

CMPU 1031 - Web Development 1

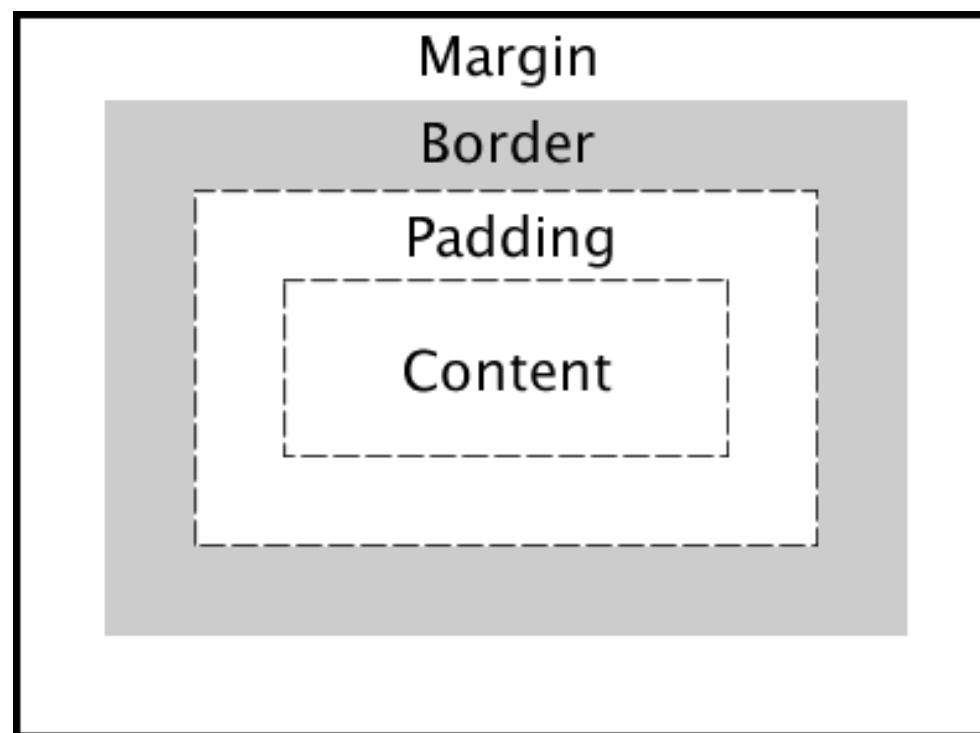
Lecturer: Dr Fatmaelzahraa Eltaher

Week 4: More on CSS



The CSS Box model

- To the browser, each element on the page is treated as a box. By following this idea, we are able to set up some properties of our HTML elements considering not only their content, but their distance from other elements and the space they occupy in the browser.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.



The CSS Box model

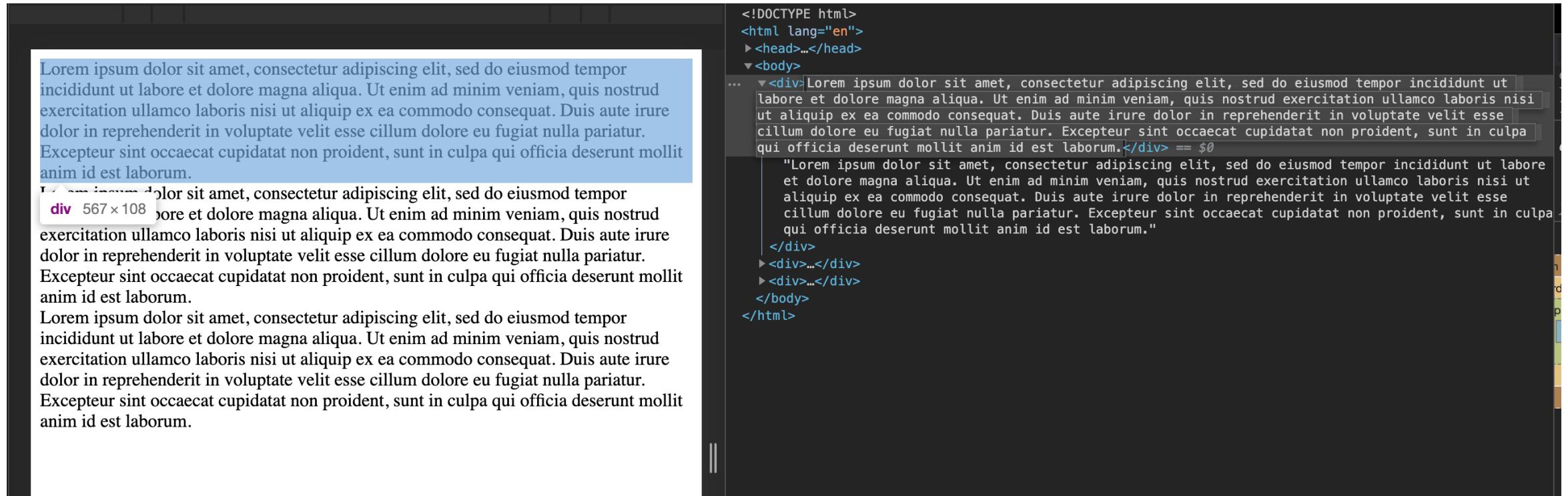
- **Content**
 - The content is what displays on the page.
 - This can be thought of as the content of the box.
- **Padding**
 - Padding adds some area around the content in the box.
 - The padding will be transparent.
- **Border**
 - The border goes around the padding and the content.
- **Margin**
 - Margin adds some area outside the border.
 - The margin will be transparent.

