes

upGrad

:first-child

The **:first-child** pseudo class matches the first child of any specified element. For example, if we write *p:first-child*, then it will match with every *<p>* that is the first child of any element, i.e., it should be the first element among the group of sibling elements.

```
<p>This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<div>
   <p>This is paragraph 3.</p>
   <p>This is paragraph 4.</p>
</div>
<p>This is paragraph 5.</p>
```

```
p:first-child {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

This is paragraph 5.

CSS Pseudo-Classes

upGrad

:first-of-type

The **:first-of-type** matches with the **first occurrence of the selected element**. This element need not be the first one in the container but should be the first one of its type. For example, in the example given below, *<h4>* is the third element in * * but is the first of its type.

```
<p>This is paragraph 1.</p>
<div>
   <p>This is paragraph 2.</p>
   <p>This is paragraph 3.</p>
   <h4>This is h4 - 1.</h4>
   <h4>This is h4 - 2.</h4>
</div>
<h4>This is h4 - 3.</h4>
<p>This is paragraph 4.</p>
<h4>This is h4 - 4.</h4>
```

```
h4:first-of-type {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is h4 – 1.

This is h4– 2.

This is h4 – 3.

This is paragraph 4.

This is h4 – 4.

Difference between :first-child and :first-of-type can be read here

CSS Pseudo-Classes

upGrad

:last-child

The **:last-child** pseudo class matches the last child of any specified element. For example, if you write *p:last-child*, then it will match with every *<p>* that is the last child of any element, i.e., it should be the last element.

```
<p>This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<div>
   <p>This is paragraph 3.</p>
   <p>This is paragraph 4.</p>
</div>
<p>This is paragraph 5.</p>
```

```
p:last-child {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

This is paragraph 5.

CSS Pseudo-Classes

upGrad

:last-of-type

:last-of-type pseudo class matches with the **last occurrence of the selected element**. This element need not be the last one in the container but should be the last one of its type. For example, in the example given below, *<h4>* is not at the end of * * but is the last one of its type.

```
<p>This is paragraph 1.</p>
<div>
   <p>This is paragraph 2.</p>
   <h4>This is h4 - 1.</h4>
   <h4>This is h4 - 2.</h4>
   <p>This is paragraph 3.</p>
</div>
<h4>This is h4 - 3.</h4>
<p>This is paragraph 4.</p>
<h4>This is h4 - 4.</h4>
```

```
h4:last-of-type {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is h4 – 1.

This is h4 – 2.

This is paragraph 3.

This is h4 – 3.

This is paragraph 4.

This is h4 – 4.

CSS Pseudo-Classes

upGrad

:not(*element*)

:not(*element*) pseudo class selects all the elements that are not specified in the bracket. For instance, in the following example, all the elements that are not <*p*> will be selected and assigned the color red.

```
<p>This is paragraph 1.</p>
<div>
   This is some text inside a div.
</div>
<h4>This is a h4.</h4>
<p>This is paragraph 2.</p>
<span>This is span.</span>
```

```
:not(p) {
   color: red;
}
```

This is paragraph 1.

This is some text inside a div.

This is a h4.

This is paragraph 2.

This is span.

CSS Pseudo-Classes

upGrad

:nth-child

In the **:nth-child(n)** pseudo class, you need to provide a number inside the brackets. Then, CSS picks up the following elements:

a) a specified element
b) a child element
c) the nth position element is picked up and provided in the bracket

For instance, in the following example, n is 2. So, every second *<p>*, i.e., the element *<p>*, which is positioned second as a child in any container, will be selected.

```
<p>This is paragraph 1.</p>
<p>This is paragraph 2.</p>
<div>
   <p>This is paragraph 3.</p>
   <p>This is paragraph 4.</p>
</div>
<p>This is paragraph 5.</p>
```

```
p:nth-child(2) {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

This is paragraph 5.

CSS Pseudo-Classes

upGrad

:nth-last-child

In the **:nth-last-child(n)** pseudo class, you need to provide a number inside the brackets. Then, CSS picks up the following elements:

a) the specified element
b) a child element
c) lemenent at the nth position from the last occurrence is provided in the bracket

For instance, in the following example, n is 2. So, every second-last *<p>,* i.e., the element *<p>,* which is positioned second from the last as a child in any container, will be selected.

