UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Web Technologies

## Lecture Week Three

### Getting Started with CSS 3

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

**ON SILENT MODE PLEASE**

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# This week's agenda

- Introducing CSS 3
- Linking CSS and HTML together
- CSS Selectors and its types
- Vendor Prefixes
- Pseudo Classes
- Understanding Box Model
- CSS Properties for Layout Designing

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# This week's agenda

- Building layouts for the web
- Media Queries
- Understanding Mobile first approach
- Making your website responsive using media queries

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Getting started with CSS3

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Getting started with CSS3...

- CSS stands for Cascading Style Sheets.
- It describes how HTML elements are to be displayed on screen or paper.
- CSS can control layouts of multiple pages all at once ultimately saving lots of time.
- It can be written in 3 different ways.
  - **External**: A CSS written in an external file and linked to HTML.
  - **Internal**: CSS written in the HTML file enclosed by a <style> tag.
  - **Inline**: CSS written as an attribute in the HTML element.

# Getting started with CSS3

- CSS is used to define styles for pages, including the design, layouts and variations in display for different devices and screen sizes.
- CSS is also used to design layouts for mobile screens and tablets with the use of @media queries.
- CSS syntax can be divided in to 3 parts. Selector, property and value. This is the standard way of writing css.

```
selector{
          property:
value;
}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

*How to*

# LINK A STYLESHEET (CSS) FILE TO YOUR HTML FILE

# Linking CSS and HTML together…

- As said before, CSS can be written in 3 different ways.
- **Inline CSS**: Inline CSS is written directly into the html elements in a HTML document. This does not require you to import anything in the HTML document.

```
<div style="border: 1px solid #000">This has a border</div>
```

- **Internal CSS**: Internal CSS is also written in the HTML document itself. However this is written inside a <style> tag and should always be placed inside the <head> element.

```
<style> div{ border: 1px solid #000; }</style>
```

# Linking CSS and HTML together

- **External CSS**: This is the standard way of writing CSS. Your every CSS codes should be written in a different file and needs to be saved with a ".css" extension. The saved CSS file is then linked to an HTML document using a <link> tag.

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

- All three types of CSS has its own uses. The most correct way to write CSS is to create a new file (External CSS) and write all your styles in that file.

# CSS Selectors and its types

CSS Attribute Selector

CSS Element Selector

CSS Id Selector

CSS Class Selector

CSS Universal Selector

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# CSS selectors and its types...

- CSS selectors are used to target HTML element that we want to style.
- A selector is a component of CSS declaration.
- There are five varieties of CSS selectors.
  - CSS Universal Selector
  - CSS Element Selector
  - CSS ID Selector
  - CSS Class Selector
  - CSS Attribute Selector

# CSS selector: Universal Selector

- A universal selector selects all the elements in a web page.
- We can declare an universal selector with the help of an asterisk (*).
- Universal selectors can be used along with other selectors in combination but is highly discouraged to do it.

```
*{
        color: blue;
        font-size:
12px;
}
```

# CSS selector: Element Selector

- Element selector is also known as the type selector.
- The selector in CSS tries to match the element of the same name in HTML document it is linked to.
- The below example targets all the <ul> elements in a HTML document.

```
ul{
        border: 1px solid
#ccc;
        font-size: 12px;
}
```

# CSS selector: ID Selector

- ID selector helps to select a HTML element that has been given an ID name.
- The ID can be selected using a “#” symbol in the CSS.
- ID selector matches every HTML element having an ID attribute with the value the same as that of the selector, without the hash sign.

```
<header id=“header”></header>
```

```
#header{
          border: 1px solid
#ccc;
          font-size: 12px;
}
```

# CSS selector: Class Selector

- Class selector helps to select a HTML element that has been given a class name.
- The class can be selected using a “.” symbol in the CSS.
- Class selector matches every HTML element having a class attribute with the value the same as that of the selector, without the dot sign.

```
<div class=“container”></div>
```

```
.container{
          margin: 0
auto;
          width:
1200px;
}
```

# CSS selector: Attribute Selector…

- Attribute selector styles content according to the attribute and the attribute value mentioned in the square bracket.
- No spaces can be present ahead of the opening square bracket.
- Attribute selectors have different ways to match the values with the attribute key.