Check the following code example:

https://codepen.io/fatma_eltaher/pen/dPYVrMB

Width and height

- Width and height are used to control the inner content, considering how large and how tall this content will be
- HTML tags have default widths and heights, and we can change them using CSS
- Notice the following page. It contains three `<div>` tags with text. Each of them have, by default, a width of 567 pixels and a height of 108 pixels.



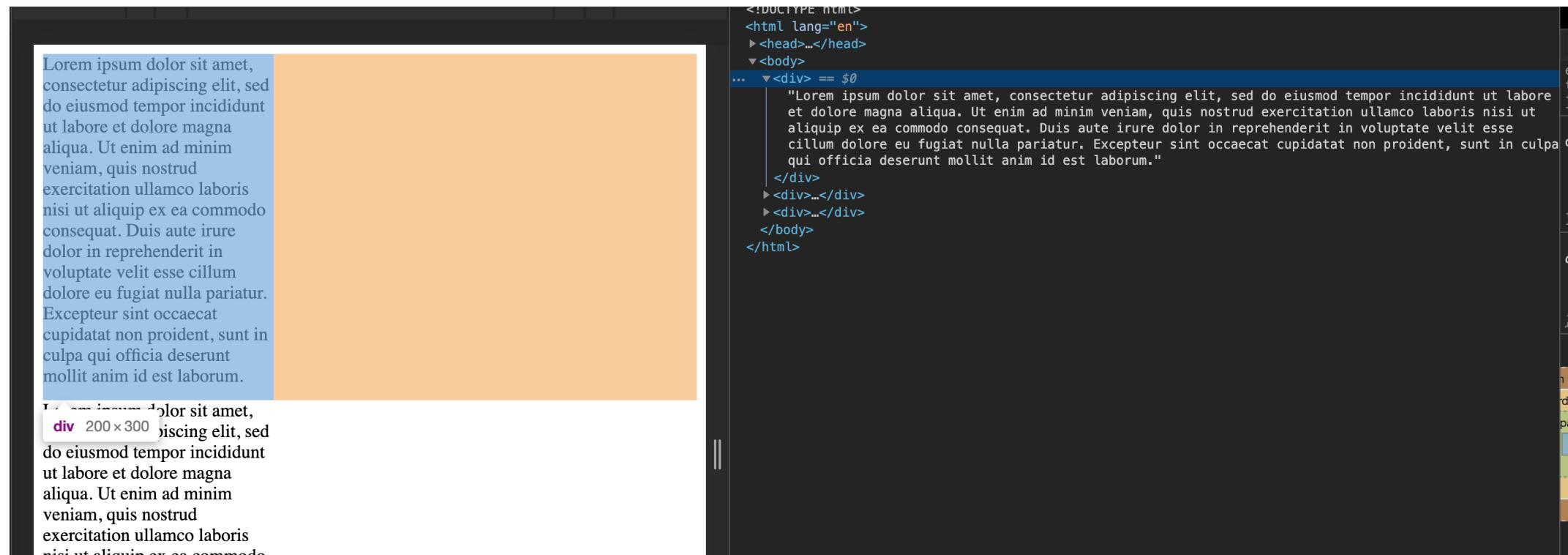
The screenshot shows a code editor with an HTML file open. The file contains three identical `<div>` blocks, each with the same content and styling. The content is a block of Latin placeholder text (Lorem ipsum).

