CSS-2

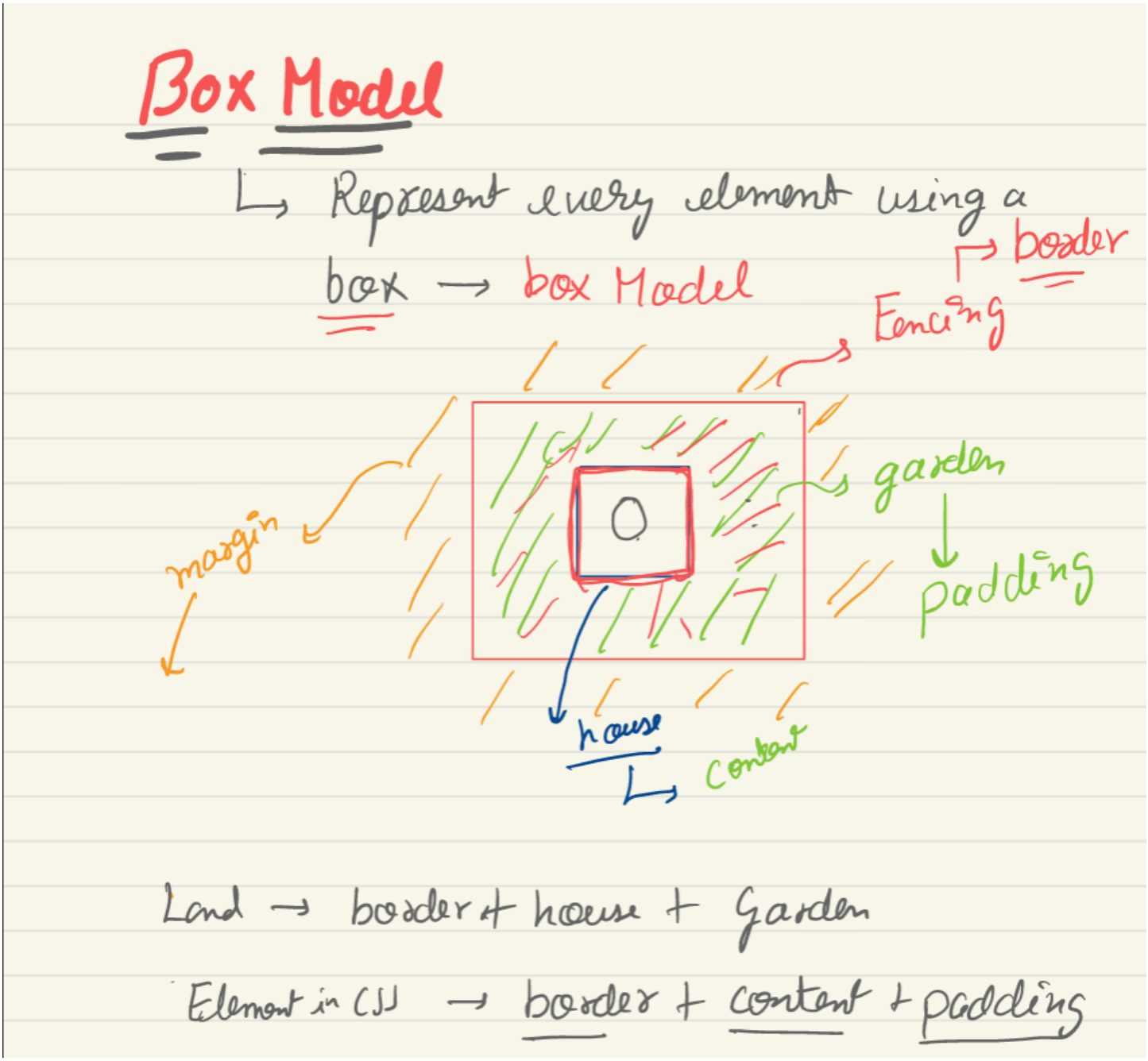
Agenda

- Box Model
- Box sizing
- Overflow with height and width
- Display : inline, block and inline block

Box Model

What do you understand by the Box model?