```
<input type="text">
```

```
Input[type="text"] {
          font-size:
16px;
          line-height:
1.2;
}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

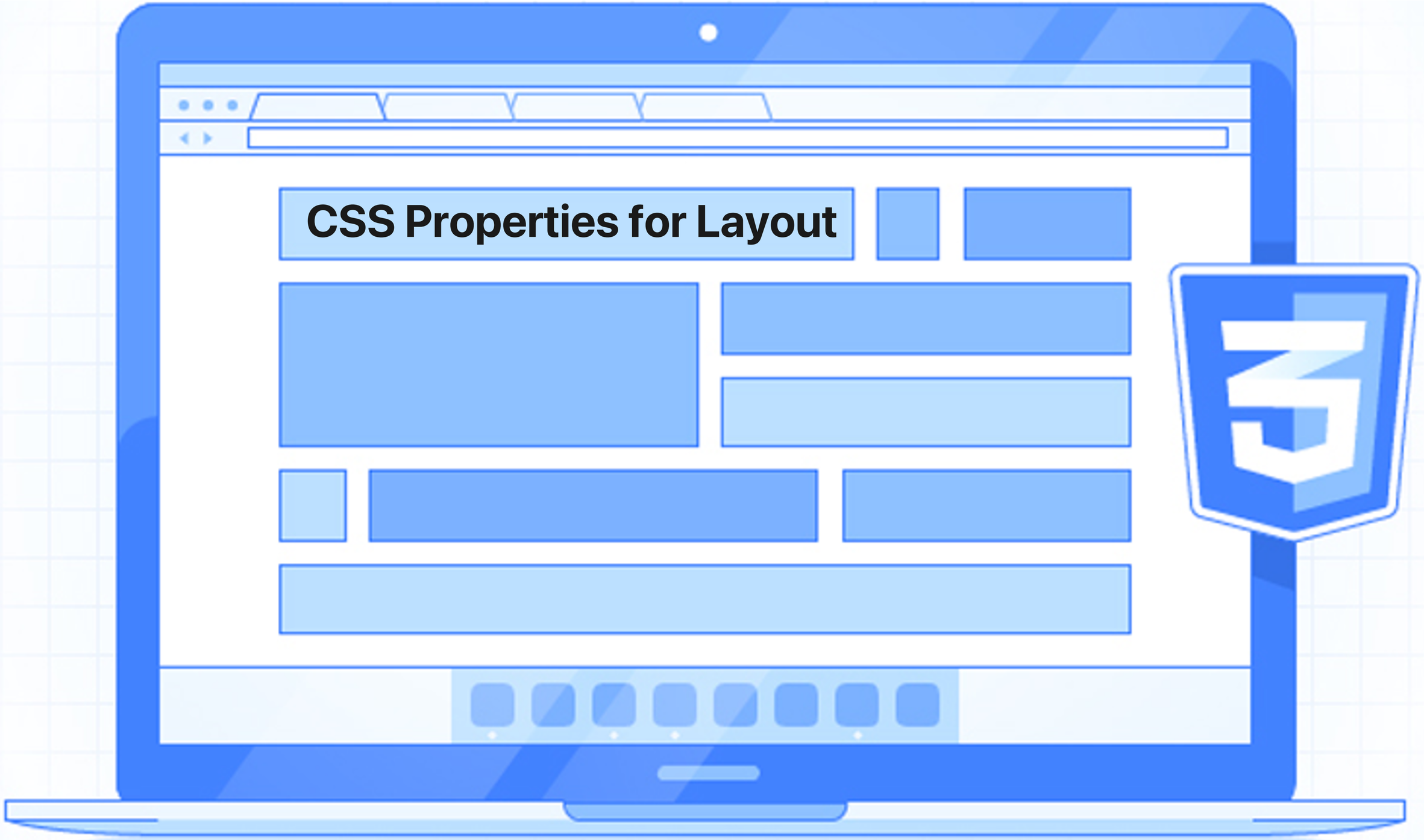
# CSS selector: Attribute Selector…

| Operator | Description |
|---|---|
| ~= | Matches elements whose attribute contains *value* as a word |
| \|= | Matches elements whose attribute matches *value* or begins with *value* followed by a dash. |
| ^= | Matches elements whose attribute begins with *value.* |
| $= | Matches elements whose attribute ends with *value.* |
| *= | Matches elements whose attribute contains *value.* |

# :hover

# Pseudo Classes

- Pseudo class is what we call a false class. It is denoted by a **colon ( : ).**
- It is a selector attached to HTML element to specify a special state.
- Pseudo-classes lets you apply a style to an element not only in relation to the content of the document tree, but also in relation to external factors like the history of the navigator (**:visited**, for example), the status of its content (like **:checked** on certain form elements), or the position of the mouse (like **:hover**, which lets you know if the mouse is over an element or not).


CSS Properties for Layout

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# CSS Properties for Layout Designing...

- Layout Designing is one of the major aspect of using CSS.
- It has come a long way since using “Table” based design to create page layouts.
- It is more flexible, dynamic and easy to create layouts.
- Layouts using CSS can be done using the combination of one or more CSS properties, each having its own respective domain.
  - Float Property
  - Position Property
  - Display Property

# Layout Designing: Float Property

- The CSS float property specifies how an element should float.
- It is used for positioning and formatting content.