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    ... <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div> == $0
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."
  </div>
  <div>...</div>
  <div>...</div>
</body>
</html>
```

Width and height

- Using CSS, we can change the width and height of those divs to 200 px and 300 px.

```
div {  
    width: 200px;  
    height: 300px;  
}
```



Width and height

- Notice that, if we change the background colour, we can clearly see the new size of the div in the box model.

```
div {  
    width: 200px;  
    height: 300px;  
    background-color: pink;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

LOREM IPSUM

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Border

- The border goes around an element. A number of border properties can be changed, and they allow not only to style our website, but to bring attention to certain elements.
- The main border properties to be changed are border-width, border-color, and border-style.



border-width:
how thick a
border should be
(pixels are
commonly used)



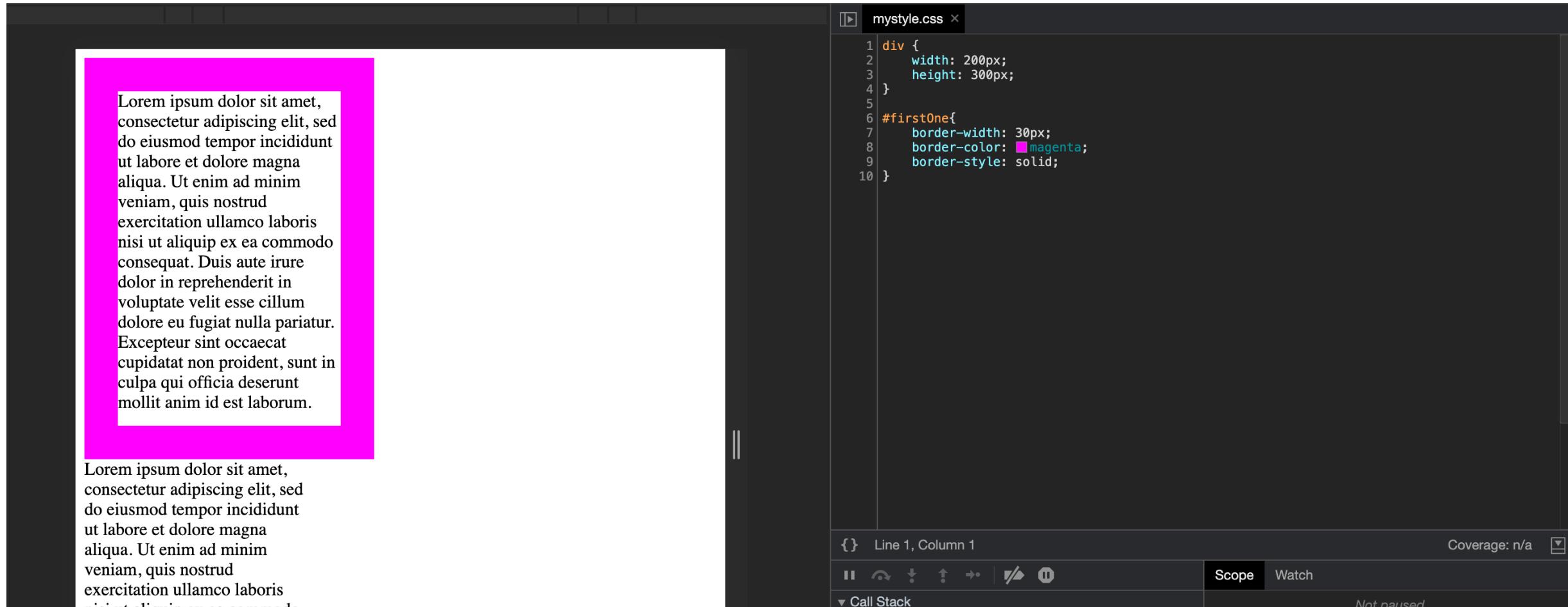
border-color: what
colour a border
should have



border-style: how
the line style
should look like

Border-width

- Suppose we decide to set up a border to one of our divs. We've set up a border 30px thick. Notice how now the div is no longer 200x300, but 260x360 as we added 30px for each border side



The screenshot shows a code editor window with a dark theme. On the left is a preview of a web page. The page contains two blocks of text. The top block is enclosed in a large magenta-bordered box, demonstrating the effect of a 30px border on the element's dimensions. The bottom block is a smaller, unbordered box. On the right is the 'mystyle.css' file, which contains the following CSS code:

```
1 div {  
2     width: 200px;  
3     height: 300px;  
4 }  
5  
6 #firstOne{  
7     border-width: 30px;  
8     border-color: magenta;  
9     border-style: solid;  
10}
```

The code defines a general rule for all div elements and a specific rule for an element with the ID 'firstOne'. The specific rule adds a 30px border to all sides, changes the border color to magenta, and sets the border style to solid.

