

CSS and CSS3

CSS Selectors & Styling

Theory

Question 1) What is a css selector examples of element, class and id selectors.

Ans: A css selector is a pattern used to select and style HTML elements. It tells the browser which html elements the styles should apply to.

Examples:

1) Element selector: targets all elements of a specific

```
P {  
  Color: blue;  
}
```

This p is paragraph elements.

2) Class selector : targets elements with specific class

```
.highlight {  
  Background-color: yellow;  
}
```

This styles any element with class="highlight" with a yellow background.

3) Id Selector – Targets a unique element with a specific id attribute. Prefixed with a hash #.

```
#header {  
  Font-size: 24px;  
}
```

This styles the element with id="header" with a font size of 24px.

Question 2: explain the concept of css specificity. How do conflicts between multiple styles get resolved?

Ans: css specificity determines which css rule is applied when multiple rules could apply to the same element. it is based on the types of selector used in a rule.

Specificity hierarchy (from lowest to highest):

- 1) Universal selector *, element. It's based on the types of selectors used in a rule.
- 2) Class selectors (.class), attribute selector ([type="text"]), and pseudo-classes (:hover)
- 3) Id selectors (#id)
- 4) Inline style (e.g., <div style="color:red;">)
- 5) !important declarations (overrides all other rules)

```
<p id="intro" class="text"> hello world</p>
```

```
P {
```

```
Color:blue;
```

```
}
```

```
.text {
```

```
Color:green;
```

```
}
```

```
#intro {
```

```
Color:red;
```

```
}
```

The paragraph will be red, because the id selector has the highest specificity.

Question3: difference between internal, external, and inline css & their pros/cons.

- Inline:

```
<p style="color: blue;">hello</p>
```

Advantages: quick one-off styling and highest specificity for immediate effect.

Disadvantages: results in messy html, poor maintainability, no reusability, hard to override

- Internal css:

```
<head>
```

```
<style>
```

```
p { color: red; }  
</style>  
</head>
```

Advantages: Centralized styling for single-page use, no extra downloads.

Disadvantages: not reusable across pages; increases html size harder to manage for large projects

- External css:

```
<link rel="stylesheet" href="styles.css">
```

Advantages: clean separation of content and style; reusable across site; cached by browsers for better performance

Disadvantages: requires extra http request (may delay rendering); styles absent if file fails to load

Lab Assignment:

Task: style the contact form(created in the html forms lab) using external CSS.

The following should be implemented:

- Change the background color of the form
- Add padding and margins to form fields
- Style the submit button with a hover effect
- Use class selectors for styling common elements and id selectors for unique elements

Contact page:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8" />
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

```
  <title>Contact Form</title>
```

```
  <link rel="stylesheet" href="style2.css" />
```

```
</head>
```

```
<body>
```

```
  <div class="form-container">
```

```
<form id="contactForm">
```

```
  <h2>Contact Us</h2>
```

```
    <label for="name">Name:</label>
```

```
    <input type="text" id="name" class="input-field"
placeholder="Your Name" required>
```

```
    <label for="email">Email:</label>
```

```
    <input type="email" id="email" class="input-field"
placeholder="Your Email" required>
```

```
    <label for="message">Message:</label>
```

```
    <textarea id="message" class="input-field"
placeholder="Your Message" rows="5"
required></textarea>
```

```
    <input type="submit" id="submitBtn" value="Send
Message">
```

```
  </form>
```

```
</div>
```

```
</body>
```

</html>

Stylesheet page:

```
#contactForm {  
  background-color: #f2f8ff;  
  padding: 30px;  
  max-width: 400px;  
  margin: 50px auto;  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0,0,0,0.1);  
  font-family: Arial, sans-serif;  
}
```

```
.input-field {  
  width: 100%;  
  padding: 10px;  
  margin: 10px 0 20px 0;  
  border: 1px solid #ccc;  
  border-radius: 5px;
```

```
    box-sizing: border-box;
}
```

```
#submitBtn {
    background-color: #007BFF;
    color: white;
    padding: 12px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
    font-size: 16px;
    transition: background-color 0.3s ease;
}
```

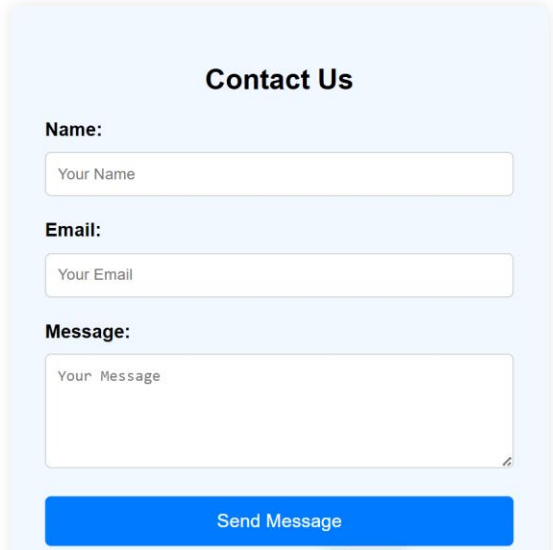
```
#submitBtn:hover {
    background-color: #0056b3;
}
```

```
#contactForm h2 {
```



```
text-align: center;
margin-bottom: 20px;
}
```

```
#contactForm label {
  font-weight: bold;
}
```



The image shows a contact form titled "Contact Us" on a light blue background. The form contains three input fields: "Name:" with placeholder text "Your Name", "Email:" with placeholder text "Your Email", and "Message:" with placeholder text "Your Message". Below these fields is a blue button labeled "Send Message".

CSS Box Model:

Theory Assignment:

Question1: explain the css box model and its components (content, padding, border, margin)

How does each affect the size of an element?

Answer:

The css box model is a fundamental concept in css that describes how the size of every html element is calculated and how it is rendered on the webpage.

It consists of 4 main components:

- Content: the actual text image or other data inside the element.
The width and height are applied directly to the content area.
- Padding: the space the content and the border.
adds space inside the element around the content
increases the total size of the element box-sizing: border-box is used .
- Border: the visible line around the padding and content.

Adds to the total width and height of the element

- Margin: the space outside the element, creating distance from other elements. Does not affect the size of the element itself but affects spacing on the page.

Total width = content width + left/right padding + left/right border + left/right margin

Total height = content height + top/bottom padding + top/bottom border + left/right margin

Question2 : what is the difference between border-box and content-box box-sizing in css?
Which is the default?