  Eg.: Let an image float left to the text in a container.
- The property can have one of the following values:

  **Left**: The element floats to the left of its container.

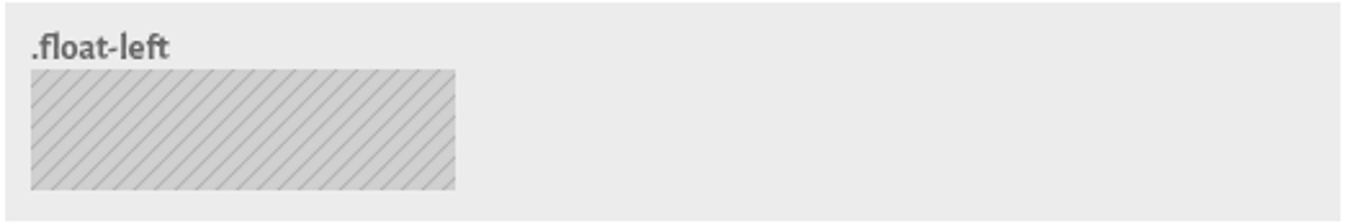
  **Right**: The element floats to the right of its container.

  **None**: The element does not float. This is default behavior.

  **Inherit**: The element inherits the float value of its parent.

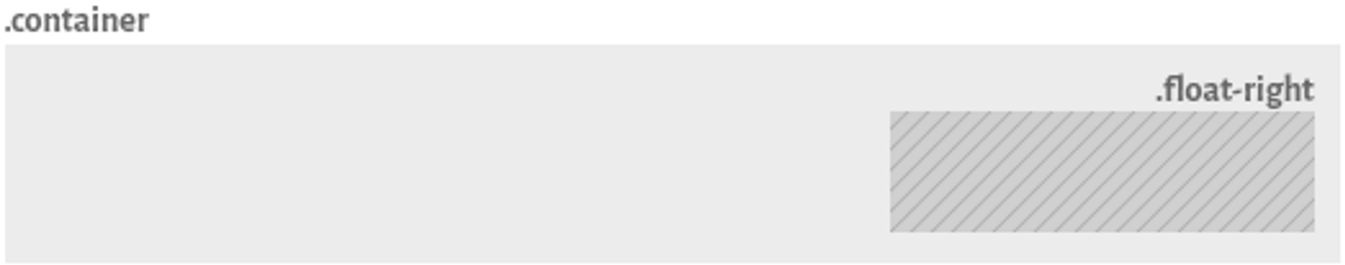
# Layout Designing: Float Left

.container




```
.float-left{

        float: left;

}
```

# Layout Designing: Float Right

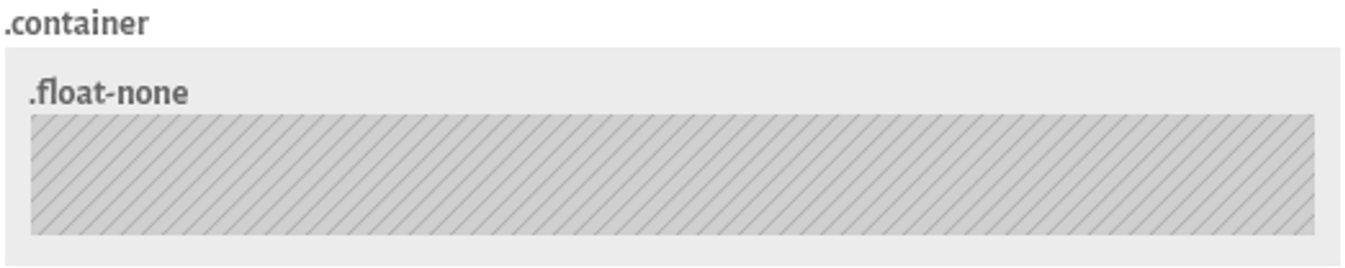



```
.float-right{

        float: right;

}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Layout Designing: Float None