Border-width

- We can change that by setting up the property `box-sizing: border-box;`
- That will make our content taking exactly the size of the box we have set up before for the div. This way, even if the border is thick, the element will still have a width of 200px and a height of 300px, and the border thickness will be subtracted from these measurements.

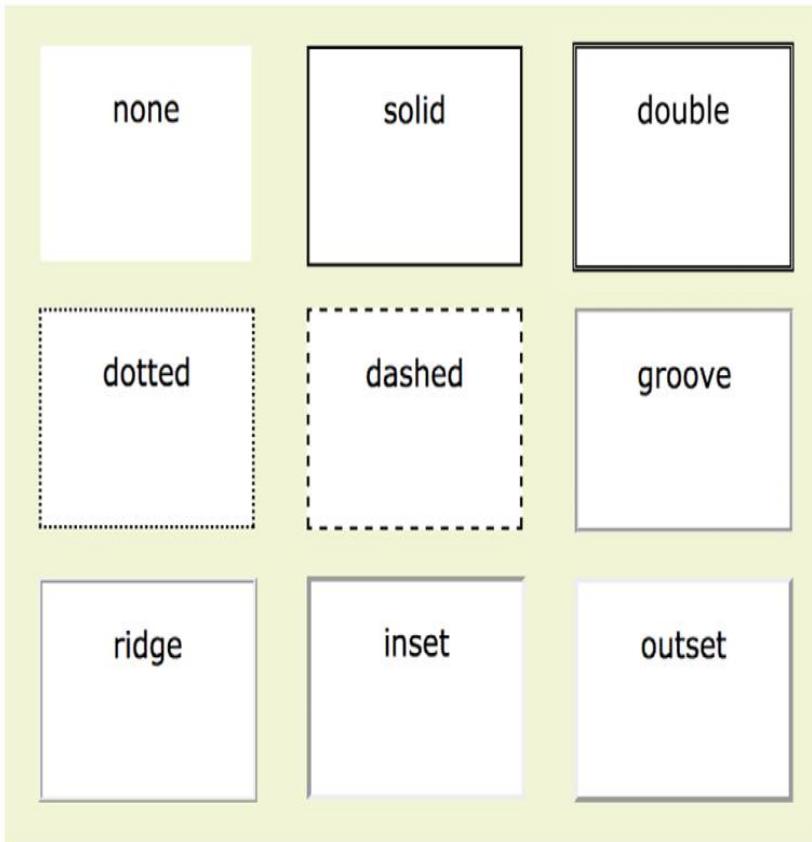
The screenshot shows a code editor with a dark theme. On the left, there is a preview window displaying a pink rectangular box with white text. The text inside reads:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

On the right, the CSS code is shown:

```
1 div {  
2     width: 200px;  
3     height: 300px;  
4 }  
5 #firstOne{  
6     border-width: 30px;  
7     border-color: magenta;  
8     border-style: solid;  
9     box-sizing: border-box;  
10 }  
11 }
```

Border-style

- Border-style will change the styling of the border.
- The values for border-style can be dotted, dashed, inset, double etc.
- We can combine style so the top and bottom look different from the left and right borders



The screenshot shows a code editor interface with three tabs: HTML, CSS, and JS. The HTML tab contains two

elements with placeholder text. The CSS tab contains a CSS rule for the first

element, setting its width and height to 200px, and applying a magenta border with a width of 5px and a dotted style to the #firstOne selector. The JS tab is empty.

```
borderCSS
Mariana

HTML
<div id="firstOne">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</div>
<div id="secondOne">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</div>

CSS
div {
  width: 200px;
  height: 200px;
}

#firstOne{
  border-width: 5px;
  border-color: magenta;
  border-style: dotted;
```

Combining border properties

- If you wish, you don't need to specify each property but combine all of them as values of the property **border**
 - The order of values is width, style and colour