Answer:

content-box:

- The width and height apply only to the content.
- Padding and border are added outside the specified width and height
- Total size increases with padding/border.
Ex: box-sizing: content-box;

Border-box:

- The width and height include content + padding + border.
- Makes sizing more predictable and easier to manage layouts

Ex: box-sizing: content-box;

Or

```
*{  
Box-sizing: border-box;  
}
```

Box page

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <title>Profile Card</title>  
  <link rel="stylesheet" href="style.css">
```

```
</head>
<body>
  <div class="profile-card" align="center">
    
    <h2 class="profile-name">Devika Kollur</h2>
    <p class="profile-bio">Front-end Developer | UI/UX
Enthusiast | Learner</p>
    <button class="follow-btn">Follow</button>
  </div>
</body>
</html>
```

Stylesheet page:

```
/* Center the card on the page */
body {
  margin: 0;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #f0f0f5;
```

```
font-family: Arial, sans-serif;  
}
```

```
/* Profile Card Container */  
.profile-card {  
width: 300px;  
border: 2px solid #ccc;  
border-radius: 12px;  
padding: 20px;  
background-color: #fff;  
box-shadow: 0 8px 20px rgba(0,0,0,0.1);  
text-align: center;  
box-sizing: border-box; /* Using border-box for the  
card */  
}
```

```
/* Profile Image */  
.profile-img {  
width: 100px;  
height: 100px;
```

```
border-radius: 50%;  
border: 3px solid #4CAF50;  
padding: 5px;  
margin-bottom: 15px;  
box-sizing: content-box; /* Demonstrate content-box  
*/  
}
```

```
/* Name */  
.profile-name {  
  margin: 0;  
  font-size: 24px;  
  color: #333;  
  margin-bottom: 8px;  
}
```

```
/* Bio */  
.profile-bio {  
  font-size: 14px;  
  color: #666;
```

```
margin-bottom: 20px;  
padding: 0 10px;  
}
```

```
/* Follow Button */
```

```
.follow-btn {  
padding: 10px 25px;  
background-color: #4CAF50;  
color: white;  
border: none;  
border-radius: 6px;  
cursor: pointer;  
transition: background-color 0.3s;  
}
```

```
.follow-btn:hover {  
background-color: #45a049;  
}
```




Devika Kollur

Front-end Developer | UI/UX Enthusiast | Learner

Follow

CSS flexbox:

Theory question:

Question1: what is CSS flexbox, and how is it useful for layout design? Explain the terms flex-container and flex-item.

Answer: CSS flexbox (flexible box layout) is a CSS layout module designed to make it easier to arrange elements in a one-dimensional layout-either horizontally (row) or vertically (column)-with flexible sizing and spacing.

It is especially useful for building responsive web designs and aligning content both vertically and horizontally without using floats or complex CSS hacks.

Flex Container:

- The parent element that has `display:flex` or `display:inline-flex`.

- It enables flexbox behavior for its child elements.

Ex:

```
.container {  
  Display: flex;  
}
```

Flex items:

- The direct children of a flex container
- They are laid out according to the rules of the flexbox model.

Ex:

```
<div class="container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
</div>
```

Question 2: describe the properties justify-content, align-items, and flex-direction used in flexbox.

Answer:

Flex-direction:

Defines the main axis-the direction in which the flex items are placed inside the flex container.

Values:

Row(Default): left to right (horizontal)

Row-reverse: right to left

Column: top to bottom(vertical)

Column-reverse: bottom to top

Ex: .container {

Display: flex;

Flex-direction: row;

}

Justify-content:

Aligns flex items along the main axis(based on flex-direction).

Controls horizontal spacing when flex-direction: row.

Values:

- Flex-start (default): items aligned to the start
- Flex-end: items aligned to the end
- Center: items centered
- Space-between: equal space between items
- Space-around: equal space around items
- Space-evenly: equal space between and around items

Ex:

.container: {

```
Display: flex;  
Justify-content: center;  
}
```

Align-items:

Aligns flex items along the cross axis
(perpendicular to the main axis).

Controls vertical alignment when flex-
direction: row.

Value:

- Stretch (default): items stretch to fill container
- Flex-start: aligned to the top
- Flex-end: aligned to the bottom
- Center: vertically centered
- Baseline: aligned based on text baseline

Ex:

```
.container {  
  Display: flex;  
  Align-items: center;  
}
```

Lab Assignment • Task: Create a simple webpage layout using Flexbox. The layout should include: • A header. • A sidebar on the left. • A main content area in the center. • A

footer. Additional Requirements:

- Use Flexbox to position and align the elements.
- Apply different justify-content and align-items properties to observe their effects.
- Ensure the layout is responsive, adjusting for smaller screens.

```
<!DOCTYPE html>
<html>
<head>
  <title>Web Page</title>
  <style>
    * {
      box-sizing: border-box;
    }

    .row {
      display: flex;
    }

    .row1 {
      display: flex;
      flex-wrap: wrap;
    }

    [class*="col-"] {
```

```
width:100%;  
}
```

```
@media only screen and ( min-width: 600px)  
{
```

```
  .col-1 { width: 8.33%; }  
  .col-2 { width: 16.66%; }  
  .col-3 { width: 25%; }  
  .col-4 { width: 33.33%; }  
  .col-5 { width: 41.66%; }  
  .col-6 { width: 50%; }  
  .col-7 { width: 58.33%; }  
  .col-8 { width: 66.66%; }  
  .col-9 { width: 75%; }  
  .col-10 { width: 83.33%; }  
  .col-11 { width: 91.66%; }  
  .col-12 { width: 100%; }
```

```
}  
  
  .heading {  
    background-color: black;  
    font-size: 16px;  
    color: white;  
    padding: 10px;
```

```
}
```

```
.head-content {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.head-content1 {  
  font-size: 24px;  
  margin: 0;  
}
```

```
.search-bar select,  
.search-bar input[type="button"]  
.search-bar input[type="text"] {  
  padding: 10px;  
  margin: 10px;  
  border: none;  
  border-radius: 4px;  
  
}
```

```
.search-bar input[type="button"] {  
  padding: 10px;  
  margin: 10px;  
  cursor: pointer;
```

```
}
```

```
.links {
```

```
    background-color:darkcyan;  
    color: white;  
    font-size: 15px;  
    height: auto;  
    display: flex;  
    padding: 10px;  
    margin: 0px;  
    flex-wrap: wrap;  
    gap: 15px;  
    justify-content: space-between;
```

