



Fundamentals
of **WEB
DESIGN**

Float, Position & Layers

Cascading Style Sheets

float Property

- The CSS "float" property is used to wrap content to the left or right of an image or other object.
- In HTML, the align attribute was used with the tag to produce the same results.

In this Example, the "float" property is set to left within an tag. The content will wrap next to the image until it reaches the bottom of the image, then it will wrap under the image.

```
<img src = "images/flowers.gif" width="158" height="150"  
style="float:left; margin: 15px 15px 15px 15px">
```



This paragraph is wrapping around a flower basket positioned to the left.

This second paragraph is also wrapping around a flower basket

positioned to the left. By default, text will continue to wrap next to the image until the content reaches the bottom of the image, then it will wrap under the image.

float Property

- In this Example, the "float" property is set to right within an tag. Again, the content will wrap next to the image until it reaches the bottom of the image, then it will wrap under the image.

```
<img src = "images/flowers.gif" width="158" height="150"  
style="float:right; margin: 15px 15px 15px 15px">
```

This paragraph is wrapping around a flower basket positioned to the right.



This second paragraph is also wrapping around a flower basket positioned to the right. By default, text will continue to wrap next to the image until the content reaches the bottom of the image, then it will wrap under the image.

clear Property

- The CSS "clear" property is used to force wrapping to stop even when you have not reached the bottom of the image.
- This property can have values of left, right, and both.
- In this example, the first paragraph wraps to the left of the image, but the second paragraph starts below the image. This is because the "clear" property has been set to right for that paragraph.

```
<p style="font-size: 14pt; clear: right">
```

This paragraph is wrapping around a flower basket positioned to the right.



This second paragraph uses the "clear" property to force wrapping to halt and thus the second paragraph starts below the image.

Overflow Content

- Overflow content is any content that will not fit inside the original object defined to hold it.
- For example, if you create a `<div>` object with the width undefined and the height undefined, then you add this content to it, the object will increase in width and/or height to make room for the extra content.

```
<div style="background-color: #ceffce;  
        border   : solid 1px #000000;  
        color    : #000000;  
        padding  : 8px;  
        position: absolute;  
        left     : 50px;  
        top      : 50px;">
```

Overflow Property Set to Visible

- You can not predictably control how much space content will take up within all browsers. The CSS "overflow" property lets you determine how overflow content should be handled.
- In the following examples, the width and height have been set to a size that is too small to hold the content within. In this example, the "overflow" property is set to visible.

```
<div style="background-color: #ccffcc;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 50px;  
top : 50px;  
width : 200px;  
height : 100px;  
overflow: visible;">
```

Layer 0 (The browser is the default layer)

About Overflow Content:

Overflow content is any content that will not fit inside the original object defined to hold it. For example, if you create a <div> object with the width undefined and the height undefined, then you add this content to it, the object will increase in width and/or height to make room for the extra content.

Overflow Set to Hidden

- With the "overflow" property set to "hidden", any extra content is simply cropped or cut off to fit within the space provided.
- This option is not likely to yield very desirable results.

```
<div style="background-color: #ccffcc;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 50px;  
top : 50px;  
width : 200px;  
height : 100px;  
overflow: hidden;">
```

Layer 0 (The browser is the default layer)

About Overflow Content:

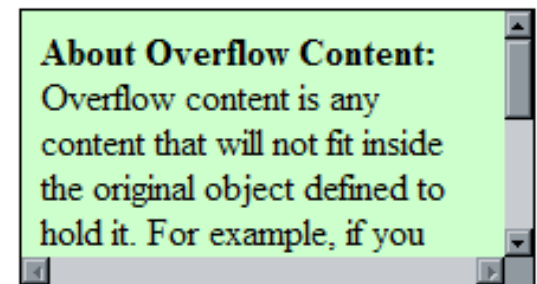
Overflow content is any content that will not fit inside the original object defined to hold it. For example, if you create a <div> object with the width undefined

Overflow Set to Scroll

- With the "overflow" property set to "scroll", any extra content is accessible using scroll bars that are added to the layer on both the right and the bottom whether they are needed or not.

```
<div style="background-color: #ccffcc;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 50px;  
top : 50px;  
width : 200px;  
height : 100px;  
overflow: scroll;">
```

Layer 0 (The browser is the default layer)

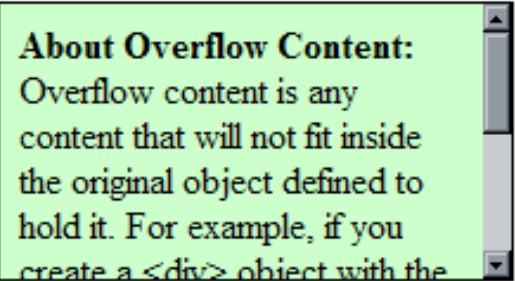


Overflow Set to Auto

- With the "overflow" property set to "auto", any extra content is accessible using scroll bars that are added to the layer on the right and/or bottom only as needed.

```
<div style="background-color: #ccffcc;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 50px;  
top : 50px;  
width : 200px;  
height : 100px;  
overflow: auto;">
```