```
.float-none{

        float: none; /*Default value*/

}
```

# Layout Designing: Clear Property

- The clear property specifies what elements can float beside the cleared element and on which side.
- The property can have one of the following values:

  **none**: Allows floating on both sides. This is default.

  **left**: No floating elements allowed on the left side.

  **right**: No floating elements allowed on the right side.

  **both**: No floating elements allowed on either sides

  **inherit**: The element inherits the clear value of its parent.
- The **parent element** of the element with float should have a clear fix hack.

# Layout Designing: Display Property

- Display property is the most important CSS property for controlling layout.
- Any html element can either have a default value of block or inline.
- The property can have one of the following layout values:

  **none**: Hides the element.

  **block**: Starts on the new line and takes full width available.

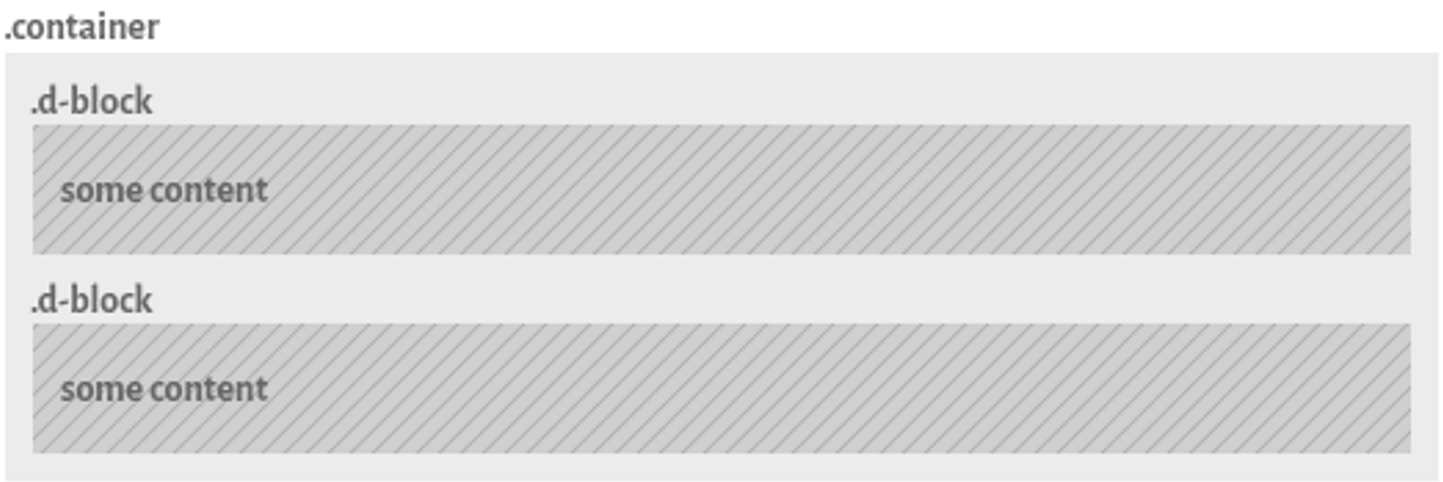
  **inline**: Starts on the same line and only takes as much width as necessary.

  **Inline-block**: Similar to inline, but allows to give a width and height.

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Layout Designing: Display Block

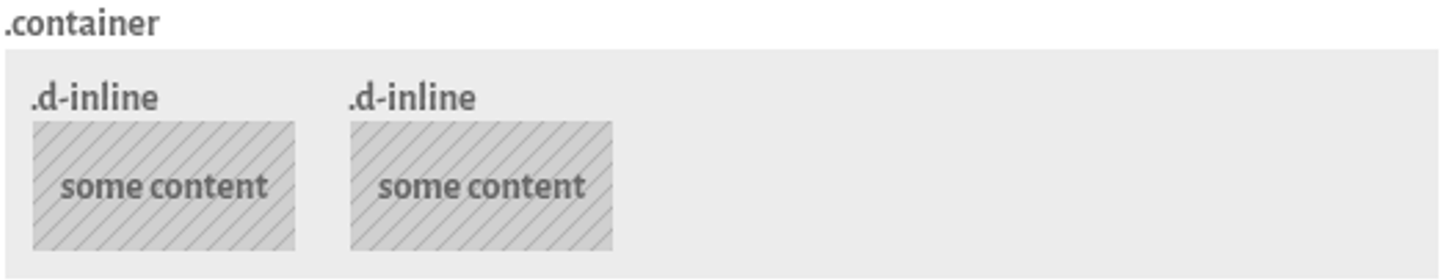



```
.d-block{

        display: block;

}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Layout Designing: Display Inline

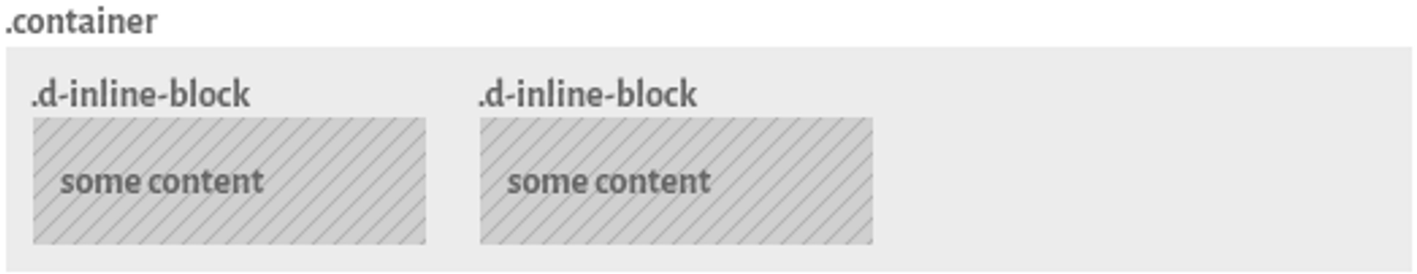



```
.d-inline{

        display: inline;

}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Layout Designing: Display Inline Block




