

# **Lecture 2.1**

## **Front End Programming**

**Course Instructor**  
**Engr. Madeha Mushtaq**

# Front end Programming

A gray rounded square with a black border containing the text 'HTML' in white.

HTML

markup language  
**content**

A dark blue rounded square with a black border containing the text 'CSS' in white.

CSS

style sheet language  
**presentation**

A teal rounded square with a black border containing the text 'JavaScript' in white.

JavaScript

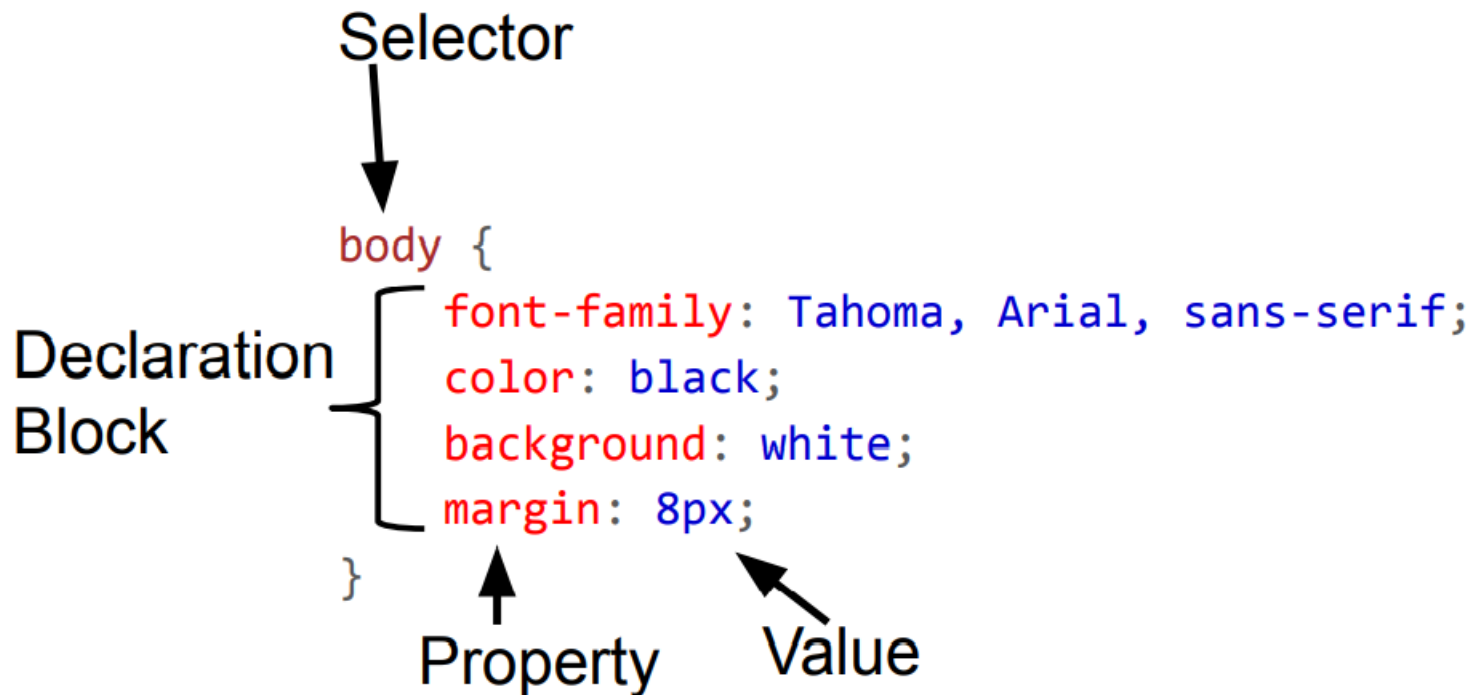
programming language  
**behavior**

# Front End Programming

- HTML: Content (what to display) is in HTML files.
- CSS: Formatting information (how to display it) is in separate style sheets (.css files).
- Style sheets were added to address:
  - Specify style to use rather than browser default.
  - Not have to code styling on every element.
  - Define style once, use in many places.

# Cascading Style Sheets (CSS)

- Style sheet contain one or more CSS Rules.



