

# Specifying Specific Elements

# Div element

## HTML

```
<div id="container">  
    The sky is blue.  
</div>
```

## CSS

```
#container {  
    width: 215px;  
    height: 190px;  
    color: rgb(239, 65, 35);  
    background-color: white;  
    padding: 5px;  
}
```

# Div element

## HTML

```
<div id="container">  
  The sky is blue.  
</div>
```

## CSS

```
#container {  
  width: 215px;  
  height: 190px;  
  color: #000000;  
  background-color: white;  
  padding: 5px;  
}
```

The sky is blue.

# Class

You can assign **multiple elements** in an HTML document names which can then be styled via CSS with the class attribute.

```
<p class="sidebar">  
    This is sidebar paragraph  
    touting the value of the  
    class  
    attribute.  
</p>
```

```
<p class="sidebar">  
    And the next paragraph.  
</p>
```

# Class

In your CSS document all class names are preceded by a period (.)

```
.sidebar {  
    font-size: 18px;  
    font-weight: bold;  
}
```

# Id

You can assign **single elements** in an HTML document names which are styled via CSS with the id attribute. Attribute values must only occur once in the document.

```
<p id="artist">  
    Pablo Picasso  
</p>
```

```
<p id="medium">  
    Oil painting  
</p>
```

# Id

In your CSS document all id names are preceded by a hashtag ( # )

```
#artist {  
    background-color: red;  
}
```

```
#medium {  
    color: green;  
}
```

# Span element

## HTML

```
<p> The sky is <span id= "hello"> blue </span> and yeah. </p>
```

## CSS

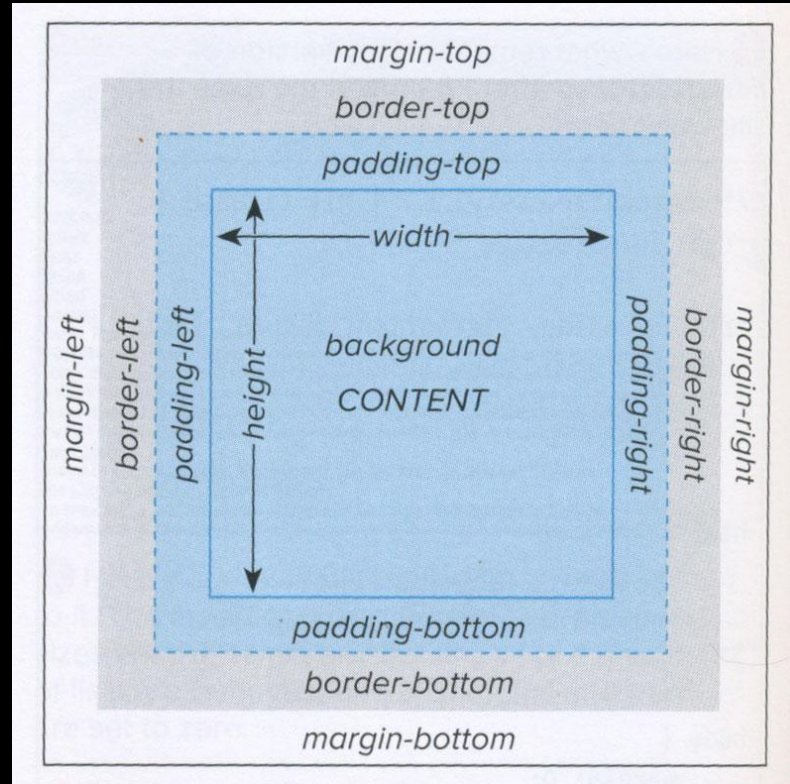
```
#hello {  
    color: blue;  
}
```



**For more CSS references go to:**

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

## Box Model



# Adjusting box properties

```
#hello {  
    width: 300px;  
    height: 300px;  
    padding: 10px;  
    border: 2px solid #0088dd;  
    margin: 10px auto 10px auto;  
}
```

# Adjusting box properties

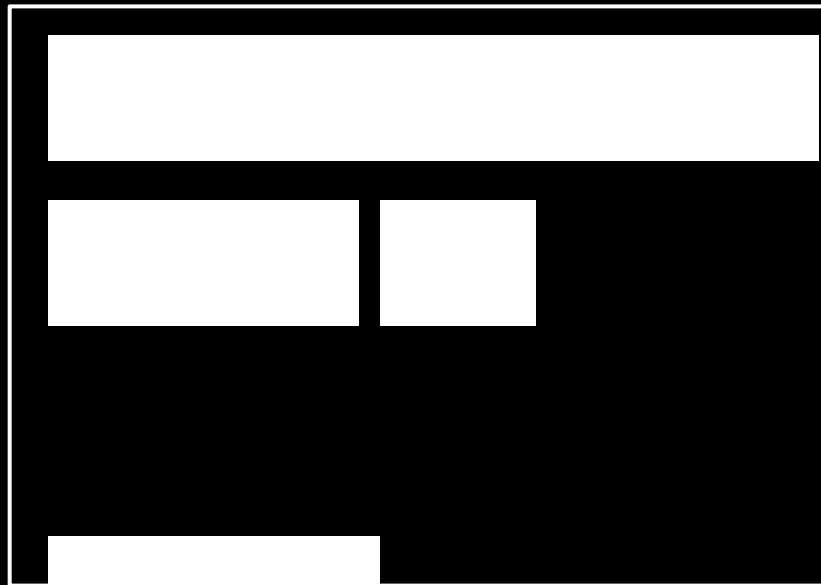
```
#hello {  
    width: 300px;  
    height: 300px;  
    padding-right: 10px;  
    Shorthand : border: 2px solid #0088dd;  
               margin: 10px auto 10px auto;  
               border:  
}
```

# Adjusting box properties

```
Individual  {
    width: 300px;
    height: 300px;
    padding-right: 10px;
    border: 2px solid #0088dd;
    margin: 10px auto 10px auto;
}
```

# Normal Flow

In the normal flow of a webpage, elements stack on top of each other vertically, from the top of the viewport down.