```
.d-inline-block{

        display: inline-block;

}
```

# Layout Designing: Position Property

- The position property specifies the type of positioning of an element.
- The property can have one of the following values:

  **static**: This is the default position of any elements. It is not affected by top, left, right, bottom values.

  **relative**: The element is positioned in relative to its normal position.

  **absolute**: The element is positioned relative to the nearest positioned ancestor. If the element has no positioned ancestor it uses the document body as its ancestor.

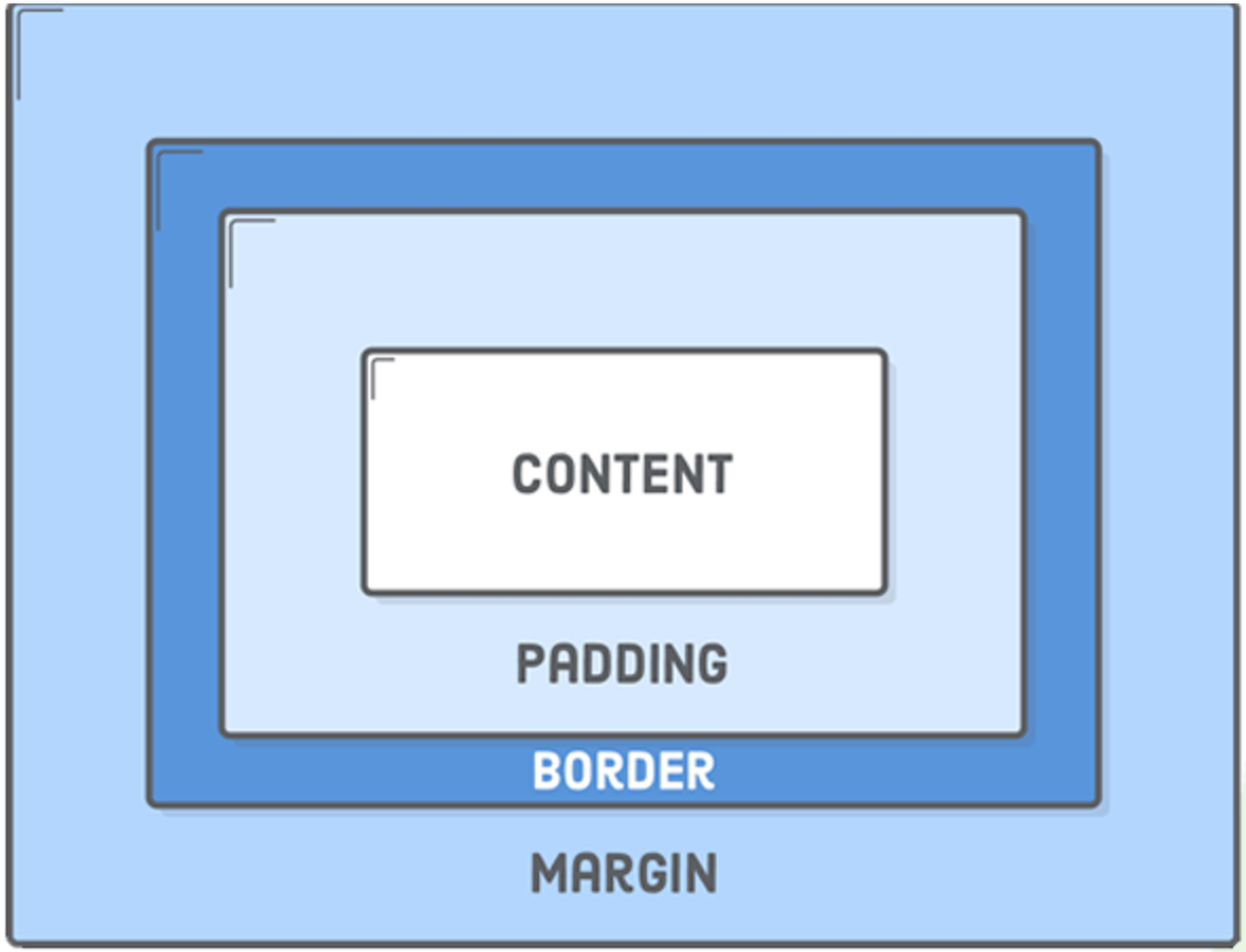
  **fixed**: The element is positioned relative to the viewport.

  **sticky**: The element is positioned based on the user's scroll position.

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# CSS Box Model



# CSS Box Model...

- All HTML elements can be considered as a box. The term box-model is used when talking about design and layout.
- It consists of margin, border, padding and content in the exact order.
- It is essentially a box that wraps around every HTML element.

  **Margin**: Clears the area outside the border. The margin is transparent.

  **Border**: A border that goes around the padding and content.

  **Padding**: Clears an area around the content. The padding is transparent.

  **Content**: The content of the box, where text, images, etc. appears.

# CSS Box Model

- A simple demonstration of how the box-model css will look like.