```
}
```

```
.links a {
```

```
    color: white;  
    margin-right: 60px;
```

```
}
```

```
.sidebar {
```



```
background-color: darkgray;
padding: 10px;
text-align: center;
}
```

```
.mainbar {
background-color: lightblue;
padding: 10px;
}
```

```
.photo {
padding: 10px;
border: solid 2px black;
margin: 10px;
border-width: 2px;
background-color:
lightgoldenrodyellow;
}
```

```
.photo img {
width: 100%;
}
```

```
h1, h2, h4, p {
margin: 5px 0;
}
```

```
a {
    text-decoration: none;
    color: black;
}
```

```
.footer {
    background-color:black;
    height: auto;
    color: white;
}
.footer a {
    color: white;
```

```

}
</style>
</head>
<body>
    <div class="row">
        <div class="heading col-12 head-content">
            <h1><b>Amazon.in</b></h1>
            <div class="search-bar head-content1"
style="display: inline;">
```

```
<select>
  <option>All Categories</option>
  <option>Alexa Skills</option>
  <option>Amazon Devices</option>
  <option>Amazon Fashion</option>
  <option>Amazon
Pharmacy</option>
  <option>Appliances</option>
  <option>Apps & Games</option>
  <option>Audible
Audiobooks</option>
  <option>baby toys</option>
  <option>Beauty</option>
  <option>Books</option>
  <option>CAr &
Motorbike</option>
  <option>Clothing &
Accessories</option>
  <option>Collectibles</option>
  <option>Computers &
Accessories</option>
  <option>Deals</option>

<option>Electronics</option>
  <option>Furniture</option>
```

<option>Garden &
Outdoors</option>

<option>Gift
Cards</option>

</select>

<input type="text" placeholder="Search
electronics...">

<input type="button" value="Search">

</div>

</div>

</div>

<div class="row">

<div class="col-12 links">

<div class="links">

<a

href="https://www.amazon.in/minitv?ref_=na
v_avod_desktop_topnav">MX Player

<a

href="https://www.amazon.in/b/3270202303
1?node=32702023031&ld=AZINSOANavDeskt
op_T3&ref_=nav_cs_sell_T3"> Sell

<a

href="https://www.amazon.in/gp/bestsellers/
?ref_=nav_cs_bestsellers">Bestsellers

</div>

</div>

</div>

<div class="row">

<div class="col-2 sidebar">

<h2>Categories</h2>

<dt>Computers &
Accessories</dt>

Accessories & Peripherals

Components

Desktop

<a

href="https://www.amazon.in/s?rh=n%3A976

[392031%2Cn%3A1375393031&dc&qid=1749188258&rnid=976392031&ref=sr_nr_n_4](https://www.amazon.in/s?rh=n%3A976392031%2Cn%3A1375393031&dc&qid=1749188258&rnid=976392031&ref=sr_nr_n_4)">External Devices & Data Storage

Laptop

Monitors

Networking Devices

Printers, inks & Accessories

Peripherals

88258&rnid=976392031&ref=sr_nr_n_9">Scanners

Tables

<h2>Item Condition</h2>

New

Renewed

<h2>Price</h2>

Under ₹1,000

₹1,000 - ₹5,000

₹5,000 - ₹10,000

₹10,000 - ₹20,000

Over ₹20,000

<h2>Deals & Discounts</h2>

All Discounts

Today's Deals

<h3>Made for Amazon Brands</h3>

[Made for Amazon](#)

<h2>Seller</h2>

[Cocoblu Retail](#)

[Clicktech Retail Private Ltd](#)

[TheGiftKart](#)

[Clicktech Business](#)

[Spigen India](#)

[DIZORO](#)

[KARWAN](#)

<h2>New Arrivals</h2>

[Last 30 days](#)

[Last 90 days](#)

<h2>Availability</h2>

[Include Out of Stock](#)

<h2>Discount</h2>

[10% Off or more](#)

[25% Off or more](#)

[35% Off or more](#)

[50% Off or more](#)

[60% Off or more](#)

[70% Off or more](#)

[Pay On Delivery](#)

[Eligible for Pay On Delivery](#)

</div>

<div class="col-8 mainbar">
 <h1 style="font-style: italic;">PC
Accessories & Gaming Accessories</h1>

<div class="row">
 <div class="col-4 photo">

<p>Corsair Katar Pro Ultra-Light
Optical USB Gaming Mouse with Backlit RGB
LED, 12400 DPI (Black)</p>

<h3>₹1,399.00</h3>

<h4 style="text-decoration:line-
through;">₹2,400.00</h4>

</div>

<div class="col-4 photo">

<p>Razer Huntsman Mini -
Mercury Edition - 60% Optical Gaming
Keyboard (Clicky Purple Switch)</p>

<h3>₹7,002.00</h3>

<h4 style="text-decoration:line-
through;">₹15,999.00</h4>

</div>

<div class="col-4 photo">

<p>Corsair Katar Pro Ultra-Light
Optical USB Gaming Mouse with Backlit RGB
LED, 12400 DPI (Black)</p>

<h3>₹1,399.00</h3>

<h4 style="text-decoration:line-
through;">₹2,400.00</h4>

</div>

<div class="col-4 photo">

<p>Razer Huntsman Mini -
Mercury Edition - 60% Optical Gaming
Keyboard (Clicky Purple Switch)</p>

<h3>₹7,002.00</h3>

<h4 style="text-decoration:line-
through;">₹15,999.00</h4>

</div>

</div>

<div class="row">

<div class="col-8 col-s-12 mainbar">

```

```

```
</div>
```

```
</div>
```

```
<h1>Trending Now...</h1>
```

```
<div class="row">
```

```
<div class="col-4 photo">
```

```
<a
```

```
href="https://www.amazon.in/dp/B0F2FTC64  
C/ref=QAHzEditorial_en_IN_1?pf_rd_r=3FWC  
RT251QTADPAGRY02&pf_rd_p=e579b4bf-  
efad-4087-b298-  
90536459625c&pf_rd_m=A1VBAL9TL5WCBF&  
pf_rd_s=merchandised-search-  
29&pf_rd_t=&pf_rd_i=1389401031&ie=UTF8  
&ref_=MiviC1">
```