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
/* width | style | color*/  
  
#firstOne{  
    border: 5px double mediumseagreen;  
}
```

Border-radius

- Another property is **border-radius**, which will style the corners of the borders.
 - In the example below, we get rounded corners by setting up the border-radius to 10px

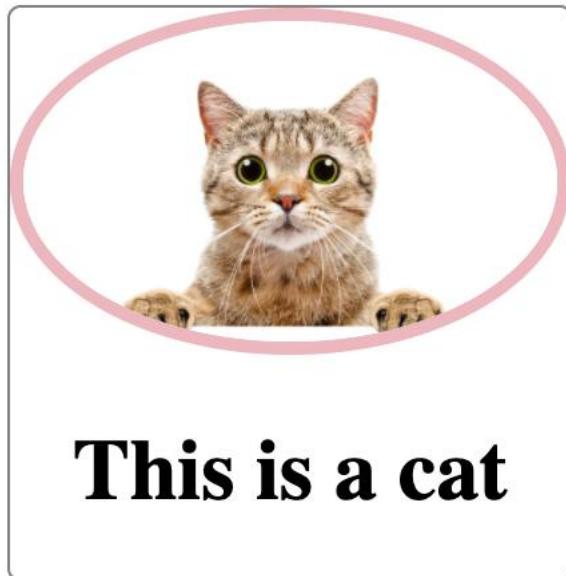
Consectetur

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

```
#firstOne{  
    border-width: 5px;  
    border-color: magenta;  
    border-style: solid;  
    box-sizing: border-box;  
    border-radius: 10px;  
}
```

Border

- [Check this markup code](#) on CodePen to see how the CSS is styling the page to result in the following image.



Padding

- Padding adds space between the content and the border
- This can be thought of as space between the content and border for a given element.
- Using the padding property puts the space on all four sides of the element. To just put padding space on one side use:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left
- It is also possible to simply use **padding: 5px;** which will give a padding of 5 px to all sides. If you want one padding at each side, set up the values as **padding: 5px 10px 15px 20px;** (the order will be top, right, bottom and left). You can also set two values such as **padding: 5px 10px;** (the order will be vertical and horizontal).

Padding

The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div id="withPadding">This content will  
2     have some padding</div>  
3 <br>  
4 <div id="withoutPadding">This content will  
5     have some padding</div>
```

The 'CSS' tab contains the following code:

```
1 #withPadding{  
2     border: 5px solid grey;  
3     width: 300px;  
4     padding: 10px 10px 10px 10px;  
5 }  
6  
7 #withoutPadding{  
8     border: 5px solid grey;  
9     width: 300px;  
10 }  
11
```

This content will have some padding

This content will have some padding

Click on the code to see the example in CodePen

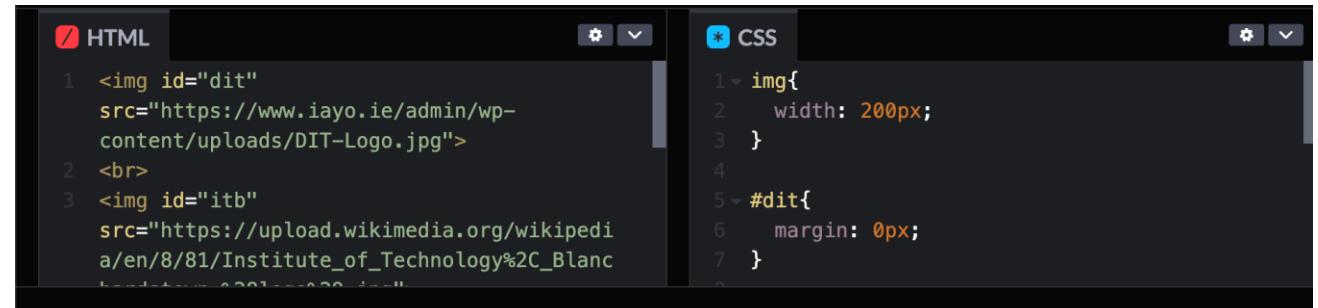


Margin

- The margin adds space outside the border of a given element.
- This can be thought of as space between the border and the other elements on the page.
- Using the margin property puts the space on all four sides of the element. To just put padding space on one side use:
 - margin-top
 - margin-right
 - margin-bottom
 - margin-left
- It is also possible to simply use **margin: 5px;** which will give a margin of 5 px to all sides. If you want one margin at each side, set up the values such as **margin: 5px 10px 15px 20px;** (the order will be top, right, bottom and left). You can also set two values such as **margin: 5px 10px;** (the order will be vertical and horizontal).