```
<div class="box-model"></div>
```

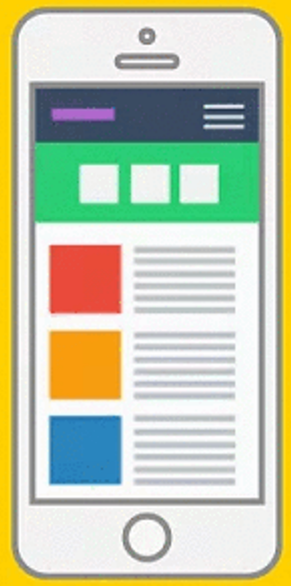
```
<style>
        .box-model{
                width: 300px;
                border: 15px solid
green;
                padding: 50px;
                margin: 20px;
        }
</style>
```


be back in 10 minutes...

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layouts for the web



UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layouts...

- So far we have understood that HTML is used to write the structure and CSS is used to give style to it.
- To create a layout we need to work with both structure and style.
- The layouts are not only limited to desktops, but you need to make sure that it looks seamless and have a similar usability throughout multiple devices.
- Making your website compatible with cross devices makes it more accessible for your users.

# Building Layouts...

- It is very important to get your structure right before giving it a style.
- There are certain principle of writing structure for the layouts.
- When creating a layout always focus on the direction of the layout you are building.
- Any website can be made in two directions. LTR (Left to Right) or RTL (Right to Left).
- When creating a website with LTR direction, your layout needs to start from left and go towards right.

UNIVERSITY OF WOLVERHAMPTON

# Building Layouts (LTR)

HERALD COLLEGE KATHMANDU

```
<div class="logo">
          <img src="image.jpg" alt="Logo">
</div>
<nav class="nav">
          <ul>
                    <li>
                              <a
href="index.html">Home</a>
                    </li>
                    <li>
                              <a
href="about.html">About</a>
                    </li>
                    ……
          </ul>
</nav>
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layouts (RTL)

```
<nav class="nav">
          <ul>
                    <li>
                              <a
href="index.html">Home</a>
                    </li>
                    <li>
                              <a
href="about.html">About</a>
                    </li>
                    ……
          </ul>
</nav>
<div class="logo">
          <img src="image.jpg" alt="Logo">
</div>
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layouts...

- Always group multiple elements where necessary. Grouping helps you position of multiple elements at one instance.
- As seen earlier when writing structure codes for layout direction is important.
- However, it is not just the horizontal direction that matters. But, the vertical direction matters as well.

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layout...

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layout...



Can you tell how many groupings should be done in the layout above?

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Building Layouts...

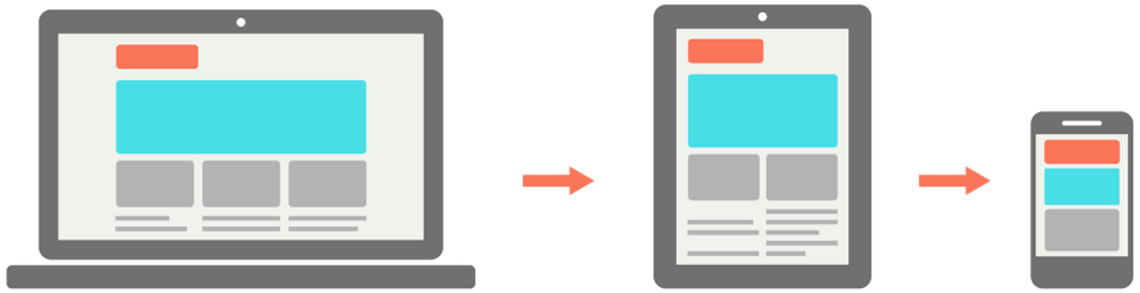
```
<div class="container">
          <div class="row">
                    <div class="col-3">
                              <figure>
                                        <img src="image.jpg"
alt="Heading" >
                              </figure>
                              <h2>Lorem ipsum dolor</h2>
                              <p> Lorem ipsum dolor sit amet,
                              consectetur adipiscing elit, sed do eiusmod
tempor
                              incididunt ut labore et dolore magna
aliqua. </p>
                    </div>
                    <div class="col-3">...</div>
                    <div class="col-3">...</div>
          </div>
</div>
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