# Writing Style Sheets

- In-line styles
- Embedded/internal styles
- External style sheet

# Cascading Style Sheets (CSS)

## Adding Styles to HTML

Separate style sheet (best way)

```
<head>
  <link rel="stylesheet" type="text/css" href="myStyles.css" />
  <style type="text/css">
    body {
      font-family: Tahoma, Arial, sans-serif;
    }
  </style>
</head>
<body>
  <div style="padding:2px; ... ">
</body>
```

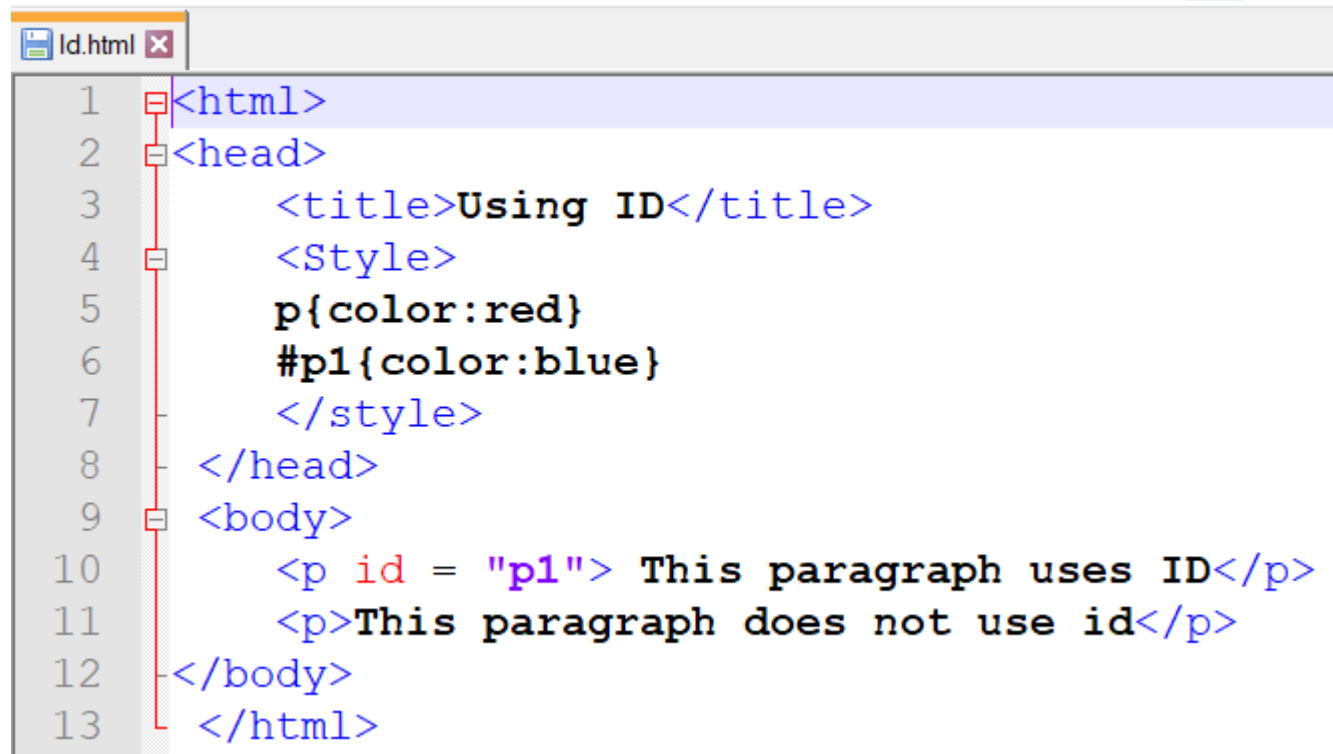
Page-specific styles

Element-specific styles

# Using ID in CSS

- Use an id to distinguish something, like a paragraph, from the others in a document.
- Create a style Id:  
`#id_name {style attributes and values}`
- Use a style Id:  
`<tag ID=id_name>`

# Using ID in CSS



```
1 <html>
2 <head>
3   <title>Using ID</title>
4   <style>
5     p{color:red}
6     #p1{color:blue}
7   </style>
8 </head>
9 <body>
10   <p id = "p1"> This paragraph uses ID</p>
11   <p>This paragraph does not use id</p>
12 </body>
13 </html>
```



# Using Classes

- HTML and XHTML require each id be unique—therefore an id value can only be used once in a document
- **ID** used to identify one element, whereas a **class** can be used to identify more than one.
- You can mark a group of elements with a common identifier using the class attribute.