```

```

```
</a>
```

```
</div>
```

```
<div class="col-4 photo">
```

<a
href="https://www.amazon.in/dp/B0F8BL3M
NT/ref=QAHZEditorial_en_IN_2?pf_rd_r=3FW
CRT251QTADPAGRY02&pf_rd_p=e579b4bf-
efad-4087-b298-
90536459625c&pf_rd_m=A1VBAL9TL5WCBF&
pf_rd_s=merchandised-search-
29&pf_rd_t=&pf_rd_i=1389401031">

</div>

<div class="col-4 photo">

<a
href="https://www.amazon.in/b/ref=QAHZEdit
orial_en_IN_3?pf_rd_r=3FWCRT251QTADPAG
RY02&pf_rd_p=e579b4bf-efad-4087-b298-
90536459625c&pf_rd_m=A1VBAL9TL5WCBF&
pf_rd_s=merchandised-search-
29&pf_rd_t=&pf_rd_i=1389401031&node=20
7504675031">


```
</a>
</div>
</div>
```

```
<div class="row">
  <div class="col-3 photo">
    
  </div>
  <div class="col-3 photo">
    
  </div>
  <div class="col-3 photo">
    
  </div>
  <div class="col-6 photo">
    
  </div>
```

</div>

<div class="row">

<div class="col-3 photo">

<a

href="https://www.amazon.in/Nothing-
Storage-Dimensity-Processor-
Display/dp/B0DC8F2X6T?pf_rd_p=e40307b2-
fbf4-4ff3-b85f-
cf2f9631bd1c&pf_rd_r=Y85ZS4N49RH912XD0
ABD&ref_=Smartphones-S3P_B0DC8F2X6T">

</div>

<div class="col-3 photo">

<a

href="https://www.amazon.in/realme-Pro-
Monet-Gold-
128GB/dp/B0DBM2XQYF?pf_rd_p=e40307b2-
fbf4-4ff3-b85f-

cf2f9631bd1c&pf_rd_r=Y85ZS4N49RH912XD0
ABD&ref_=Smartphones-S3P_B0DBM2XQYF">

</div>

<div class="col-3 photo">

<a

href="https://www.amazon.in/Nothing-
Phone-Black-256GB-
Storage/dp/B0DZNP6L5B?pf_rd_p=e40307b2-
fbf4-4ff3-b85f-
cf2f9631bd1c&pf_rd_r=Y85ZS4N49RH912XD0
ABD&ref_=Smartphones-S3P_B0DZNP6L5B">

</div>

<div class="col-3 photo">

<a

href="https://www.amazon.in/Nothing-
Storage-Snapdragon-Charging-

Expansion/dp/B0DZTMZL4X?pf_rd_p=e40307
b2-fbf4-4ff3-b85f-
cf2f9631bd1c&pf_rd_r=Y85ZS4N49RH912XD0
ABD&ref_=Smartphones-S3P_B0DZTMZL4X">

</div>

</div>

</div>

<div class="col-2 sidebar">

<h1> <a

href="https://www.amazon.in/electronics/b/?i
e=UTF8&node=976419031&ref_=topnav_stor
etab_top_elec_mega">Electronics</h1><
br>

<a

href="https://www.amazon.in/b/?_encoding=
UTF8&node=1389401031&ref_=sv_top_elec_
mega_1">Mobiles & Accessories

<a

href="https://www.amazon.in/b/?_encoding=

[UTF8&node=976392031&ref_=sv_top_elec_mega_2">Laptop & Accessories](https://www.amazon.in/b/?_encoding=UTF8&node=976392031&ref_=sv_top_elec_mega_2)

[TV & Home Entertainment](https://www.amazon.in/b/?_encoding=UTF8&node=1389375031&ref_=sv_top_elec_mega_3)

[Audio](https://www.amazon.in/b/?_encoding=UTF8&node=1389335031&ref_=sv_top_elec_mega_4)

[Cameras](https://www.amazon.in/b/?_encoding=UTF8&node=1388977031&ref_=sv_top_elec_mega_5)

[Computers Peripherals](https://www.amazon.in/b/?_encoding=UTF8&node=1375248031&ref_=sv_top_elec_mega_6)

[Smart Technology](https://www.amazon.in/b/?_encoding=UTF8&node=13773797031&ref_=sv_top_elec_mega_7)

[Musical Instruments](https://www.amazon.in/b/?_encoding=UTF8&node=3677697031&ref_=sv_top_elec_mega_8)

[Office & Stationery](https://www.amazon.in/b/?_encoding=UTF8&node=2454172031&ref_=sv_top_elec_mega_9)

Get to Know Us

[Careers](https://www.aboutamazon.in/?utm_source=gateway&utm_medium=footer)

[Press Releases](https://press.aboutamazon.in/?utm_source=gateway&utm_medium=footer)

```

        <a
href="https://www.amazon.science/">Amazo
n Science</a><br><br>
    </div>
</div>

```

```

    <div class="col-4 footer">
        <div class="foot-links">
            <h2><b>Connect with
Us</b></h2><br>
            <a
href="https://www.amazon.in/gp/redirect.ht
ml/ref=footer_fb?location=http://www.facebo
ok.com/AmazonIN&token=2075D5EAC7BB214
089728E2183FD391706D41E94&6">Facebook
</a><br><br>
            <a
href="https://www.amazon.in/gp/redirect.ht
ml/ref=footer_twitter?location=http://twitter.
com/AmazonIN&token=A309DFBFCB1E37A80
8FF531934855DC817F130B6&6">Twitter</a>
<br><br>
            <a
href="https://www.amazon.in/gp/redirect.ht
ml?location=https://www.instagram.com/ama

```

zondotin&token=264882C912E9D005CB1D9B
61F12E125D5DF9BFC7&source=standards">In
stagram

</div>

</div>

<div class="col-4 footer">

<div class="foot-links">

<h2>Make Money with
Us</h2>

<a

href="https://www.amazon.in/b/?node=2838
698031&Id=AZINSOANavDesktopFooter_C&re
f_=nav_footer_sell_C">Sell on
Amazon

<a

href="https://accelerator.amazon.in/?ref_=ma
p_1_b2b_GW_FT">Sell under Amazon
Accelerator

<a

href="https://brandservices.amazon.in/?ref=A
OINABRLGNRFOOT&Id=AOINABRLGNRFOOT">
Protect and Build your Brand

https://sell.amazon.in/grow-your-business/amazon-global-selling.html?Id=AZIN_Footer_V1&ref=AZIN_Footer_V1> Amazon Global Selling

https://supply.amazon.com/?ref_=footer_sta&lang=en-IN>Supply to Amazon

https://affiliate-program.amazon.in/?utm_campaign=assocshowcase&utm_medium=footer&utm_source=GW&ref_=footer_assoc>Become an Affiliate

https://services.amazon.in/services/fulfilment-by-amazon/benefits.html/ref=az_footer_fba?Id=AWRGINFBAfooter>Fulfilment by Amazon

<https://advertising.amazon.in/?ref=Amz.in>>Advertise your Products

<https://www.amazonpay.in/merchant>>Amazon pay on Merchants