# Block Level Elements

Block level elements are blocks that will take up the full width of whatever contains them by default.

Examples:

**div, p, ul, li, h1, section**



# Inline Level Elements

In-line elements occupy only the space bounded by the tags that define them. They are still boxes (i.e. follow box model properties), but do not stack.

Examples:

**span, em, strong, img**





# CSS Floats

Use float when you are displaying content on a page that can change. Written copy on a page will be of different lengths. Use float where the placement of elements in relation to each other is important and changing. If your content interacts with itself, use floats.

```
#sidebar {  
    float: left;  
}
```

other float values:

```
float: right;  
float: none;  
float: inherit;
```

# Float

A floated box is rendered within the normal document flow, is shifted to the left or right as far as is possible within the constraints of the element it is contained by. It is somewhat analogous to a print layout where text wraps around an image in a column. In HTML/CSS the image is floated and the text responds to that (by filling the “empty” space next to the image).

>Lorem ipsum dolor sit  
amet, consectetur adipi-  
scing elit. Vestibulum  
dictum, lacus nec ultri-  
cies feugiat, sem enim  
venenatis odio, non consequat nibh  
odio ut urna. Pellentesque commodo  
enim est, sed vestibulum urna aliquam  
eget. Donec adipiscing semper feugiat.  
Morbi vehicula ligula quis vestibulum  
pellentesque. Integer at mi in elit  
tristique laoreet. Lorem ipsum dolor  
sit amet, consectetur adipiscing elit.  
Aenean nulla dolor, lacinia et gravida  
quis, suscipit a sapien. Pellentesque  
dapibus sem vitae sodales faucibus.  
Sed aliquet pellentesque erat non  
porta. Donec a nulla tincidunt, fau-  
cibus risus in, lacinia erat. Quisque  
vestibulum leo vel convallis sodales.  
Sed vel dignissim velit.



# Float Example



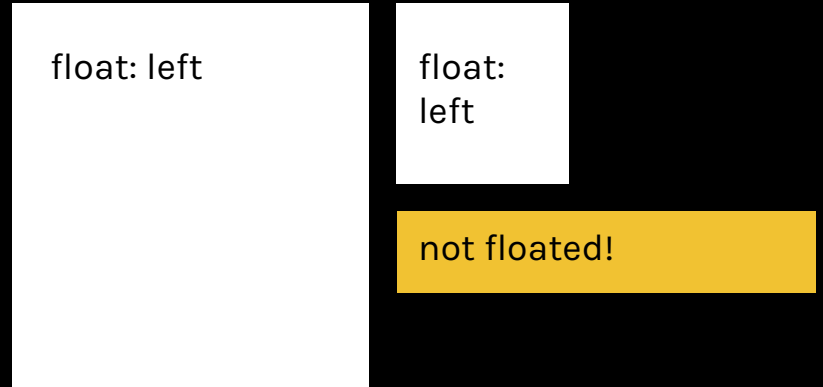
```
.column {  
    float: left;  
}
```

# Clear

An element that has the **clear** property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float.

# Clear

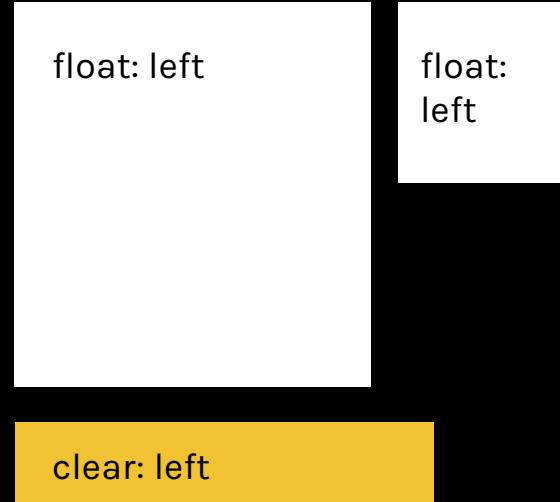
An element that has the **clear** property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float.



# Clear

An element that has the **clear** property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float.

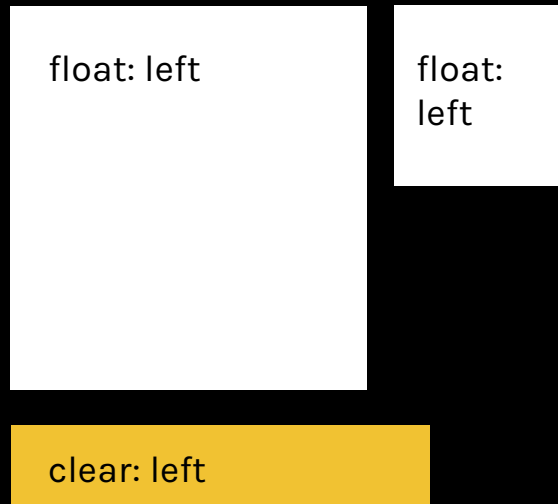
```
#footer {  
    clear: left;  
}
```



# Clear

An element that has the **clear** property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float.

```
#footer {  
    clear: left;  
}
```



other clear values:  
**left, right, inherit,**  
**both, none.**

**For more on CSS Floats go to:**

**<http://alistapart.com/article/css-floats-101>**



Exercise #5:

# CSS Floats