# Using Classes in CSS

```
class.html x
1 <html>
2 <head>
3   <title>Using Classes</title>
4   <style>
5     p{color:red}
6     p.p1{color:blue}
7     .p2{color:green}
8   </style>
9 </head>
10 <body>
11   <p>This paragraph does not use class</p>
12   <p class = "p1">This paragraph uses p1 class</p>
13   <p class = "p2">This paragraph uses p2 class</p>
14   <h1 class = "p2">This Heading uses p2 class</h1>
15 </body>
16 </html>
```

# Cascading Style Sheets (CSS)

CSS Selector	CSS	HTML
Tag name	<pre>h1 {   color: red; }</pre>	<pre>&lt;h1&gt;Today's Specials&lt;/h1&gt;</pre>
Class attribute	<pre>.large {   font-size: 16pt; }</pre>	<pre>&lt;p class="large"&gt;...</pre>
Tag and Class	<pre>p.large {...}</pre>	<pre>&lt;p class="large"&gt;...</pre>
Element id	<pre>#p20 {   font-weight: bold; }</pre>	<pre>&lt;p id="p20"&gt;...</pre>

# Difference between classes and Ids

- You can't have more than one tag with the same ID value.
- You can apply the same Class value to multiple document tags
- Classes or Id?
  - use ID's for any elements that are simply used once on a page.

# Consider a web application to develop

- CONNECTIONS: A smaller version of social web
- Users can register themselves
- After registration they can send a text message to anybody who is registered with the website
- User's can view the received messages.

# The process

- Determine the requirements of the site
- Group the required information
- Make a site map
- Identify key elements for each page.
- Decide about the arrangement of information on each page
- Translate the design into code.

# Determining requirements

- Things You Want the Site to Do
- Determining the requirement of the owner of the site which are not already listed
  - User must register themselves in-order to send and receive messages.

# Grouping and categorization

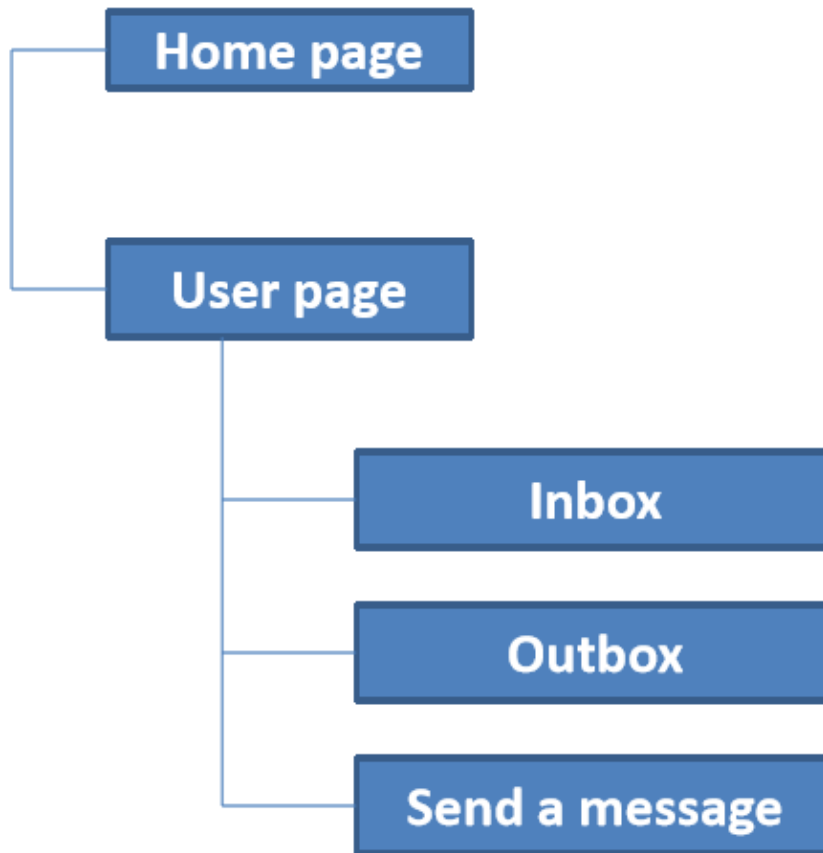
- Group related information/sections
  - Home page: login form, registration form
  - User page: user's information, recent messages
  - Inbox: Received messages
  - Outbox: Sent messages
  - Send a message: Send message form



# Make a site map

- Site map shows how many pages you need and how they relate
- Usually look like either a family tree or folder list.