```
    </div>
  </div>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
<section style="background-color: black;
padding: 60px 40px;">
```

```
  <div style="display: flex; flex-direction:
column; gap: 30px;">
```

```
    <!-- "NEW" Tag with Horizontal Line -->
```

```
    <div style="display: flex; align-items:
center;">
```

```
      <div style="background-color: red; color:
black; font-weight: bold; font-size: 28px;
padding: 8px 18px;">NEW</div>
```

```
      <div style="flex-grow: 1; height: 4px;
background-color: red;"></div>
```

```
    </div>
```

```
    <!-- "ARRIVALS" and Button -->
```



```
<h1 style="color: white; font-size: 100px;
font-weight: bold; margin: 0;">ARRIVALS</h1>
<a href="#" style="background-color: white;
color: black; font-weight: bold; padding: 10px
20px; text-decoration: none; width: fit-
content;">EXPLORE NOW</a>
```

```
<!-- Product Image Carousel -->
<div style="display: flex; gap: 40px;
overflow-x: auto; padding-top: 30px;">
  
  
  
</div>
```

</div>
</section>

<section>
 <div>

</section>

Amazon.in

All Categories

Search electronics

Search

MX PlayerSellBestsellersMobilesToday's DealsCustomer ServiceFashionNew ReleasesAmazon PayLogin/Sign Up

Categories

Computers & Accessories

Accessories & Peripherals

Components

Desktop

External Devices & Data Storage

Laptop

Monitors

Networking Devices

Printers, inks & Accessories

Scanners

Tablets

Item Condition

New

Renewed

Price

Under ₹1,000

₹1,000 - ₹5,000

₹5,000 - ₹10,000

₹10,000 - ₹20,000

₹20,000 - ₹50,000


Over ₹50,000


Deals & Discounts

PC Accessories & Gaming Accessories

ELECTRONIC

AND Accessories






Corsair Katar Pro Ultra-Light Optical USB Gaming Mouse with Backlit RGB LED, 12400 DPI (Black)

₹1,399.00


₹2,400.00



Razer Huntsman Mini - Mercury Edition - 60% Optical Gaming Keyboard (Clicky Purple Switch)

₹7,002.00


₹15,999.00



Corsair Katar Pro Ultra-Light Optical USB Gaming Mouse with Backlit RGB LED, 12400 DPI (Black)

₹1,399.00

₹2,400.00



Razer Huntsman Mini - Mercury Edition - 60% Optical Gaming Keyboard (Clicky Purple Switch)

₹7,002.00

₹15,999.00

Electronics

Mobiles & Accessories

Laptop & Accessories

TV & Home Entertainment

Audio

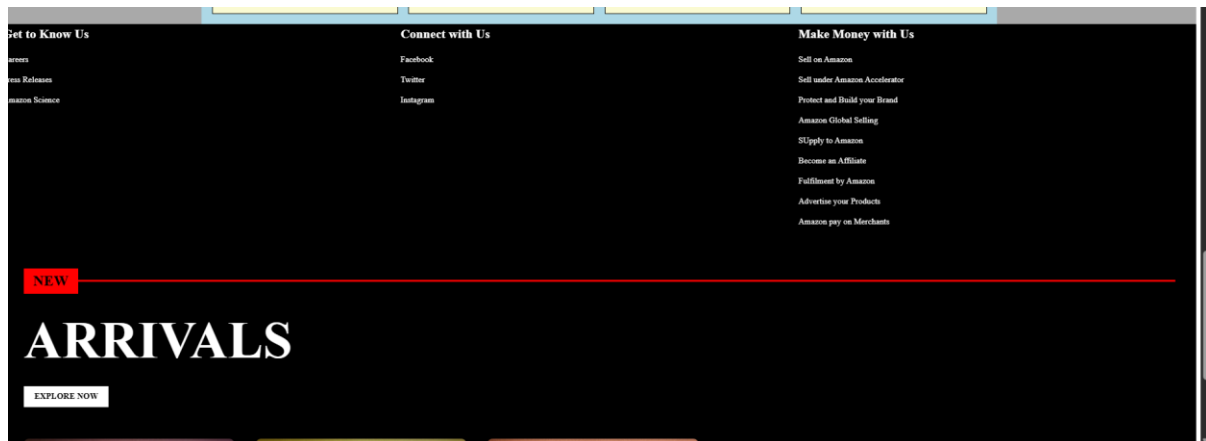
Cameras

Computers Peripherals

Smart Technology

Musical Instruments

Office & Stationery



- CSS Grid

Theory Assignment

- Question 1: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

Answer: **CSS Grid** is a powerful layout system in CSS that allows you to create **two-dimensional layouts** (both rows and columns). It divides a webpage into major regions or defines the relationship in terms of size, position, and layer between parts of a control built from HTML primitives.

Flexbox vs grid

Flexbox: one dimensional (row or column), aligning items in a row or column (e.g., navbars, menus), good

for distributing space and alignment between items,
flexible for items that flow

Grid: two-dimensional (row and columns), creating full-page layout or complex grids (e.g., dashboards, galleries), excellent for precise placement of items in both directions, great for defining areas and adjusting at breakpoints

When use grid over flexbox: Use **Grid** when you need a structured **2D layout** like a webpage with a header, sidebar, main content, and footer.

Use **Flexbox** when arranging elements **in a single line** (horizontal or vertical) such as navigation bars or buttons.

- Question 2: Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.

Answer:

grid-template-columns

- Defines the number and size of columns in the grid.
- Example:

- .container {
- display: grid;
- grid-template-columns: 200px 1fr 2fr;
- /* First column fixed, second flexible, third double size */
- }

grid-template-rows

- Defines the number and size of rows in the grid.
- Example:
- .container {
- display: grid;
- grid-template-rows: 100px auto 50px;
- /* Header fixed 100px, main auto, footer 50px */
- }

grid-gap (or gap in modern CSS)

- Defines the spacing between rows and columns.
- Example:
- .container {

- `display: grid;`
- `grid-template-columns: repeat(3, 1fr); /* 3 equal columns */`
- `grid-template-rows: 100px 200px;`
- `gap: 20px; /* Adds 20px space between rows & columns */`

}

Lab Assignment

- Task: Create a 3x3 grid of product cards using CSS Grid. Each card should contain:
 - A product image.
 - A product title.
 - A price. Additional Requirements:
 - Use `grid-template-columns` to create the grid layout.
 - Use `grid-gap` to add spacing between the grid items.
 - Apply hover effects to each card for better interactivity.

<!DOCTYPE html>

<html lang="en">

