



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

Web Programming

Lecture 6

Cascading Style Sheets (CSS)

Part 3

By

Asst. Lect. Ali Al-khawaja

CSS Colors

- Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values
- In CSS, a color can be specified by using a predefined color name:



CSS Background Color

- You can set the background color for HTML elements
- `<h1 style="background-color:DodgerBlue;">Hello World</h1>`
`<p style="background-color:Tomato;">Lorem ipsum...</p>`

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS Text Color

- You can set the color of text
- ```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

# CSS Color Values

---

- In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:
- Same as color name "Tomato"

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

- Same as color name "Tomato", but 50% transparent:

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

# CSS Color Values Code :

---

- `<h1 style="background-color:rgb(255, 99, 71);">...</h1>`  
`<h1 style="background-color:#ff6347;">...</h1>`  
`<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>`  
  
`<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>`  
`<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>`

# RGB Value

---

- In CSS, a color can be specified as an RGB value, using this formula:
- **`rgb(red, green, blue)`**
- Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
- For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255) and the others are set to 0.
- To display black, set all color parameters to 0, like this: `rgb(0, 0, 0)`.
- To display white, set all color parameters to 255, like this: `rgb(255, 255, 255)`

`rgb(0, 255, 144)`

RED



GREEN



BLUE



`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(60, 179, 113)`

`rgb(238, 130, 238)`

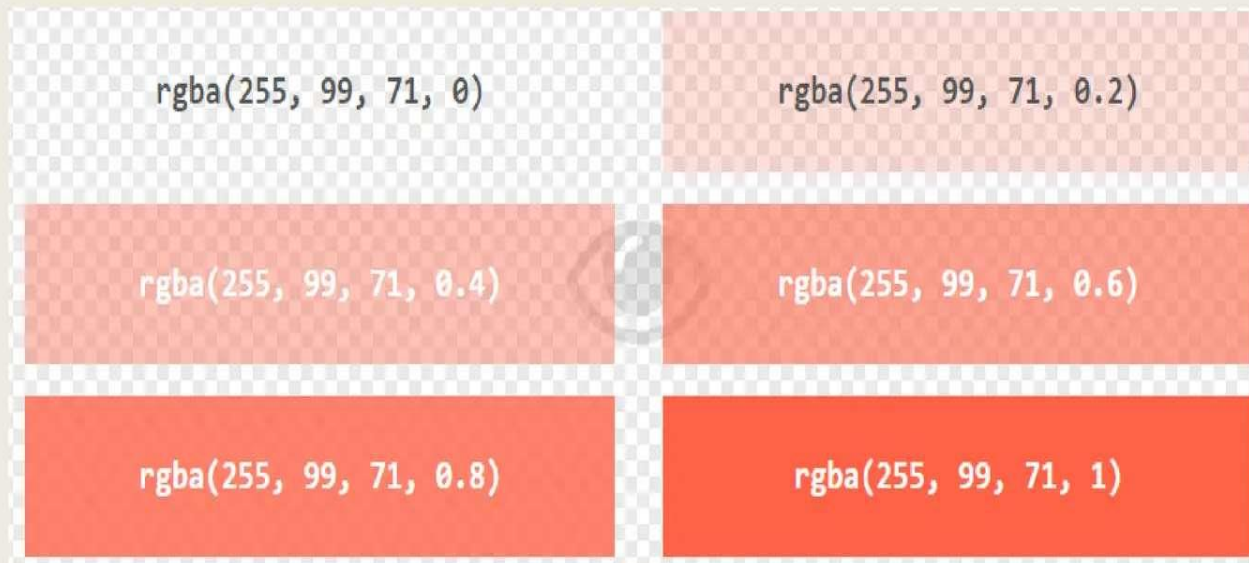
`rgb(255, 165, 0)`

`rgb(106, 90, 205)`

# RGBA Value

---

- RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.
- An RGBA color value is specified with:
- ***rgba(red, green, blue, alpha)***
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):
- Experiment by mixing the RGBA values below



# CSS HEX Colors

---

- A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.
- HEX Value
- In CSS, a color can be specified using a hexadecimal value in the form:
- **#rrggb**
- Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
- For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).
- To display black, set all values to 00, like this: #000000.
- To display white, set all values to ff, like this: #ffffff.
- Experiment by mixing the HEX values below

#006aff

RED GREEN BLUE

00 6a ff

#ff0000	#0000ff
#3cb371	#ee82ee
#ffa500	#6a5acd

# 3 Digit HEX Value

---

- Sometimes you will see a 3-digit hex code in the CSS source.
- The 3-digit hex code is a shorthand for some 6-digit hex codes.
- The 3-digit hex code has the following form:
- ***#rgb***
- Where r, g, and b represent the red, green, and blue components with values between 0 and f.
- The 3-digit hex code can only be used when both the values (RR, GG, and BB) are the same for each component. So, if we have #ff00cc, it can be written like this: #f0c.
- Here is an example:

# 3 Digit HEX Value

---