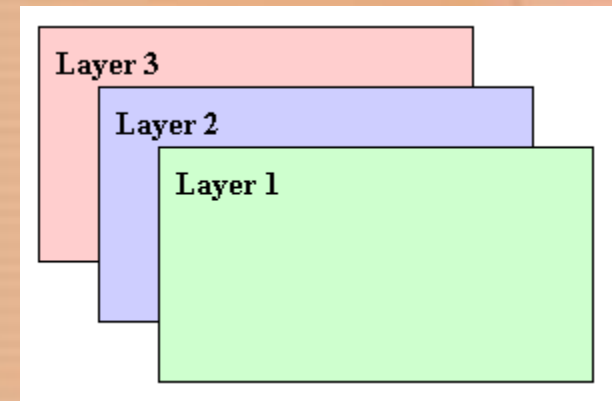
Layer 0 (The browser is the default layer)



About Overflow Content:
Overflow content is any content that will not fit inside the original object defined to hold it. For example, if you create a `<div>` object with the

Layers

- Layers can be thought of as transparencies used on an overhead projector. Each layer is stacked one on top of the other, and everything in the top layer is visible, while lower layers may be partially blocked by the upper layers.
- Layers allow you to define content that can be placed anywhere on a Web page and can overlap, if you desire.
- In this example there are three layers. You will notice that when a layer is at the top, it covers the lower layers where they overlap.



Layers

- In standard HTML, every object created is part of the default layer.
- The Web page displayed within a browser window is considered to be the default layer.
- Normally, when you create an object using the `<div>` tag, it is applied directly to the Web page like any other tag, thus becoming part of the default layer for that page. The division `<div>` tag is used to define block elements on a Web page.
- But if you add CSS position properties to an object such as the `<div>` tag, you can create a new layer.
- This new layer can be positioned with either **relative** or **absolute** coordinates.

Position Within The Browser Window

- By default, your browser window is layer 0 and all new layers are anchored to it using the "left" and "top" properties to define the location within your browser window to place the new layer.
- **Absolute** positioning within the browser window means you are defining a new layer whose starting location is determined by measuring from the top left corner of the browser window.

Position Within The Browser Window

- In the example below, two layers are defined using the <div> tag. To use absolute positioning, the position property is set to absolute and the top and left properties are defined for each <div> .
- By default, layers defined later sit on top of layers defined earlier.

```
<div style="background-color: #ceffcc;  
        border : solid 1px #000000;  
        color  : #000000;  
        padding : 8px;  
        position: absolute;  
        left   : 10px;  
        top    : 110px;  
        width  : 200px;  
        height : 100px;">  
    <b>Layer 1</b><br />  
<br />  
</div>
```

```
<div style="background-color: #ceefff;  
        border : solid 1px #000000;  
        color  : #000000;  
        padding : 8px;  
        position: absolute;  
        left   : 40px;  
        top    : 140px;  
        width  : 200px;  
        height : 100px;">
```

Nested Layers

- Absolute positioning can also occur within an existing layer.
- This happens automatically if you define a new layer from within an existing layer. In this case the new layer will have an absolute location offset from the top left corner of the containing layer instead of the browser.

Nested Layers

- In the example below, two layers are defined using the `<div>` tag. The second layer is defined within the first layer, so its absolute position is from the top left corner of layer one.

```
<div style="background-color: #ffcccc;
border : solid 1px #000000;
color : #000000;
padding : 8px;
position: absolute;
left : 70px;
top : 70px;
width : 200px;
height : 100px;">
<b>Layer 1</b><br />
left : 70px;<br />
top : 70px;<br />
<br />
<div style="background-color: #ccccff;
border : solid 1px #000000;
color : #000000;
padding : 8px;
position: absolute;
left : 70px;
top : 70px;
width : 200px;
height : 100px;">
```

Relative placement of a layer

- **Relative** positioning within the browser window means you are defining a new layer whose location is determined relative to its current position within the content. The advantage to using relative positioning is that as the content grows or moves, so does the layer, because the layer is anchored to that specific location within the content.
- To use relative positioning, the position property is set to relative and the top and left properties define the offset from the current position.

Relative placement of a layer

- In the example below, layer one is defined as relative with no offset from the current position and layer two as absolute layers nested within layer one.

```
<div style="background-color: #ccefcc;
border : solid 1px #000000;
color : #000000;
padding : 8px;
position: relative;
left : 0px;
top : 0px;
width : 200px;
height : 100px;
z-index : 200;">
```

Layer 1
 (A relative positioned layer)

```
<div style="background-color: #ccccff;
border : solid 1px #000000;
color : #000000;
padding : 8px;
position: absolute;
left : 70px;
top : 70px;
width : 200px;
height : 100px;">
```

z-index Property

- The CSS "z-index" property is used to control the order of layers.
- The "z-index" property is set to a number value from 1 to 900 where the higher the number, the higher up in the stack a layer is placed.

z-index Property

- In the example below, layer one is defined first with a z-index of two, which is higher than the other two layers, thus, it displays on top of the other layers.

```
<div style="background-color: #ccccff;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 40px;  
top : 140px;  
width : 200px;  
height : 100px;  
z-index : 2">
```

```
<b>Layer 2</b><br />  
<br />  
</div>
```

```
<div style="background-color: #ffcccc;  
border : solid 1px #000000;  
color : #000000;  
padding : 8px;  
position: absolute;  
left : 10px;  
top : 110px;  
width : 200px;  
height : 100px;  
z-index : 1">
```