```
<p>This is paragraph 1.</p>
<div>
   <p>This is paragraph 2.</p>
   <p>This is paragraph 3.</p>
   <p>This is paragraph 4.</p>
</div>
```

```
p:nth-last-child(2) {
   color: red;
}
```

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

CSS Pseudo-Classes

upGrad

:nth-of-type

:nth-of-type(n) pseudo-class matches with the **nth occurrence of the selected element**. It will select an element of the mentioned type, which occurs for the nth time.

```
<p>This is paragraph 1.</p>
<div>
   <h4>This is h4 - 1.</h4>
   <p>This is paragraph 2.</p>
   <h4>This is h4 - 2.</h4>
   <p>This is paragraph 3.</p>
</div>
<h4>This is h4 - 3.</h4>
<p>This is paragraph 4.</p>
<h4>This is h4 - 4.</h4>
```

```
h4:nth-of-type(2) {
   color: red;
}
```

This is paragraph 1.

This is h4 – 1.

This is paragraph 2.

This is h4 – 2.

This is paragraph 3.

This is h4– 3.

This is paragraph 4.

This is h4 – 4.

CSS Pseudo-Classes

upGrad

:nth-last-of-type

:nth-last-of-type(n) pseudo-class matches with the **nth occurrence of the selected element from the last**. It will select an element of the mentioned type, which occurs for the nth time from the last occurrence.

```
<p>This is paragraph 1.</p>
<div>
   <h4>This is h4 - 1.</h4>
   <p>This is paragraph 2.</p>
   <h4>This is h4 - 2.</h4>
   <p>This is paragraph 3.</p>
</div>
<h4>This is h4 - 3.</h4>
<p>This is paragraph 4.</p>
<h4>This is h4 - 4.</h4>
```

```
h4:nth-last-of-type(2) {
   color: red;
}
```

This is paragraph 1.

This is h4 – 1.

This is paragraph 2.

This is h4– 2.

This is paragraph 3.

This is h4 – 3.

This is paragraph 4.

This is h4 – 4.

upGrad

Poll 11 (15 Sec)

What is the output of the CSS command/tag in the adjoining image

1. Selects every even child of the element.
2. Selects the second child of the element.
3. Selects the odd child of the element.
4. Throws an error.

```
div:nth-child(even) {
   color:blue
}
```

Poll 11 (Answer)

What is the output of the CSS command/tag in the adjoining image

1. **Selects every even child of the element.**
2. Selects the second child of the element.
3. Selects the odd child of the element.
4. Throws an error.

```
div:nth-child(even) {
   color:blue
}
```

Poll 12 (15 Sec)

Which of the following pseudo-class selector is used to check whether the element is checked or not?

1. selector:active
2. selector:focus
3. selector:checked
4. selector:empty

Poll 12 (Answer)

Which of the following pseudo-class selector is used to check whether the element is checked or not?

1. selector:active
2. selector:focus
3. **selector:checked**
4. selector:empty

upGrad

CSS Pseudo Elements

Once upon a time…

CSS Pseudo-Elements

upGrad

Introduction to Pseudo-Elements

- CSS pseudo elements, unlike CSS pseudo classes, are used to style specific parts of an element instead of filtering them.
- For instance, they can be used to style the first letter or the first line of an element.
- Additionally, you can also place the content before or after the element.
- Pseudo elements can be accessed using the **‘::’** combinator.
- The primary difference between pseudo classes and pseudo elements is that pseudo classes are used to **filter the elements and apply styles to the filtered elements**, whereas pseudo elements are used to **apply styles to specific parts of an element’s content.**

The syntax of a pseudo element is as follows:

```
selector::pseudo-element {
   property: value;
}
```

CSS Pseudo-Elements

upGrad

::first-line

- The **::first-line** pseudo-element can be applied only to block elements.
- As the name suggests, the ::first-line property applies style only to the first line of the element.
- The following styles can be applied using the ::first-line property.

```
p::first-line {
    color: white;
    background-color: black;
}
```

The first line property is applied to this paragraph element. Only the first line is affected.

CSS Pseudo Elements

upGrad

::first-letter

- The **::first-letter** pseudo-element can be applied only to block elements.
- As the name suggests, the ::first-letter property applies style only to the first letter of the element.
- The following styles can be applied using the ::first-letter property.

```
p::first-letter {
   color: red;
   font-size: 2rem;
}
```

The first line property is applied to this paragraph element. Only the first line is affected.

CSS Pseudo Elements

upGrad

::before and ::after

- The **::before** and **::after** pseudo-elements are used to add content before and after an element, respectively.
- You can use the **content** property to add content.