```
body {
 background-color: #fc9; /* same as #ffcc99 */
}
h1 {
 color: #f0f; /* same as #ff00ff */
}
p {
 color: #b58; /* same as #bb5588 */
}
```

CSS 3-digit Hex Code

This is a paragraph.

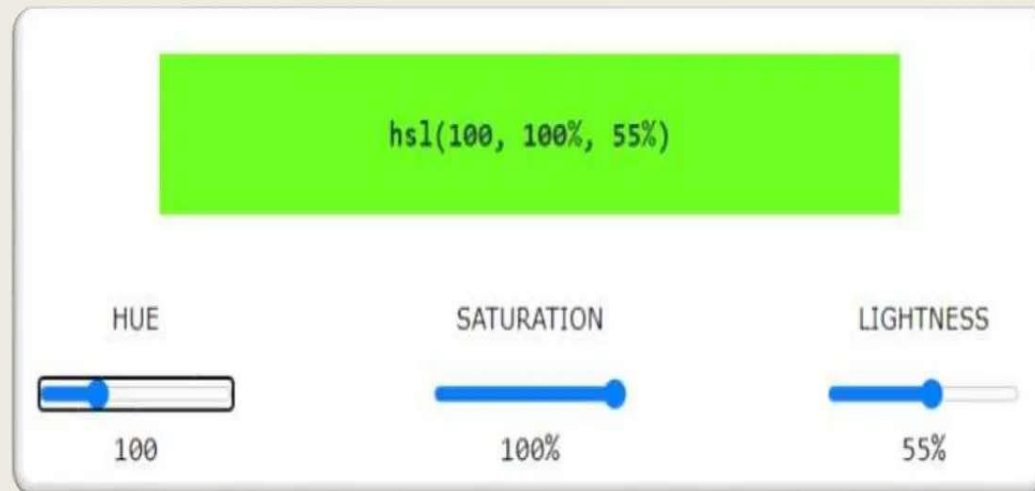
# CSS HSL Colors

---

- HSL stands for hue, saturation, and lightness.
- **`hsl(hue, saturation, lightness)`**
- Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
- Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.
- Lightness is also a percentage. 0% is black, 50% is neither light or dark, 100% is white

# CSS HSL Colors

---



<code>hsl(0, 100%, 50%)</code>	<code>hsl(240, 100%, 50%)</code>
<code>hsl(147, 50%, 47%)</code>	<code>hsl(300, 76%, 72%)</code>
<code>hsl(39, 100%, 50%)</code>	<code>hsl(248, 53%, 58%)</code>

# Saturation

---

- Saturation can be described as the intensity of a color.
- 100% is pure color, no shades of gray.
- 50% is 50% gray, but you can still see the color.
- 0% is completely gray; you can no longer see the color.

# Lightness

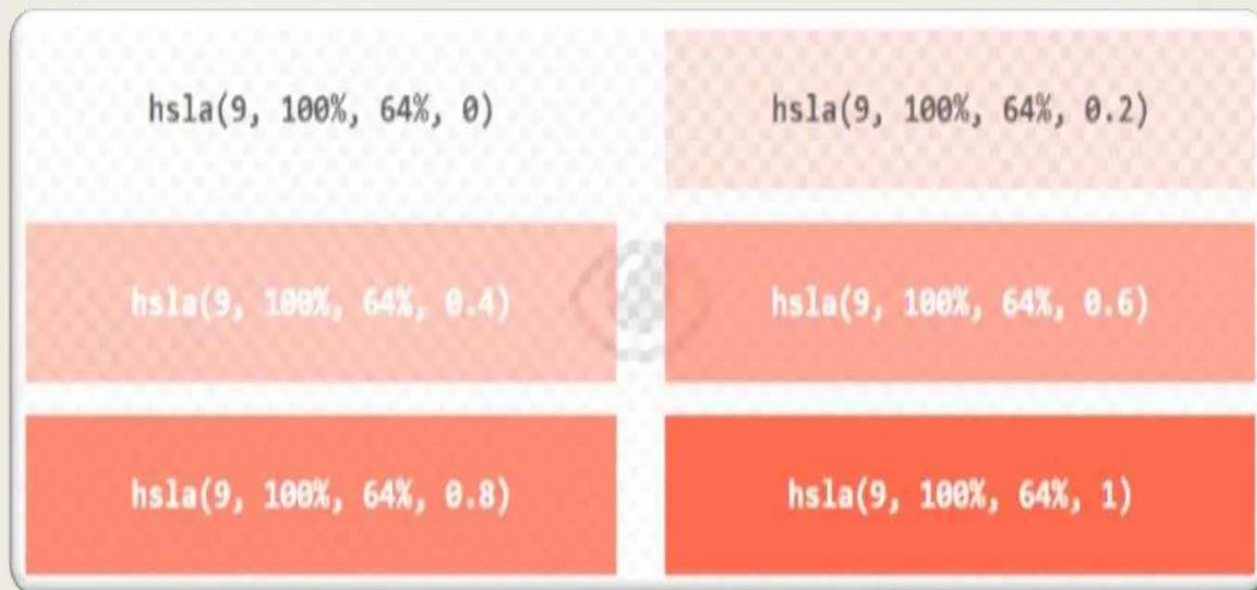
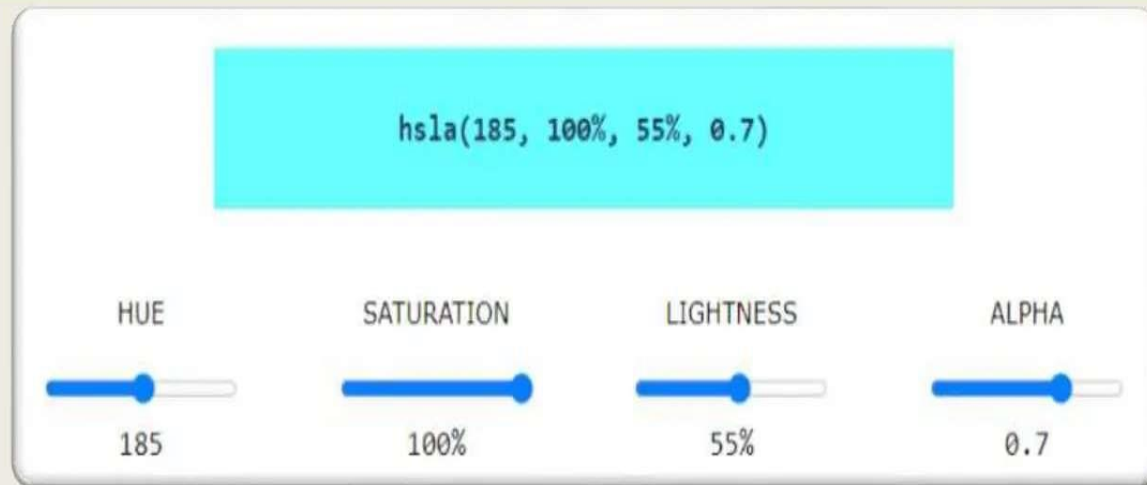
- The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) and 100% means full lightness (white).

# HSLA Value

---

- HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.
- An HSLA color value is specified with:
- ***hsla(hue, saturation, lightness, alpha)***
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

# HSLA Value



# CSS Links

---

- With CSS, links can be styled in many different ways.



## Styling Links

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