- Represent every element using a box(squares and rectangles) is called **Box Model**.
- Anology - let us say you become a billionaire one day by winning a lottery and you start by buying land.
- You will start by fencing, The house is going to have a garden surrounding it, and your area outside the house which is owned by the government.
- Here (**x --> y in CSS** can be read as x represents y in CSS)
    - House --> Element
    - Garden --> Padding
    - Outside area --> margin
    - Fencing --> Border
    - Land -> Border + House + Garden
    - Eleemnt in CSS -> border + content + padding

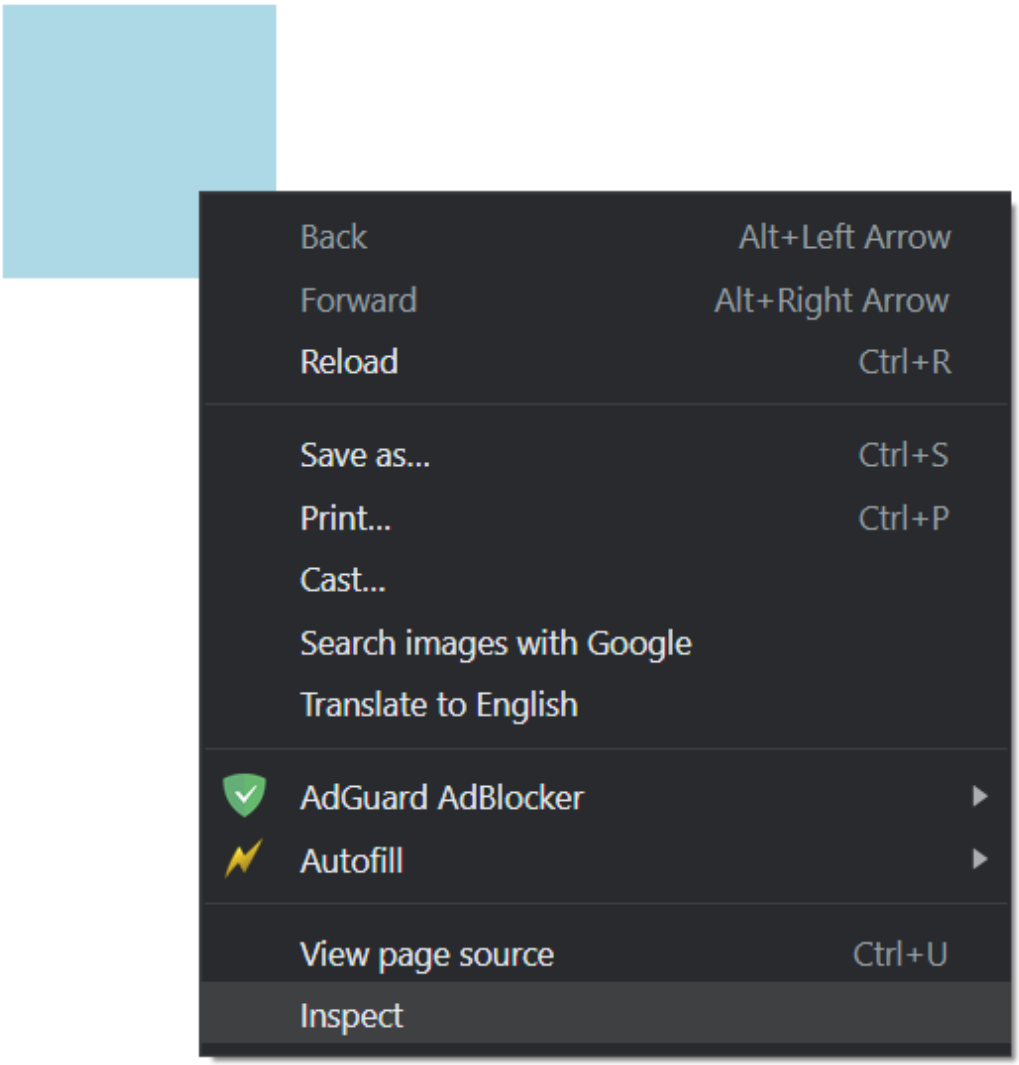


We will start with box_border_html.html

```
<!DOCTYPE html>
 <html lang = "en">
 <head>
     <meta charset = "UTF - 8">
     <meta http-equiv = "X - UA - Compatible" content = "IE = edge">
     <meta name = "viewport" content = "width = device - width, initial-
scale = 1.0">
    <title>Document</title>
 <style>
     .box{
```

```
        /* content */
        height:100px;
        width:100px;
        background-color: lightblue;
    }
</style>
</head>
<body>
    <div class = "box"></div>
</body>
</html>
```