# Make a site map



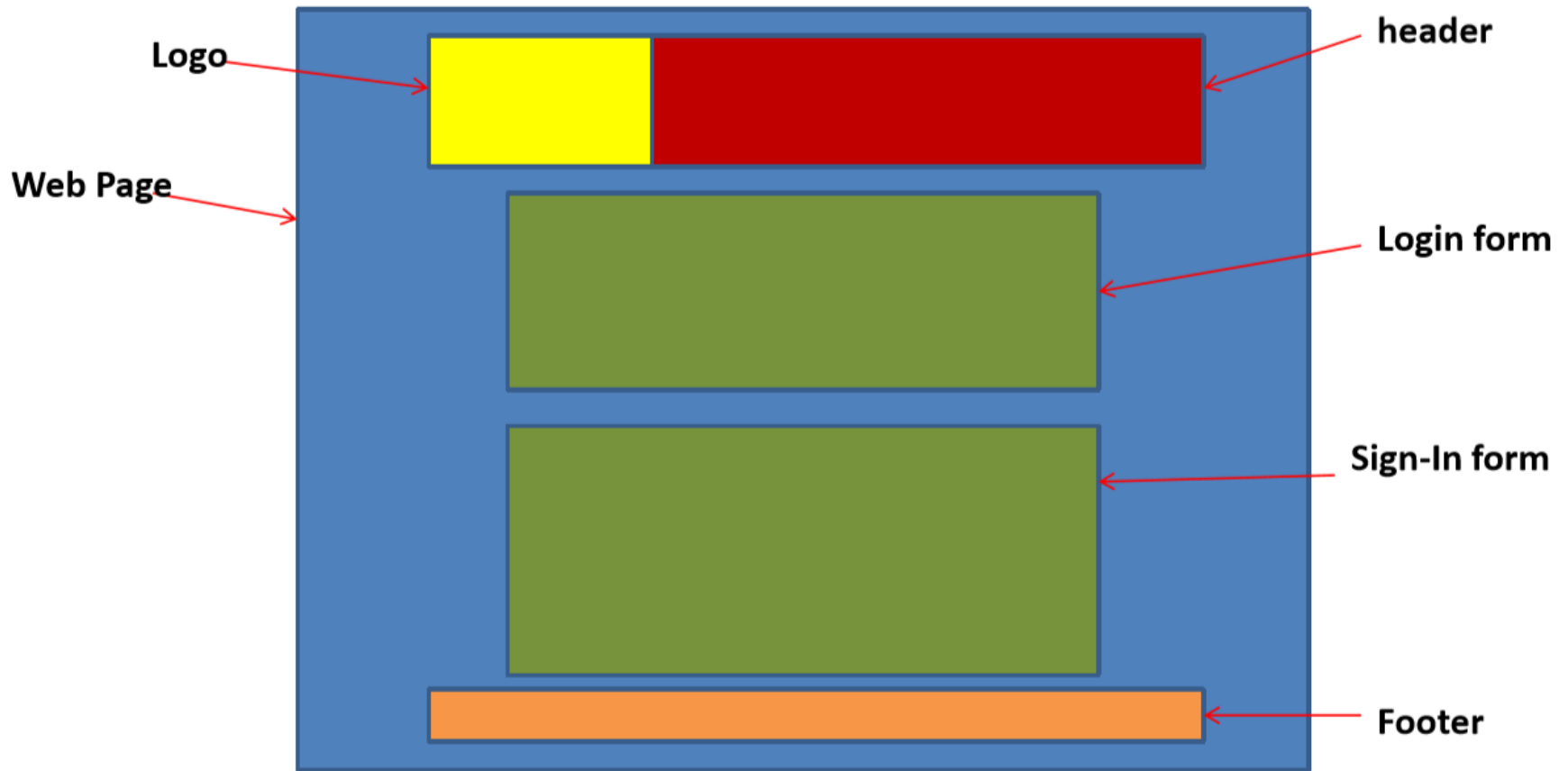
# Key element for each page

- Common sections:
- (header, navigation, footer etc).
- Page specific sections:
- Home page: login form, registration form
- User page: user's information, recent messages.