Margin

Hint: the body tag has a default margin. Therefore, even if you set up the margin of an element to zero, this element might still have some space around it. It is common practice to set the **body{margin: 0;}** before working on styling the website.



The screenshot shows the browser's developer tools with the HTML and CSS panes open. The HTML pane contains the following code:

```
1 
2 <br>
3 
```

The CSS pane contains the following code:

```
* CSS
1 img{
2   width: 200px;
3 }
5 #dit{
6   margin: 0px;
7 }
```



Click on the code to see the example in CodePen

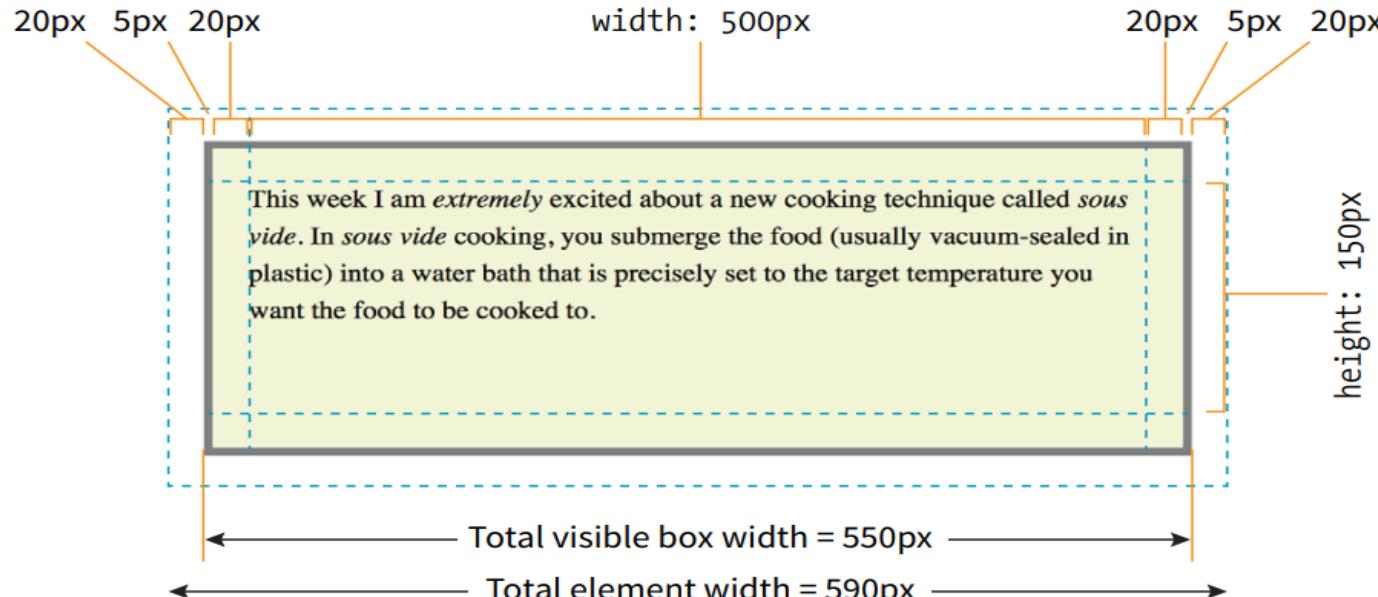
The width and height with the content-box model

```
p {  
background: #f2f5d5;  
width: 500px;  
height: 150px;  
padding: 20px;  
border: 5px solid gray;  
margin: 20px;  
}
```

Element box =

$$20\text{px} + 5\text{px} + 20\text{px} + \text{500px width} + 20\text{px} + 5\text{px} + 20\text{px} = 590 \text{ pixels}$$

content-box model



box-sizing: border-box;

Handling Overflow

When an element is sized too small for its contents, you can specify what to do with the content that doesn't fit by using the **overflow** property

Overflow

Values: visible | hidden | scroll | auto

visible

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even inside) will get scratched by the flying sand, so it has to be a good seal.

hidden

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even inside) will get scratched by the flying sand, so it has to be a good seal.

scroll

labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even

auto (short text)

Applying the masks to the glasses is the most labor-intensive part of the process.

auto (long text)