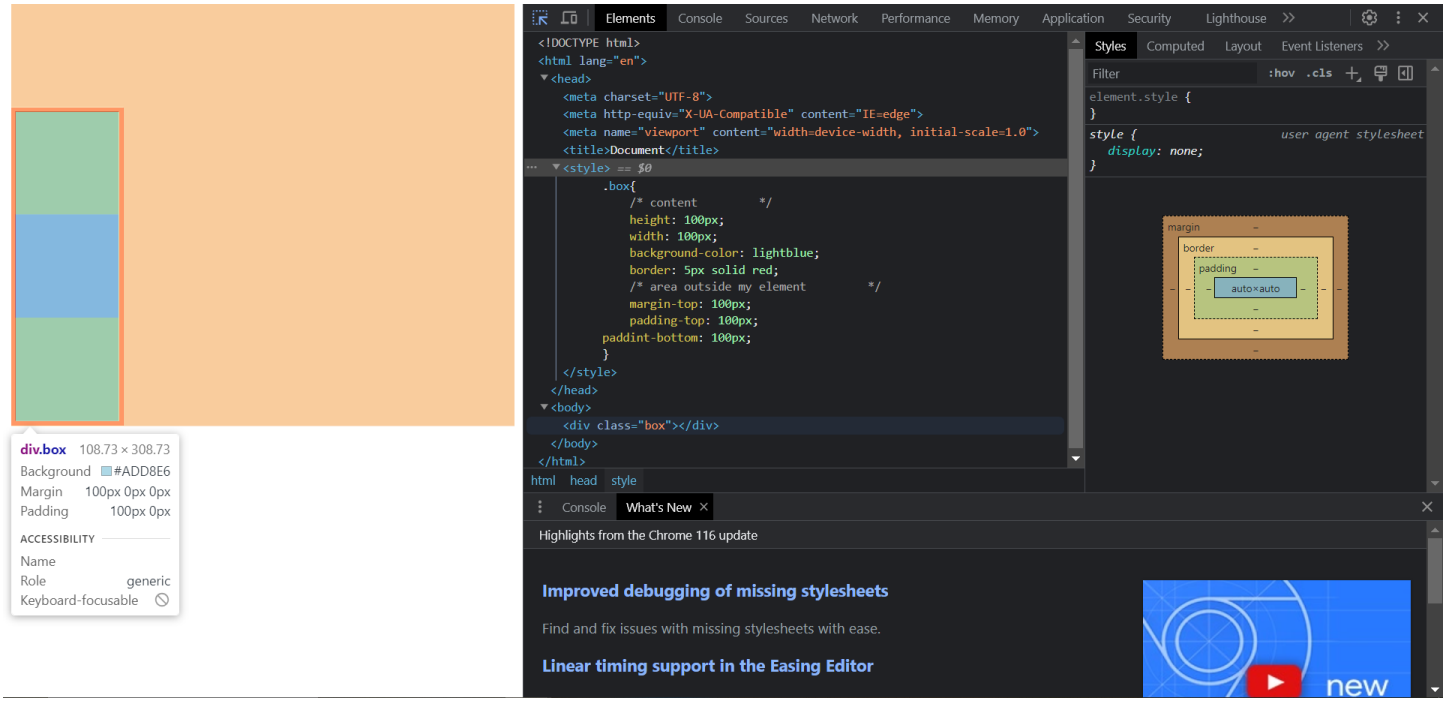
Run this in Browser right-click on the box on the screen and select inspect. Now we will have two areas first is html part and the other is css part. We have certain more things but we will go to the box we have created.



Add border `border: 5px solid red`.

- If I add the margin of 100px to the top, will the size of the box increase?
  --> No margin is something outside the box, it's not inside the box so the size remains as it is.
- If I add padding of 100px it the size of the box change? If yes how?
  --> Yes, the size of the box will change. The box becomes larger and a padding of 100px is added to all the sides. If we wish to add padding only on one side we can use `padding-top: 100px`.

Add border, padding, and margin to this box.




```
<style>
    .box{
        /* content          */
        height: 100px;
        width: 100px;
        background-color: light-blue;
        border: 5px solid red;
        /* area outside my element          */
        margin-top: 100px;
        padding-top: 100px;
        padding-bottom: 100px;
    }
</style>
```

Height of the element is -> border bottom + border top + padding top + height of content
Width of the element is -> horizontal border + horizontal padding + width of the content
Element -> Content + Padding + Border

CSS Height and Width Behavior

Understanding how height and width behave in different types of elements is crucial for mastering CSS layout and design. Here's a detailed explanation focusing on inline and block-level elements.