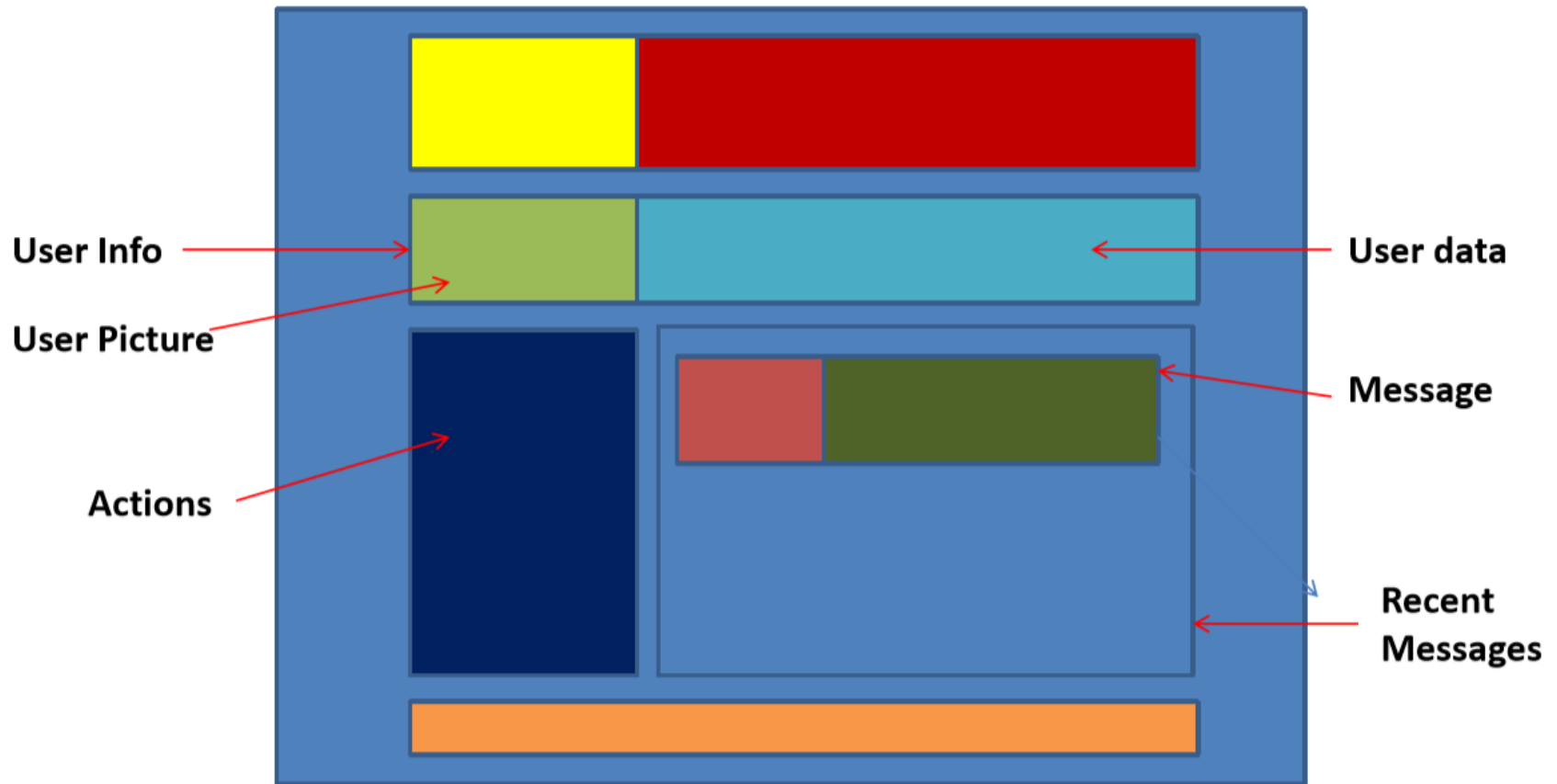
# Key element for each page

- Inbox page: received messages
- Outbox page: messages sent by the user
- Send message page: form to send a message.