```
a {
 color: hotpink;
}
```

# Link Buttons

---

- The four links states are:
- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

# Link Buttons

---

- ```
/* unvisited link */
a:link {
    color: red;
}
/* visited link */
a:visited {
    color: green;
}
/* mouse over link */
a:hover {
    color: hotpink;
}
/* selected link */
a:active {
    color: blue;
}
```

This is a link

This is a link

This is a link

Link Buttons

- This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons

```
• a:link, a:visited {  
    background-color: #f44336;  
    color: white;  
    padding: 14px 25px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
}
```

```
a:hover, a:active {  
    background-color: red;  
}
```

Link Button

A link styled as a button:

This is a link

Link Button

A link styled as a button:

This is a link

CSS Padding

- Padding is used to create space around an element's content, inside of any defined borders.

This element has a padding of 70px.

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).
- Padding - Individual Sides
- CSS has properties for specifying the padding for each side of an element:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left
- All the padding properties can have the following values:
 - length - specifies a padding in px, pt, cm, etc.
 - % - specifies a padding in % of the width of the containing element
 - inherit - specifies that the padding should be inherited from the parent element

CSS Padding

- ```
div {
 padding-top: 50px;
 padding-right: 30px;
 padding-bottom: 50px;
 padding-left: 80px;
}
```

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

# Padding - Shorthand Property

---

- To shorten the code, it is possible to specify all the padding properties in one property.
- The padding property is a shorthand property for the following individual padding properties:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left
- So, here is how it works:

# Padding - Shorthand Property

---

- If the padding property has four values:
- padding: 25px 50px 75px 100px;
- top padding is 25px
- right padding is 50px
- bottom padding is 75px
- left padding is 100px
- ```
div {  
    padding: 25px 50px 75px 100px;  
}
```

This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.

Example

- If the padding property has two values:
- padding: 25px 50px;
- top and bottom paddings are 25px
- right and left paddings are 50px

```
div {  
  padding: 25px 50px;  
}
```

This div element has a top and bottom padding of 25px, and a right and left padding of 50px.

Think you

Any questions ??