# Change the Color of Text

<h2 style="color: blue">CatPhotoApp</h2>

# Change the Font Size of an Element

h1 {  
  font-size: 30px;  
}

# Style Multiple Elements with a CSS Class

.blue-text {  
  color: blue;  
}

<h2 class="blue-text">CatPhotoApp</h2>

# Import a Google Font

<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">

h2{

font-family: Lobster;

}

# Specify How Fonts Should Degrade

For example, if you wanted an element to use the Helvetica font, but also degrade to the Sans-Serif font when Helvetica wasn't available, you could use this CSS style:

p {  
  font-family: Helvetica, Sans-Serif;  
}

# Add Images to your Website

<img src="https://www.your-image-source.com/your-image.jpg" alt="Author standing on a beach with two thumbs up. ">

The text inside an altattribute is used for screen readers to improve accessibility and is displayed if the image fails to load.

# Size your Images

<style>  
  .larger-image {  
    width: 500px;  
  }  
</style>

# Add Borders Around your Elements

<style>  
  .thin-red-border {  
    border-color: red;  
    border-width: 5px;  
    border-style: solid;  
  }  
</style>

# Apply multiple classes to an element

<img class="class1 class2">

# Add Rounded Corners with a Border Radius

.thick-green-border {

border-color: green;

border-width: 10px;

border-style: solid;

**border-radius:10px;**

}

# Make Circular Images with a Border Radius

**border-radius:50%;**

# Link to External Pages with Anchor Elements

Anchor : a piece of heavy metal that is lowered to the bottom of the sea, lake etc to prevent a ship or boat moving

# Nest an Anchor Element within a Paragraph

<p>

<a href=” ”>

</a>

</p>

# Make Dead Links using the Hash Symbol

Replace the value of your a element's href attribute with a #, also known as a hash symbol, to turn it into a dead link.

<p>Click here for <a href="#">cat photos</a>.</p>

# Turn an Image into a Link

<a href="#">

<img

src="https://bit.ly/fcc-running-cats"

alt="Three kittens running towards the camera. "

>

</a>

# Create a Bulleted Unordered List

<ul>  
  <li>milk</li>  
  <li>cheese</li>  
</ul>

# Create an Ordered List

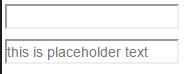
<ol>  
  <li>Garfield</li>  
  <li>Sylvester</li>  
</ol>

# Create a Text Field

<input type="text">

# Add Placeholder Text to a Text Field

<input type="text" placeholder="this is placeholder text">



# Create a Form Element

<form action="/url-where-you-want-to-submit-form-data">

<input type=”text”>

</form>

# Add a Submit Button to a Form

<form action="#">

<button type="submit"></button>

</form>

# Use HTML5 to Require a Field

<input type="text" required>

# Create a Set of Radio Buttons

<label><input type="radio" name="indoor-outdoor"> Indoor</label>

# Create a Set of Checkboxes

<label><input type="checkbox" name="personality"> Loving</label>

<label><input type="checkbox" name="personality"> Lazy</label>

<label><input type="checkbox" name="personality"> Energetic</label>

# Check Radio Buttons and Checkboxes by Default

<input type="radio" name="test-name" checked>

# Nest Many Elements within a Single Div Element

<div>

</div>

# Give a Background Color to a Div Element

.green-background {  
  background-color: green;  
}

# Set the ID of an Element

<h2 id="cat-photo-app">

# Use an ID Attribute to Style an Element

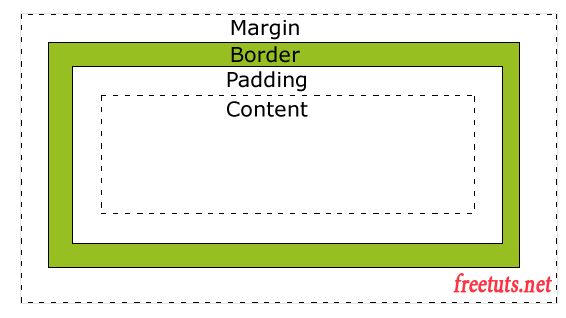
#cat-photo-element {  
  background-color: green;  
}

# Adjusting the Padding of an Element

padding, margin, border.

Margin : khoảng cách giữa 2 thẻ HTML.

Padding : khoảng cách giữa thẻ HTML và nội dung.



# Adjust the Margin of an Element

.green-box {

background-color: green;

padding: 20px;

margin: 20px;

}

# Add a Negative Margin to an Element’

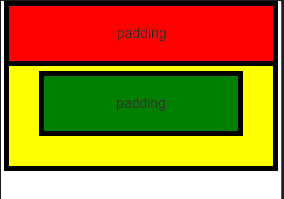
.red-box {

background-color: red;

padding: 20px;

***margin: -15px;***

}



# Add Different Padding to Each Side of an Element

.red-box {

background-color: red;

***padding-top: 40px;***

***padding-right: 20px;***

***padding-bottom: 20px;***

***padding-left: 40px;***

}

.green-box {

background-color: green;

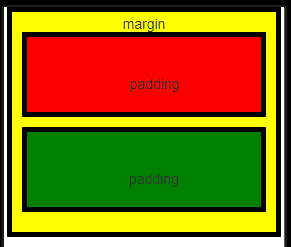
***padding-top:40px;***

***padding-left:40px;***

***padding-bottom:20px;***

***padding-right:20px;***

}



# Add Different Margins to Each Side of an Element

.yellow-box {

background-color: yellow;

padding: 10px;

}

.red-box {

background-color: red;