# Arranging elements on pages



# Arranging elements on pages



# Translating design into code

- Liquid vs Fixed design
- The Div tag
- Coding the design

# Liquid vs Fixed design

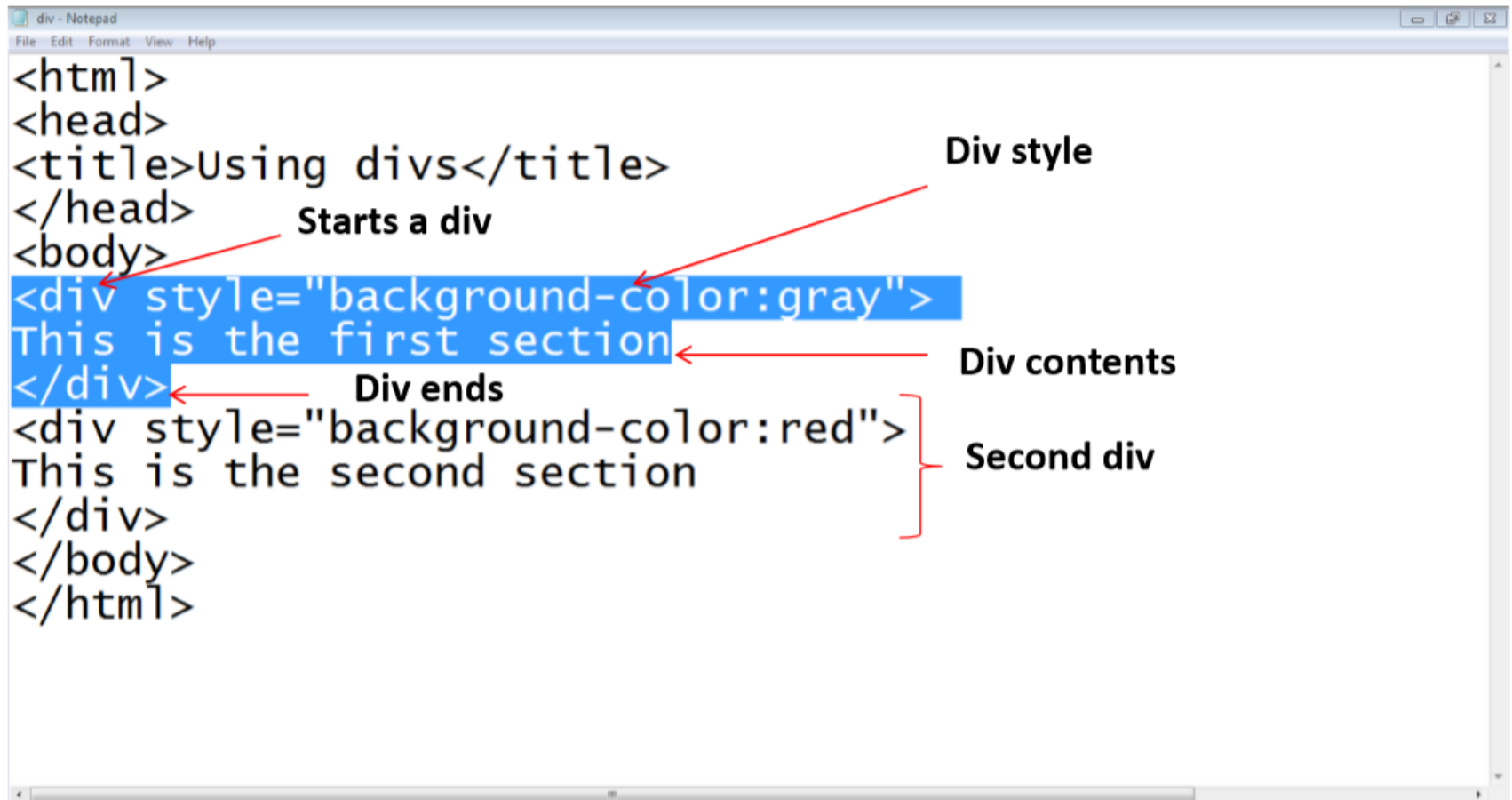
- Liquid Design
  - designs automatically fits to the screen
  - Use percentage values to declare height and width of sections.
- Fixed design
  - Designs remain fixed width
  - Use fixed values to declare height and width of sections