Applying the masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even

Handling Overflow

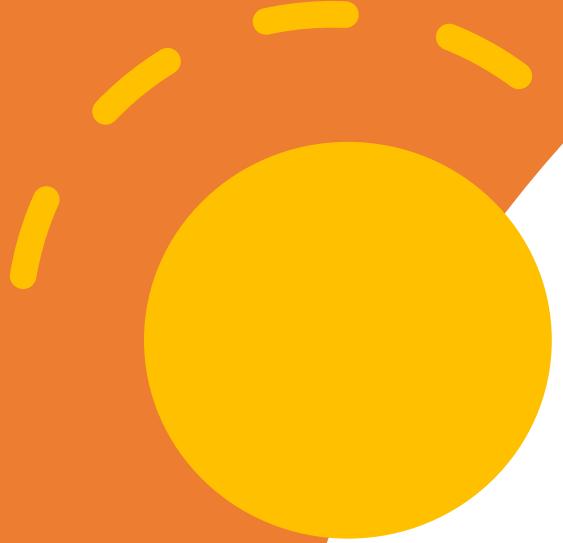
The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="box">
2   This is a long text that doesn't fit inside
3   the box because the width and height are small.
4 </div>
```

The 'CSS' tab contains the following code:

```
1 .box {
2   width: 200px;
3   height: 20px;
4   border: 2px solid blue;
5   padding: 10px;
6   overflow: visible; /* Try: hidden | scroll | auto */
7 }
8
```

This is a long text that doesn't
fit inside the box because the
width and height are small.

A yellow circular icon resembling a bomb or a bombshell, positioned on the left side of the slide. It has a yellow circle with a black outline and five short yellow lines radiating from its top edge.

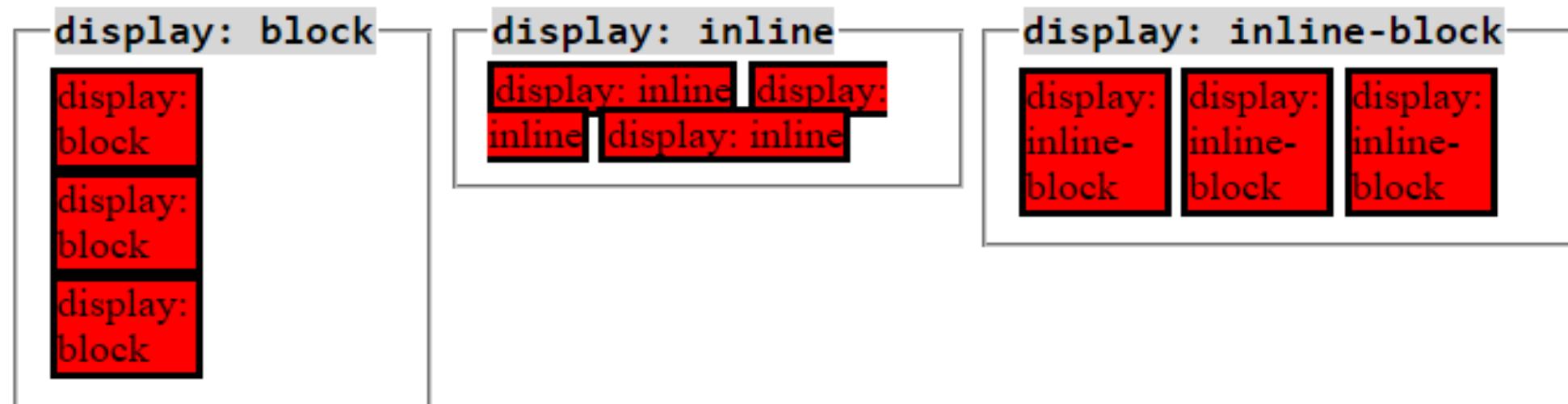
Practice

Display property

Some elements are, by default, **in-line** elements (for example, `` and `<a>`) – they go side by side without pushing other elements to the next line.

Some other elements are, by default, **block** elements (for example, `<p>` and `<h1>`) and will push other elements to the next line.

To deal with these differences, we can use the property **display** and set up how we want to distribute our elements on the page.



Display property

`display: none;` -> is commonly used with JavaScript to hide and show elements without deleting and recreating them.

`display: inline;` -> the width and height of the elements are ignored, and margin and padding push elements away **horizontally**

`display: block;` -> block breaks the flow of the document; width, height, margin and padding are respected.

`display: inline-block;` -> like an inline element, but width, height, margin and padding are respected.

Display: inline

The screenshot shows a code editor interface with three tabs: HTML, CSS, and JS. The HTML tab contains the following code:

```
1 <p class="line">This is the first  
paragraph.</p>  
2 <p class="line">This is the second  
paragraph.</p>  
3 <p class="line">This is the third  
paragraph.</p>
```

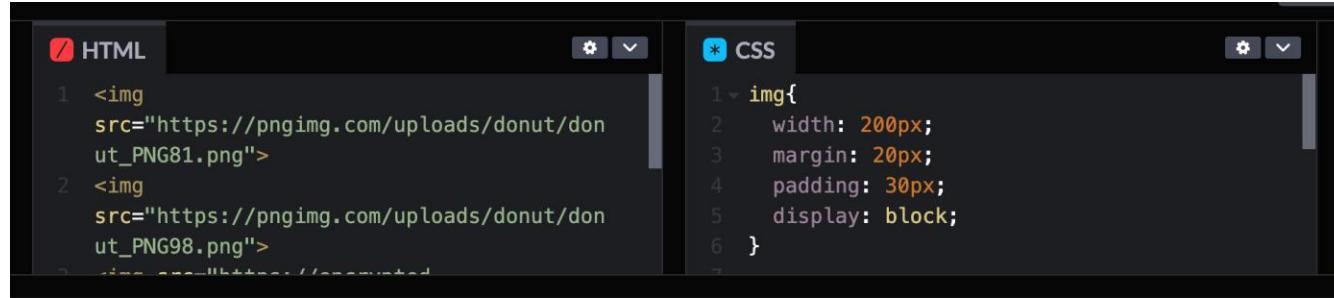
The CSS tab contains the following code:

```
* width and height are ignored*/  
9  
10 .line{  
11   display: inline;  
12 }
```

The JS tab has a single line of code: "1". Below the editor, the rendered output is displayed in three separate boxes, each containing the text "This is the first paragraph.", "This is the second paragraph.", and "This is the third paragraph." respectively. Each box is outlined with a blue dashed border.

Click on the code to see the example in CodePen

Display: block



The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```


```

The 'CSS' tab contains the following code:

```
img{
  width: 200px;
  margin: 20px;
  padding: 30px;
  display: block;
}
```



Click on the code to see the example in CodePen



Display: inline-block



```
HTML
1 <div></div>
2 <div></div>
3 <div></div>

CSS
1 div {
2   width: 300px;
3   height: 300px;
4   background-color: blue;
5   border: 5px solid black;
6   display: inline-block;
7 }

JS
1
```

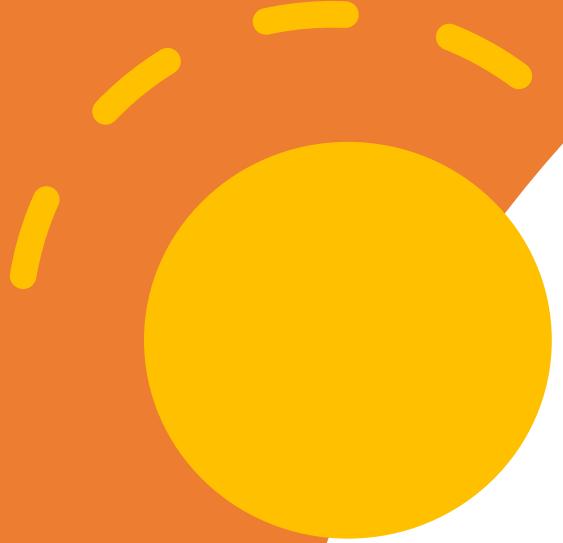
Click on the code to see the example in CodePen

Background image

- We have learned it is possible to add a background-image to an HTML element by adding the CSS statement “background-image: url(“image.png”);
- We can modify other properties to determine if we want the image to cover the full HTML element or if we want to set up the size
- We can also decide if the image is fixed or scrolled with the rest of the page elements
- We can also set if the image repeats over the HTML element
- When choosing an image as a background for your page, always choose one that complements the page and doesn't make it hard to read.

Background image - examples

- You can layer different backgrounds:
https://codepen.io/fatma_eltaher/pen/JoYrzMQ
- You can combine different properties and make a gradient background:
https://codepen.io/fatma_eltaher/pen/jEbGJxZ
- Different properties can be used to set the behaviour of the background:
https://codepen.io/fatma_eltaher/pen/gbaGEKg

A yellow circular icon resembling a bomb or a bombshell, positioned on the left side of the slide. It has a yellow circle with a black outline and five yellow dashed lines radiating from its top edge.

Practice