**margin-top: 40px;**

**margin-right: 20px;**

**margin-bottom: 20px;**

**margin-left: 40px;**

}

.green-box {

background-color: green;

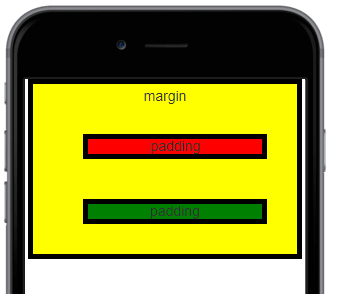
**margin-top:40px;**

**margin-left:40px;**

**margin-bottom:20px;**

**margin-right:20px;**

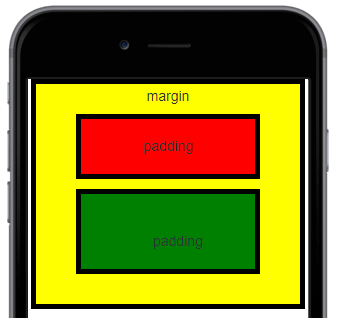
}



# Use Clockwise Notation to Specify the Padding of an Element

padding: 10px 20px 10px 20px;

These four values work like a clock: top, right, bottom, left (clockwise)



.yellow-box {

background-color: yellow;

padding: 20px 40px 20px 40px;

}

.red-box {

background-color: red;

padding: 20px 40px 20px 40px;

}

.green-box {

background-color: green;

padding: 40px 20px 20px 40px;

# Style the HTML Body Element

<style>

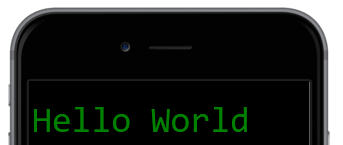
body {

background-color: black;

}

</style>

# Inherit Styles from the Body Element



<style>

body {

background-color: black;

color:green;

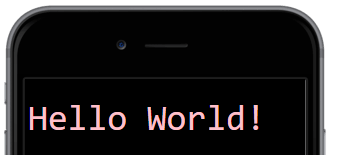
font-family:Monospace;

}

</style>

<h1>Hello World</h1>

# Prioritize One Style Over Another



Our "pink-text" class overrode our bodyelement's CSS declaration!

<style>

body {

background-color: black;

font-family: Monospace;

color: green;

}

**.pink-text{ //override (color: green) in body{}**

**color:pink;**

**}**

</style>

<h1 class="pink-text">Hello World!</h1>

# Override Styles in Subsequent CSS

(happening or coming after something else)

So the next logical question is, what can we do to override our pink-text class?

.blue-text is declared second, it overrides the attributes of .pink-text



<style>

body {

background-color: black;

font-family: Monospace;

color: green;

}

.pink-text {

color: pink;

}

.blue-text{

color:blue;

}

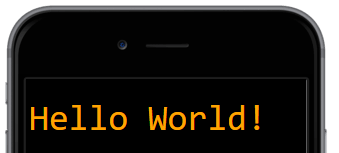
</style>

<h1 class="pink-text blue-text">Hello World!</h1>

# Override Class Declarations by Styling ID Attributes

We just proved that browsers read CSS from top to bottom. That means that, in the event of a conflict, the browser will use whichever CSS declaration came last.

But we're not done yet. There are other ways that you can override CSS. Do you remember id attributes?



#orange-text {

color: orange;

}

.pink-text {

color: pink;

}

.blue-text {

color: blue;

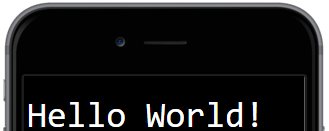
}

<h1 id="orange-text" class="pink-text blue-text">Hello World!</h1>

# Override Class Declarations with Inline Styles

So we've proven that id declarations override class declarations, regardless of where they are declared in your style element CSS.

There are other ways that you can override CSS. Do you remember inline styles?



<style>

body {

background-color: black;

font-family: Monospace;

color: green;

}

#orange-text {

color: orange;

}

.pink-text {

color: pink;

}

.blue-text {

color: blue;

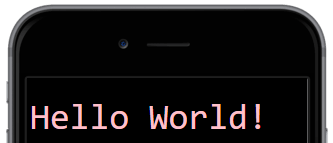
}

</style>

<h1 id="orange-text" class="pink-text blue-text" style="color: white">Hello World!</h1>

# Override All Other Styles by using Important

Yay! We just proved that in-line styles will override all the CSS declarations in your style element.



<style>

#orange-text {

color: orange;

}

.pink-text {

color: pink !important;

}

.blue-text {

color: blue;

}

</style>

<h1 id="orange-text" class="pink-text blue-text" style="color: white">Hello World!</h1>

# Use Hex Code for Specific Colors

<style>

body {

background-color: #000000; //=black

}

</style>

# Use Hex Code to Mix Colors

.red-text {

color: #FF0000;

}

.green-text {

color: #00FF00;

}

.dodger-blue-text {

color: #2998E4;

}

.orange-text {

color: #FFA500;

}

# Use Abbreviated Hex Code

made shorter



.red-text {

color:#F00;

}

.fuchsia-text {

color:#F0F;

}

.cyan-text {

color: #0FF;

}

.green-text {

color: #0F0;

}

# Use RGB values to Color Elements

<style>

body {

background-color: rgb(0, 0, 0); **//black**

}

</style>

# Use RGB to Mix Colors



.red-text {

color: rgb(255, 0, 0);

}

.orchid-text {

color: rgb(218, 112, 214);

}

.sienna-text {

color: rgb(160, 82, 45);

}

.blue-text {

color: rgb(0, 0, 255);

}