# The <div> tag

- Defines a division or section in an HTML document
- Visually, allows us to make containers that can be formatted.
- Can be declared as
  - `<div>.....</div>`

# The <div> tag



The screenshot shows a Notepad window titled "div - Notepad" with the following HTML code:

```
<html>
<head>
<title>Using divs</title>
</head>
<body>
<div style="background-color:gray">
This is the first section
</div>
<div style="background-color:red">
This is the second section
</div>
</body>
</html>
```

Annotations with red arrows and brackets point to specific parts of the code:

- Starts a div**: Points to the opening `<div>` tag.
- Div style**: Points to the `style="background-color:gray"` attribute.
- Div contents**: Points to the text `This is the first section`.
- Div ends**: Points to the closing `</div>` tag.
- Second div**: A bracket points to the entire second `<div>` block, including its opening tag, content, and closing tag.

# The <div> tag



# Coding the design



# Coding the design

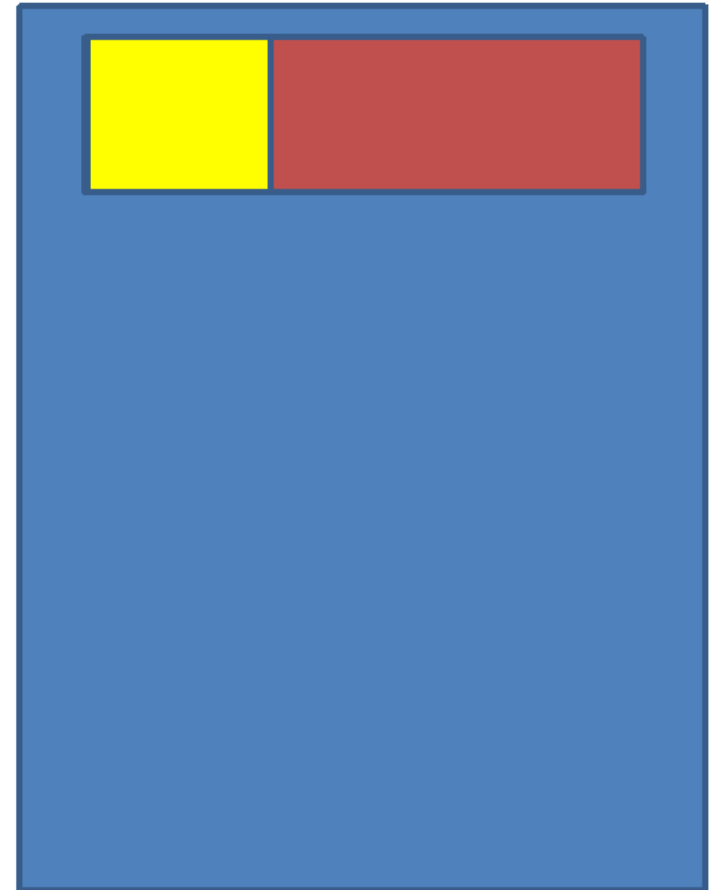
```
#header{
    background-color: lightblue;
    width:100%;
    height:50px;
    text-align: center;
}
#sidebar-left{
    float:left;
    width:15%;
    background-color: red;
}
#main{
    float:left;
    width:70%;
    background-color: lightgray;
}
#sidebar-right{
    float:left;
    width:15%;
    background-color: red;
}
#footer{
    clear:both;
    height: 50px;
    width: 100%;
    text-align: center;
    background-color: lightblue;
}
#sidebar-left, #main, #sidebar-right{
    min-height: 600px }
```

# Coding the design

```
<html>
  <head>
    <link href = "example.css" type = "text/css" rel= "stylesheet"
  </head>
  <body>
    <div id="header">Header</div>
    <div id="sidebar-left">Left</div>
    <div id="main">Main</div>
    <div id="sidebar-right">Right</div>
    <div id="footer">Footer</div>
  </body>
</html>
```