```
<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-
width, initial-scale=1.0">

  <title>Product Grid</title>

  <link rel="stylesheet" href="styles.css">
</head>

<body>

  <h1 class="title">Product Grid</h1>


  <div class="grid-container">

    <div class="card">

      <h2>Product 1</h2>

      <p>$25.00</p>

    </div>

    <div class="card">
```

```

```

```
<h2>Product 2</h2>
```

```
<p>$35.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 3</h2>
```

```
<p>$45.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 4</h2>
```

```
<p>$30.00</p>
```

```
</div>
```

```
<div class="card">
```



```

```

```
<h2>Product 5</h2>
```

```
<p>$40.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 6</h2>
```

```
<p>$55.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 7</h2>
```

```
<p>$60.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 8</h2>
```

```
<p>$75.00</p>
```

```
</div>
```

```
<div class="card">
```

```

```

```
<h2>Product 9</h2>
```

```
<p>$90.00</p>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Css file:

```
body {  
    font-family: Arial, sans-serif;
```

```
background: #f5f5f5;  
margin: 0;  
padding: 20px;  
}
```

```
.title {  
  text-align: center;  
  margin-bottom: 20px;  
}
```

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 20px;  
  padding: 10px;  
}
```

```
.card {  
  background: white;  
  border-radius: 8px;
```

```
padding: 15px;
text-align: center;
box-shadow: 0px 2px 8px rgba(0,0,0,0.1);
transition: transform 0.3s, box-shadow 0.3s;
}
```

```
.card img {
  max-width: 100%;
  border-radius: 5px;
}
```

```
.card h2 {
  font-size: 18px;
  margin: 10px 0;
}
```

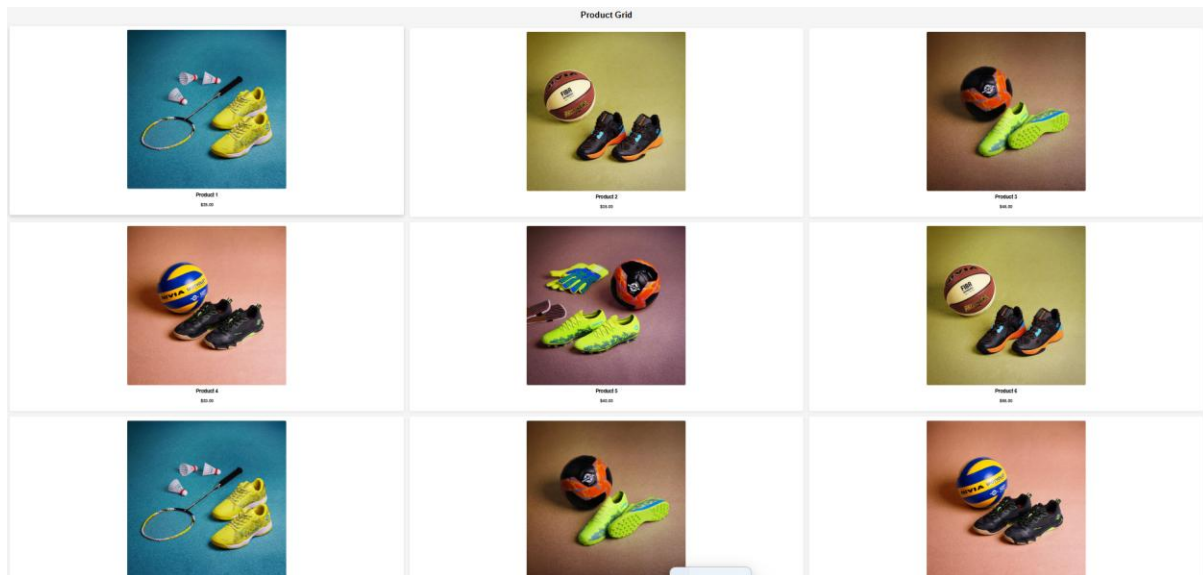
```
.card p {
  font-size: 16px;
  color: #333;
  font-weight: bold;
```

```
}
```

```
.card:hover {  
  transform: translateY(-8px);  
  box-shadow: 0px 6px 15px rgba(0,0,0,0.2);  
}
```

```
/* Responsive design */  
@media (max-width: 900px) {  
  .grid-container {  
    grid-template-columns: repeat(2, 1fr);  
  }  
}
```

```
@media (max-width: 600px) {  
  .grid-container {  
    grid-template-columns: 1fr;  
  }  
}
```



- Responsive Web Design with Media Queries

Theory Assignment

- Question 1: What are media queries in CSS, and why are they important for responsive design?
 - Answer: **Media queries** in CSS are rules that allow you to apply styles only when certain conditions about the device's characteristics (such as screen width, height, orientation, or resolution) are met.
 - They are written using the @media rule.

Importance in responsive design:

1. **Adaptability:** They allow websites to adjust their layout and appearance for different devices (desktop, tablet, mobile).
2. **Improved User Experience:** Users can view content comfortably regardless of their screen size.
3. **Mobile-first design:** Developers can design websites that scale naturally from small to large screens.
4. **Accessibility:** Ensures that designs remain functional and readable across various devices.

- Question 2: Write a basic media query that adjusts the font size of a webpage for screens smaller than 600px

Answer:

```
body {  
    font-size: 18px; /* Default font size for larger screens  
*/  
}
```