```
p::before {
    content: url('css3.png');
}

p::after {
    content: 'Hi';
}
```

```
<p>Message</p>
```

Poll 13 (15 Sec)

What will be the output of the CSS code in the adjoining image?

1. The entire paragraph will be colored red and has a font size of 30px.
2. The first character of the paragraph will be colored green and has a size of 60px. The first line will be colored red and has a size of 30px. The remaining paragraph will have the default style.
3. The entire paragraph will be colored green and has a font size of 60px.

```
p::first-line {
    color: red;
    font-size: 30px;
}

p::first-letter {
    color: green;
    font-size: 60px;
}
```

Poll 13 (Answer)

What will be the output of the CSS code in the adjoining image?

1. The entire paragraph will be colored red and has a font size of 30px.
2. **The first character of the paragraph will be colored green and has a size of 60px, and the first line will be colored red and has a size of 30px. The remaining paragraph will have the default style.**
3. The entire paragraph will be colored green and has a font size of 60px.

```
p::first-line {
      color: red;
      font-size: 30px;
}

p::first-letter {
      color: green;
      font-size: 60px;
}
```

upGrad

Project Work

(Let us add some more CSS to our project.)

You can refer to the solution here.

Hands-On Exercise 1 (3 mins)

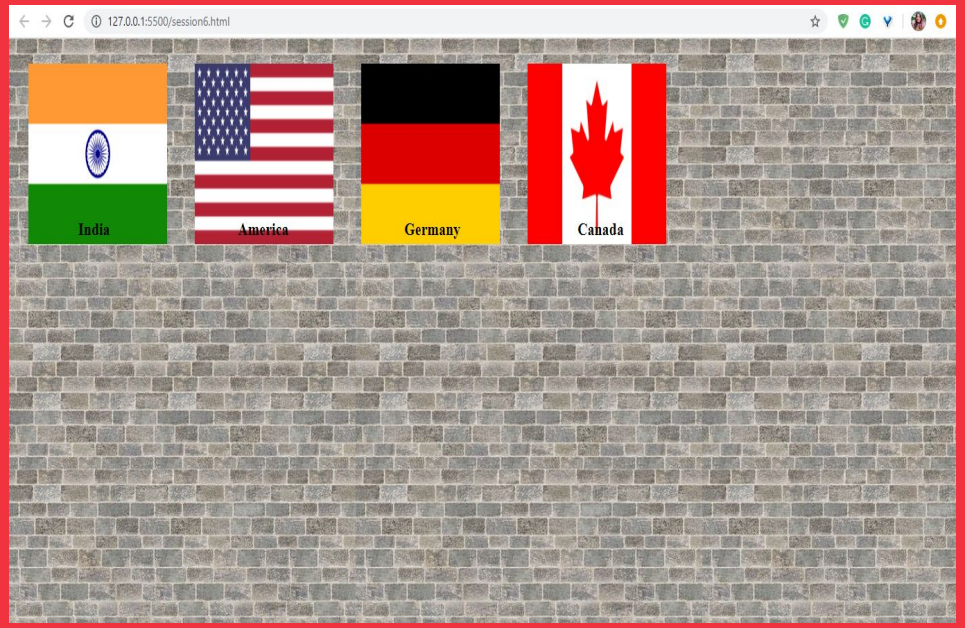
upGrad

Write a CSS code for the given HTML code such that the output is as shown in the screenshot to the right.

- The background image in the body is: https://images.pexels.com/photos/220182/pexels-photo-220182.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500
- When hovered over each flag div, the cursor should be of type pointer.
- The font size of the country names is 20px.
- The images should be aligned using the 'float' property.

You can find the stub code here.

You can find the solution here.



Key Takeaways

- The CSS outline is the line outside the margin of the element.
- The display and visibility properties in CSS.
- The position property in CSS and its application.
- The ‘float’ property allows elements to be placed beside each other.
- CSS combinators allow you to select multiple selectors. Note that ‘ ’ (space) is used for descendants, ‘>’ is used for children, ‘+’ is used for adjacent siblings and ‘~’ is used for general siblings.
- A pseudo-class (specified by ‘:’) is used to denote the states of elements or to find the nth element. Pseudo-elements (specified by ‘::’) are used to change a specific part of an element, such as the first line or the first word.

The following tasks are to be completed after today's session:

MCQs
Coding Questions
Project - Checkpoint 6

In the next class, we will discuss...

- What is ‘flexbox’ and why is it the most sought after CSS property?
- How to place elements in a grid using CSS?

upGrad

#RahoAmbitious

Thank you!