Height and Width Behavior

Inline Elements

Characteristics:

- Inline elements do not start on a new line.
- They take only as much width as needed by their content.
- Examples: `<a>`, `<span>`, `<img>`

Height:

- The height of inline elements is defined by the content they contain.
- You cannot explicitly set the height of inline elements. The height property does not affect these elements directly.

Width:

- The width of inline elements is also determined by the content.
- Setting width explicitly does not change the element's display behavior.

Example:

```
<span class="inline-element">This is an inline element.</span>
```

```css
.inline-element {
    height: 100px; /* No effect */
    width: 200px;  /* No effect */
    background-color: lightblue;
}
```

Block-level Elements

Characteristics:

- Block-level elements start on a new line.
- They take up the full width of their parent container by default.
- Examples: `<div>`, `<p>`, `<h1>`, `<section>`

Height:

- The height of block-level elements is determined by the content if not explicitly set.
- You can set a user-defined height for block-level elements.

Width:

- By default, the width of block-level elements is the full width of the parent container.
- You can set a user-defined width for block-level elements.

Example:

```html
<div class="block-element">This is a block-level element.</div>
```

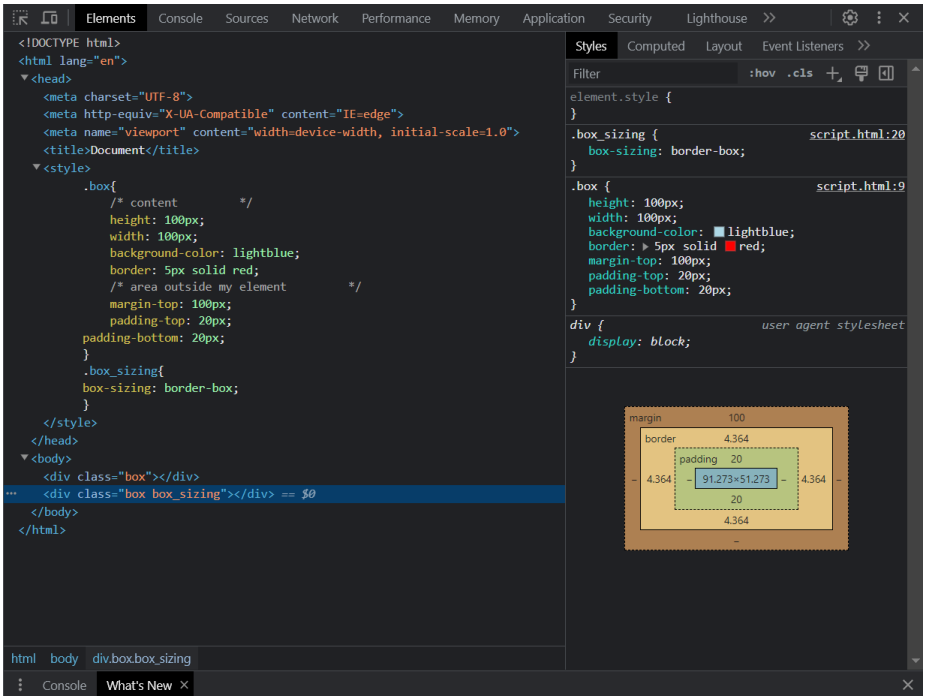
```css
.block-element {
    height: 100px; /* User-defined height */
    width: 200px;  /* User-defined width */
    background-color: lightcoral;
}
```

Summary

- **Inline Elements:**
  - Height: Defined by content, cannot be explicitly set.
  - Width: Defined by content, cannot be explicitly set.
- **Block-level Elements:**
  - Height: Defined by content or user-defined.
  - Width: Full width of parent container by default or user-defined.

Box Sizing

Another method to write the above code is using box-sizing. Look at the code and output given here.




```
<!DOCTYPE html>
 <html lang = "en">
 <head>
     <meta charset = "UTF - 8">
     <meta http-equiv = "X - UA - Compatible" content = "IE = edge">
     <meta name = "viewport" content = "width = device-width, initial-
scale = 1.0">
    <title>Document</title>
 <style>
    .box{
        /* content          */
```

```
        height: 100px;
        width: 100px;
        background-color: lightblue;
        border: 5px solid red;
        /* area outside my element          */
        margin-top: 100px;
        padding-top: 20px;
        padding-bottom: 20px;
    }
    .box_sizing{
        box-sizing: border-box;
    }

</style>
 </head>
 <body>
     <div class = "box"></div>
     <div class = "box box_sizing"></div>
 </body>
 </html>
```

- Can you spot the difference? What has happened over here?