```
@media screen and (max-width: 600px) {
```

```
body {  
    font-size: 14px; /* Smaller font size for mobile  
screens */  
}  
}
```

By default, the font size is 18px for desktops or larger screens.

When the screen width is **600px or smaller**, the font size changes to 14px.

Lab Assignment

- Task: Build a responsive webpage that includes:
- A navigation bar.
- A content section with two columns.
- A footer. Additional Requirements:
- Use media queries to make the webpage responsive for mobile devices.
- On smaller screens (below 768px), stack the columns vertically.
- Adjust the font sizes and padding to improve readability on mobile.


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Responsive Webpage</title>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
  }

  /* Navigation Bar */
```

```
nav {  
    background: #333;  
    color: #fff;  
    padding: 15px;  
    text-align: center;  
}
```

```
nav a {  
    color: #fff;  
    text-decoration: none;  
    margin: 0 15px;  
    font-size: 18px;  
}
```

```
nav a:hover {  
    text-decoration: underline;  
}
```

```
/* Content Section */  
.container {
```

```
display: grid;
grid-template-columns: 2fr 1fr;
gap: 20px;
padding: 20px;
}
```

```
.content, .sidebar {
padding: 20px;
background: #f4f4f4;
border-radius: 5px;
}
```

```
/* Footer */
footer {
background: #333;
color: #fff;
text-align: center;
padding: 15px;
margin-top: 20px;
}
```

```
@media (max-width: 768px) {  
  .container {  
    grid-template-columns: 1fr;  
  }  
}
```

```
nav {  
  font-size: 16px;  
  padding: 10px;  
}
```

```
nav a {  
  display: block;  
  margin: 8px 0;  
}
```

```
.content, .sidebar {  
  font-size: 16px;  
  padding: 15px;  
}
```

```
footer {  
  font-size: 14px;  
  padding: 10px;  
}  
}
```

```
@media (max-width: 480px) {  
  nav {  
    font-size: 14px;  
  }  
}
```

```
.content, .sidebar {  
  font-size: 14px;  
  padding: 10px;  
}
```

```
footer {  
  font-size: 12px;  
}
```

```
}  
</style>  
</head>  
<body>  
  
<nav>  
  <a href="#">Home</a>  
  <a href="#">About</a>  
  <a href="#">Services</a>  
  <a href="#">Contact</a>  
</nav>  
  
<div class="container">  
  <div class="content">  
    <h2>Main Content</h2>  
    <p>  
      This is the main content area. On large screens, it  
      appears beside the sidebar.  
  
      On smaller screens, the layout stacks vertically for  
      better readability.  
    </p>
```

</div>

<div class="sidebar">

<h2>Sidebar</h2>

<p>

This is the sidebar section with related information, links, or ads.

It will appear below the main content on smaller screens.

</p>

</div>

</div>

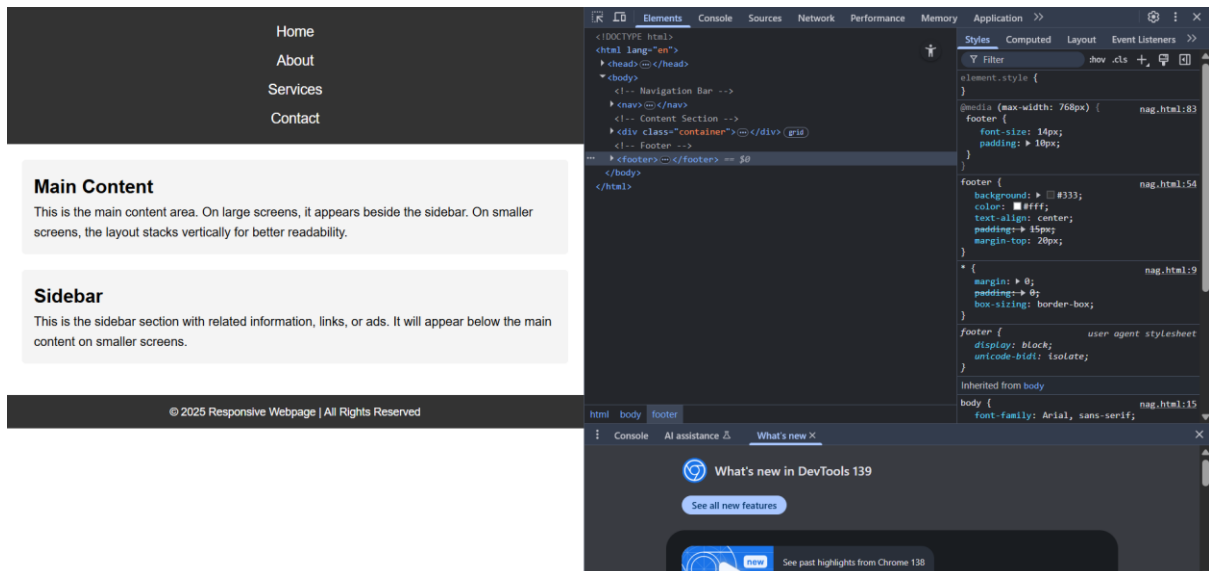
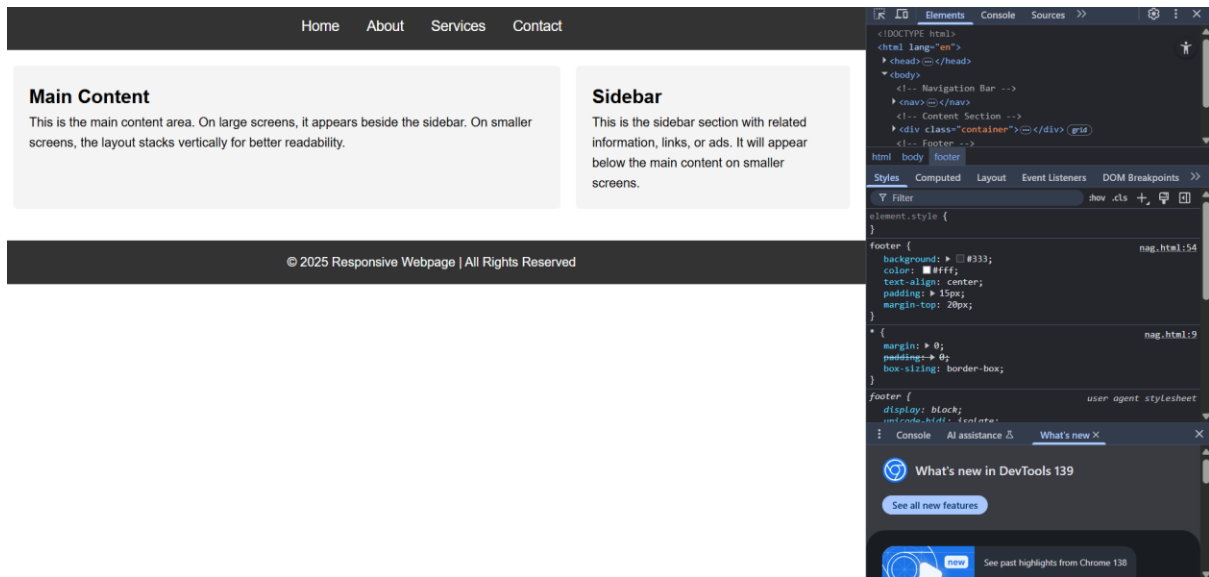
<footer>

<p>© 2025 Responsive Webpage | All Rights Reserved</p>

</footer>

</body>

</html>



• Typography and Web Fonts

Theory Assignment

- Question 1: Explain the difference between web-safe fonts and custom web fonts. Why might you use a web-safe font over a custom font?