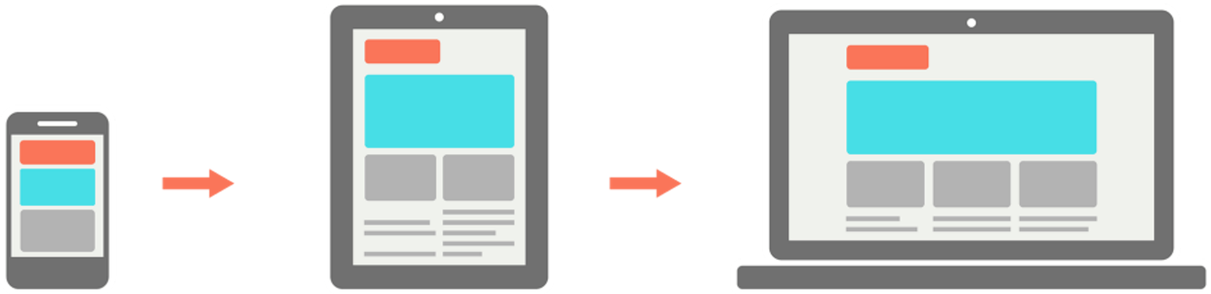
# Building Layouts...

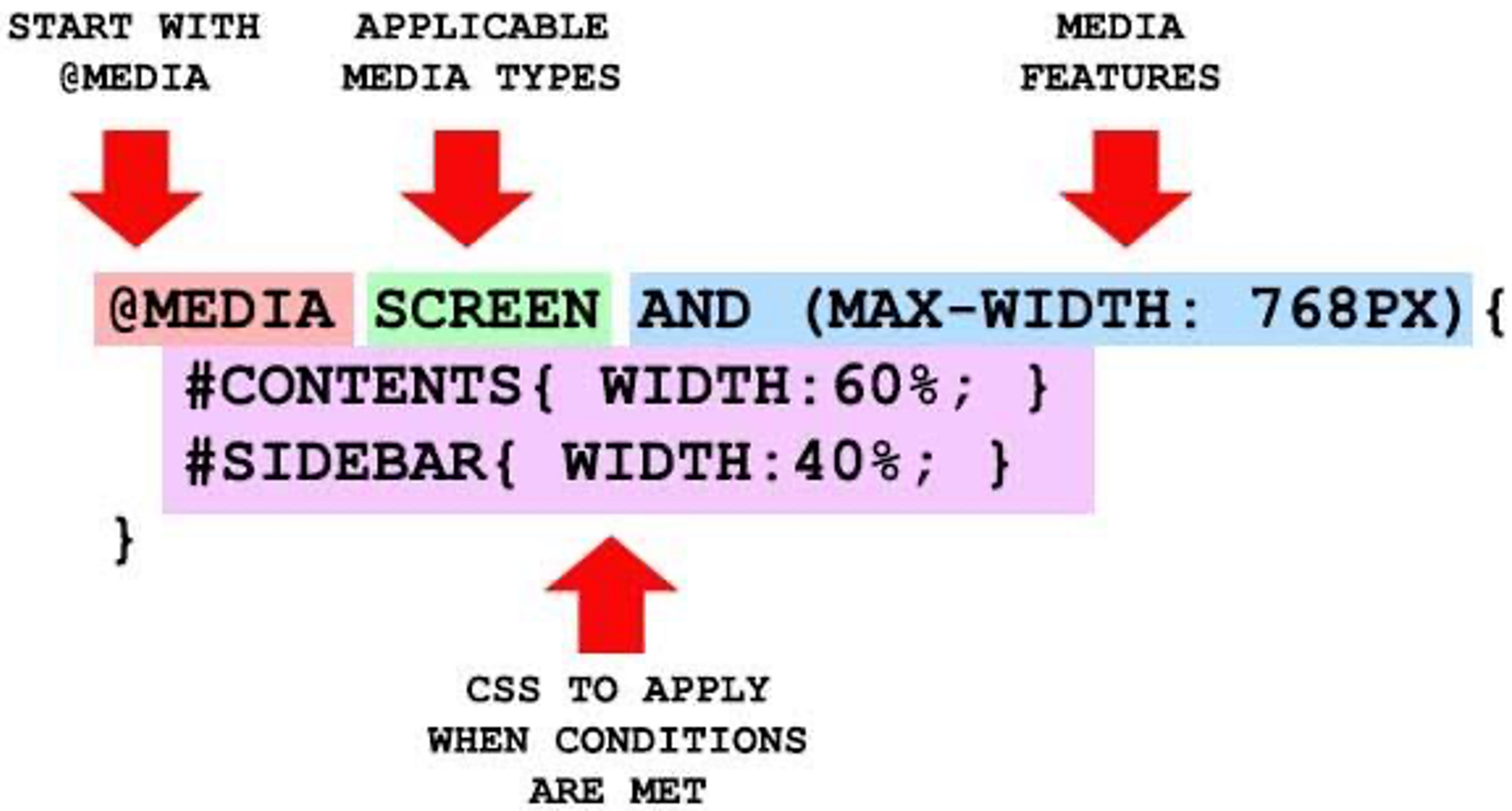
- Web layouts can be made in two approach.