# Coding the design

- `<div id="container">`
- `<div id="header">`
- `<div id="logo"></div>`
- `</div> //header div ends`
- `</div> //container div ends`

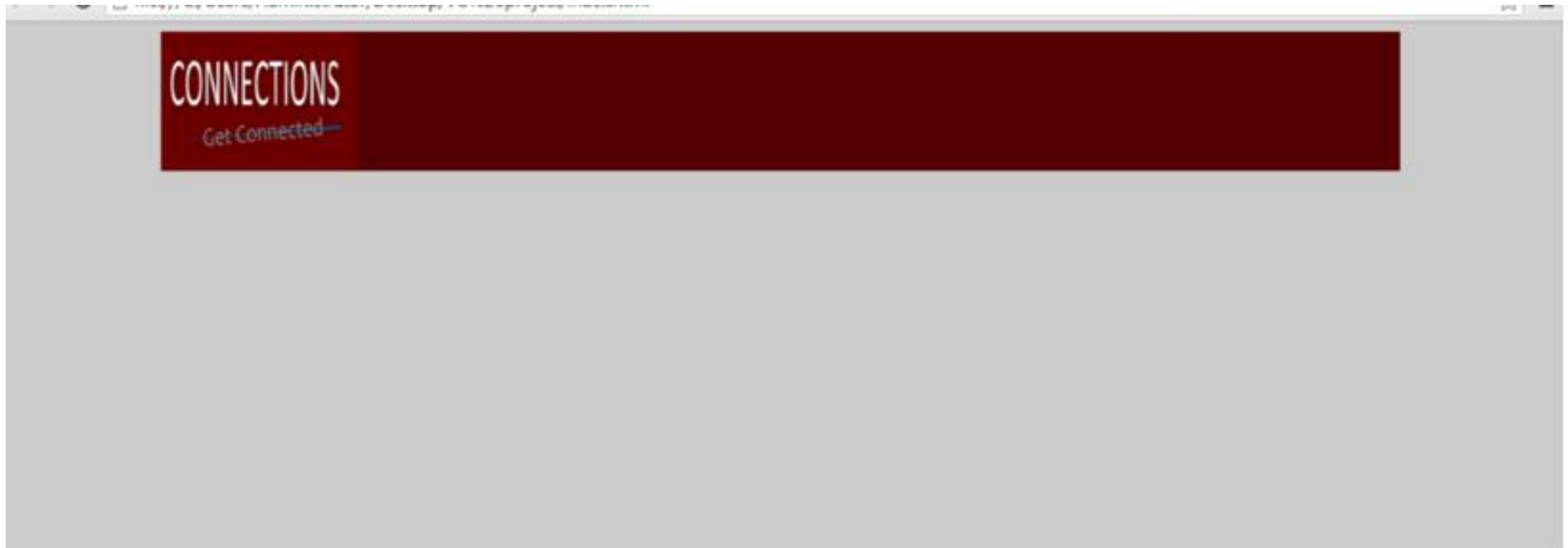


# Coding the design

- Divs required to make the structure:
- Main container
- Header
- Logo
- Centre container
- Form container
- Form heading
- Form row



# Coding the design



# Coding the design

The image shows a web application design for a user login page. At the top, there is a dark red header bar. On the left side of this bar, the word "CONNECTIONS" is written in a white, serif font, with the tagline "Get Connected" in a smaller, italicized font below it. The main content area has a light gray background. In the center, there is a white box with a dark red header that says "User Login". Below this header, there are two input fields: "Your ID" and "Your Password". To the right of each label is a white rectangular input field. Below the password field is a "Login" button with a gray gradient and a dark border.

CONNECTIONS  
Get Connected

User Login

Your ID

Your Password

Login

# Coding the design

**CONNECTIONS**  
Get Connected

## User Login

Your ID

Your Password

Login

## User Registration

Your Full Name:

Your Email Address:

Your Password:

Your Picture:

Choose File No file chosen

Register

# References

- **Chapter 9** Beginning HTML, XHTML, CSS, and JavaScript, by Jon Duckett, Wiley Publishing; 2009, ISBN: 978-0-470-54070-1.