Answer:

- **Web-safe fonts:**

Web-safe fonts are a set of fonts that are pre-installed on most operating systems and devices. Examples include **Arial, Times New Roman, Verdana, Courier New, and Georgia**. Since these fonts are already available on users' devices, they don't need to be downloaded from the internet, making them **fast and reliable**.

- **Custom web fonts:**

Custom web fonts (like those from **Google Fonts, Adobe Fonts, or custom @font-face imports**) are fonts that are not pre-installed on devices. Instead, they are downloaded from a server when the webpage loads. They provide **greater design flexibility** and allow developers to use unique typography.

- **Why use web-safe fonts over custom fonts?**

- Faster page loading speed (no extra font files to download).
- More consistent rendering across different browsers and devices.
- Useful in cases where performance is critical (e.g., low-bandwidth conditions).
- Avoids dependency on external font providers.

In short, **web-safe fonts** prioritize performance and **compatibility**, while **custom fonts** prioritize design and branding.

• Question 2: What is the font-family property in CSS?
How do you apply a custom Google Font to a webpage?

Answer:

- The **font-family property** in CSS specifies the font style to be applied to text on a webpage. It can include multiple fonts as a **fallback system**, in case the preferred font is not available on the user's device.

Example:

```
p {  
  font-family: "Arial", "Helvetica", sans-serif;  
}
```

Here:

- The browser tries **Arial** first.
- If Arial is not available, it tries **Helvetica**.
- If neither is available, it defaults to a **generic sans-serif** font.

Lab Assignment

- Task: Create a blog post layout with the following:
- A title, subtitle, and body content.
- Use at least two different fonts (one for headings, one for body content).
- Style the text to be responsive and easy to read.

Additional Requirements:

- Use a custom font from Google Fonts.
- Adjust line-height, font-size, and spacing for improved readability.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Blog Post Layout</title>
```

```
  <link
```

```
    href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&family=Merriweather:wght@400;700&display=swap" rel="stylesheet">
```

```
<style>
```

```
body {  
    font-family: 'Roboto', sans-serif; /* Body font */  
    margin: 20px;  
    background: #f9f9f9;  
    color: #333;  
    line-height: 1.6;  
}
```

```
.blog-post {  
    max-width: 800px;  
    margin: auto;  
    background: #fff;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0px 2px 8px rgba(0, 0, 0, 0.1);  
}
```

```
h1, h2 {
```

```
font-family: 'Merriweather', serif;  
margin-bottom: 15px;  
}
```

```
h1 {  
  font-size: 2.5rem;  
  color: #222;  
}
```

```
h2 {  
  font-size: 1.5rem;  
  color: #555;  
  margin-bottom: 20px;  
}
```

```
p {  
  font-size: 1.1rem;  
  margin-bottom: 15px;  
}
```

```
@media (max-width: 768px) {  
  h1 {  
    font-size: 2rem;  
  }  
  h2 {  
    font-size: 1.3rem;  
  }  
  p {  
    font-size: 1rem;  
  }  
}
```

```
@media (max-width: 480px) {  
  h1 {  
    font-size: 1.6rem;  
  }  
  h2 {  
    font-size: 1.1rem;  
  }  
  p {
```

```
        font-size: 0.95rem;
        line-height: 1.5;
    }
}
</style>
</head>
<body>

<div class="blog-post">
    <h1>The Future of Web Design</h1>
    <h2>Responsive layouts and modern
typography</h2>
    <p>
        Web design has evolved significantly over the years,
        shifting from static layouts to fully
        responsive and interactive user experiences. With
        the rise of mobile-first design, ensuring
        that websites adapt seamlessly across devices is
        more important than ever.
    </p>
    <p>
```

Typography plays a critical role in readability and aesthetics. Using a combination of

web-safe fonts and custom Google Fonts allows designers to balance performance and creativity.

Good typography improves user engagement and ensures content is both accessible and beautiful.

</p>

<p>

As web technologies continue to advance, we can expect even more innovative ways to deliver

content that is fast, accessible, and visually engaging.

</p>

</div>

</body>

</html>

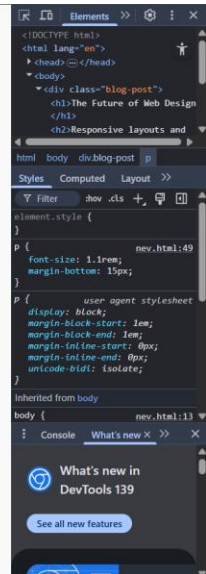
The Future of Web Design

Responsive layouts and modern typography

Web design has evolved significantly over the years, shifting from static layouts to fully responsive and interactive user experiences. With the rise of mobile-first design, ensuring that websites adapt seamlessly across devices is more important than ever.

Typography plays a critical role in readability and aesthetics. Using a combination of web-safe fonts and custom Google Fonts allows designers to balance performance and creativity. Good typography improves user engagement and ensures content is both accessible and beautiful.

As web technologies continue to advance, we can expect even more innovative ways to deliver content that is fast, accessible, and visually engaging.



The Future of Web Design

Responsive layouts and modern typography

Web design has evolved significantly over the years, shifting from static layouts to fully responsive and interactive user experiences. With the rise of mobile-first design, ensuring that websites adapt seamlessly across devices is more important than ever.

Typography plays a critical role in readability and aesthetics. Using a combination of web-safe fonts and custom Google Fonts allows designers to balance performance and creativity. Good typography improves user engagement and ensures content is both accessible and beautiful.

As web technologies continue to advance, we can expect even more innovative ways to deliver content that is fast, accessible, and visually engaging.