--> On adding an extra property `box-sizing: border-box;`, the size of my box is reduced. This implies that something is subtracted from by border box.

Calculating Height and Width of Element

1. **Content Box =** content height + verticle border + vertical padding.
2. **Border Box =** content height - verticle border - vertical padding.

The same is true for the **width**.

Adding 20px on left and right:

```
    .box{
        /* content          */
        height: 100px;
        width: 100px;
        background-color: lightblue;
```

```
        border: 5px solid red;
        /* area outside my element          */
        margin-top: 100px;
        padding-top: 20px;
            padding-bottom: 20px;
        padding-right: 20px;
        padding-left: 20px;
    }
```

Then again the padding width is subtracted from our box.

Summing Up

All these variables padding margin borders are initialized with the creation of the object. We changed it as per our requirement.

- **Element** --> Content + Padding + Border
- **Element Height**
  default way --> height (user-defined) + padding-top + border-top + padding-bottom + border-bottom
  border box --> height (element) - padding-top - padding-bottom - border-top - border-bottom
  Note: Simiarly width will also be calculated.

Use case

Que : We usually use borderbox. Why?
`Ans`: Because these boxes contain some content. We define some height and width i.e. dimensions. Now the designer has some specifications and we want the element to be in the same dimensions.

CSS Display Properties: Block, Inline, and Inline-Block

In CSS, elements can be styled with different `display` properties which affect their layout and behavior on the page. The three main types are `block`, `inline`, and

`inline-block`. Below is an explanation of these properties, including their default behavior and how they interact with height, width, padding, margin, and borders.

Default Display Property of User-Agent Stylesheet

User-agent stylesheets are the default styles applied by browsers to HTML elements. These styles ensure that web pages have a consistent appearance across different browsers even if no CSS is provided by the developer. Here are the default display properties for common elements:

- **Block Elements:** These elements take up the full width available and start on a new line.
  - Examples: `<div>`, `<h1>` to `<h6>`, `<p>`, `<ul>`, `<li>`
- **Inline Elements:** These elements take only as much width as needed by their content and do not start on a new line.
  - Examples: `<span>`, `<a>`, `<img>`, `<strong>`
- **Inline-block Elements:** These elements are similar to inline elements but can have user-defined width and height.
  - Examples: Elements with `<input>`, `button`

Detailed Explanation with Code Example

1. Block Elements

Characteristics:

- Default behavior: Full width of the parent container.
- Starts on a new line.
- Can have user-defined width, height, padding, margins, and borders.

Example:

```
<div class="block">This is a block-level element.</div>
```

```
.block {
    background-color: lightblue;
```

```
    height: 100px;  /* User-defined height */
    width: 100px;   /* User-defined width */
    padding: 10px;  /* Padding in all directions */
    margin: 40px 0 0 20px; /* Margin (top and left) */
    border: 2px solid red; /* Border (top and left) */
}
```

2. Inline Elements

Characteristics:

- Default behavior: Only as wide as the content.
- Does not start on a new line.
- Cannot have user-defined width or height set.
- Can have padding, margins, and borders, but margins only affect horizontally.

Example:

```
<span class="inline">This is an inline element.</span>
```

```
.inline {
    background-color: lightgreen;
    height: 100px;  /* User-defined height (does not work) */
    width: 200px;   /* User-defined width (does not work) */
    display: inline;
    padding: 10px;  /* Padding in all directions */
    margin-left: 10px; /* Horizontal margin works */
    margin-top: 10px;  /* Vertical margin does not work */
    border: 5px solid red; /* Border in all directions */
}
```

3. Inline-block Elements

Characteristics:

- Combines the characteristics of both inline and block elements.
- Does not start on a new line.

- Can have user-defined width and height set.
- Can have padding, margins, and borders.

Example:

```
<div class="inline-block">This is an inline-block element.</div>
```

```
.inline-block {
    background-color: mediumorchid;
    display: inline-block;
    height: 100px;  /* User-defined height */
    width: 100px;   /* User-defined width */
    padding: 10px;  /* Padding in all directions */
    margin-left: 10px; /* Horizontal margin works */
    margin-top: 10px;  /* Vertical margin  work */
    border: 5px solid red; /* Border in all directions */
}
```