Responsive Web Design

Mobile First Web Design




START WITH @MEDIA
APPLICABLE MEDIA TYPES
MEDIA FEATURES
@MEDIA SCREEN AND (MAX-WIDTH: 768PX){
#CONTENTS{ WIDTH:60%; }
#SIDEBAR{ WIDTH:40%; }
}
CSS TO APPLY WHEN CONDITIONS ARE MET

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Media Queries...

- Media Query is a technique introduced in CSS3.
- It use the @media rule to include a block of CSS properties only if a certain condition is true.
- For mobile viewports to support media queries, you will require to include a <meta> tag in the <head> element.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Mobile First Approach

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Mobile First Approach...

- Mobile First, as the name suggests, means that we start the design from the mobile end.
- The mobile-end has more restrictions on the design.
- Gradually expand its features to create a tablet and desktop version.
- While writing styles for mobile first approach. We start with designing for mobile version then use media queries to target larger devices.

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Mobile First Approach

```
.container{
          width: 100%;
          max-width: 450px;
          margin: 0 auto;
          padding: 0 15px;
}

@media screen and (min-width: 560px){
          .container{
                    max-width: 530px;
          }
}

@media screen and (min-width: 728px){
          .container{
                    max-width: 700px;
          }
}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Responsive Web Design...

- Responsive web design is an approach of creating your desktop version application accessible to various smart devices.
- With responsive design, we cut out the unnecessary elements or shrink the design to fit to the device.
- While writing styles for responsive web design. We start with designing for desktop version then use media queries to target smaller devices.

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Responsive Web Design

```
.container{
        width: 100%;
        max-width: 1140px;
        margin: 0 auto;
        padding: 0 15px;
}

@media screen and (max-width: 728px){
        .container{
                max-width: 600px;
        }
}

@media screen and (max-width: 630px){
        .container{
                max-width: 500px;
        }
}
```


HERALD COLLEGE KATHMANDU
MOBILE
FIRST
Vs
RESPONSIVE
WEB DESIGN

# Mobile First Vs Responsive Design

| Mobile First Design | Responsive Web Design |
| --- | --- |
| Minimalistic way of designing | Recommended by Google! |
| Can be leveraged and added more power in larger versions | Deliver better user experience for users using across devices |
| Faster to load and delivers flawless experience | Consistent brand experience |
| Idle for if your user base is widely based in mobile devices. | Idle for if your user base is widely based in desktop |
| The desktop versions are not readily available early | The mobile experience cannot be defined early |

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Mobile First Vs Responsive Design...

```
.container{
        width: 100%;
        max-width: 450px;
        margin: 0 auto;
        padding: 0 15px;
}

@media screen and (min-width: 560px){
        .container{
                max-width: 530px;
        }
}

@media screen and (min-width: 728px){
        .container{
                max-width: 700px;
        }
}
```

```
.container{
        width: 100%;
        max-width: 1140px;
        margin: 0 auto;
        padding: 0 15px;
}

@media screen and (max-width: 728px){
        .container{
                max-width: 600px;
        }
}

@media screen and (max-width: 630px){
        .container{
                max-width: 500px;
        }
}
```

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# What to expect in Lab

- Revision on CSS, layouts and box-model
- Q & A session on additional common CSS properties.

  (Research Required)
- Feedback on your project idea and proposal.
- Practical exercises based on the lecture and beyond.

  (Research is very necessary)

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# What to expect in Lab

- Revision on Mobile First Approach and Responsive Web Design.
- Practical exercises based on the lecture and beyond.

  (Research is very necessary)

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Before you come for Lab, Research!!

- Mozilla Developers Network: CSS
- CSS: W3C Schools
- CSS: Tutorial Points
- CSS3: Tutorial Points
- CSS Pseudo classes

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Before you come for Lab, Research!!

- Mobile First Approach
- Responsive Web Design Breakpoints
- Media Queries

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

# Thank you!