Programming Javascript for Web and Mobile

The Box Model

Pepe García jgarciah@faculty.ie.edu

The box model

All tags in HTML will behave like **boxes**. These boxes have some common properties that we need to be aware of.

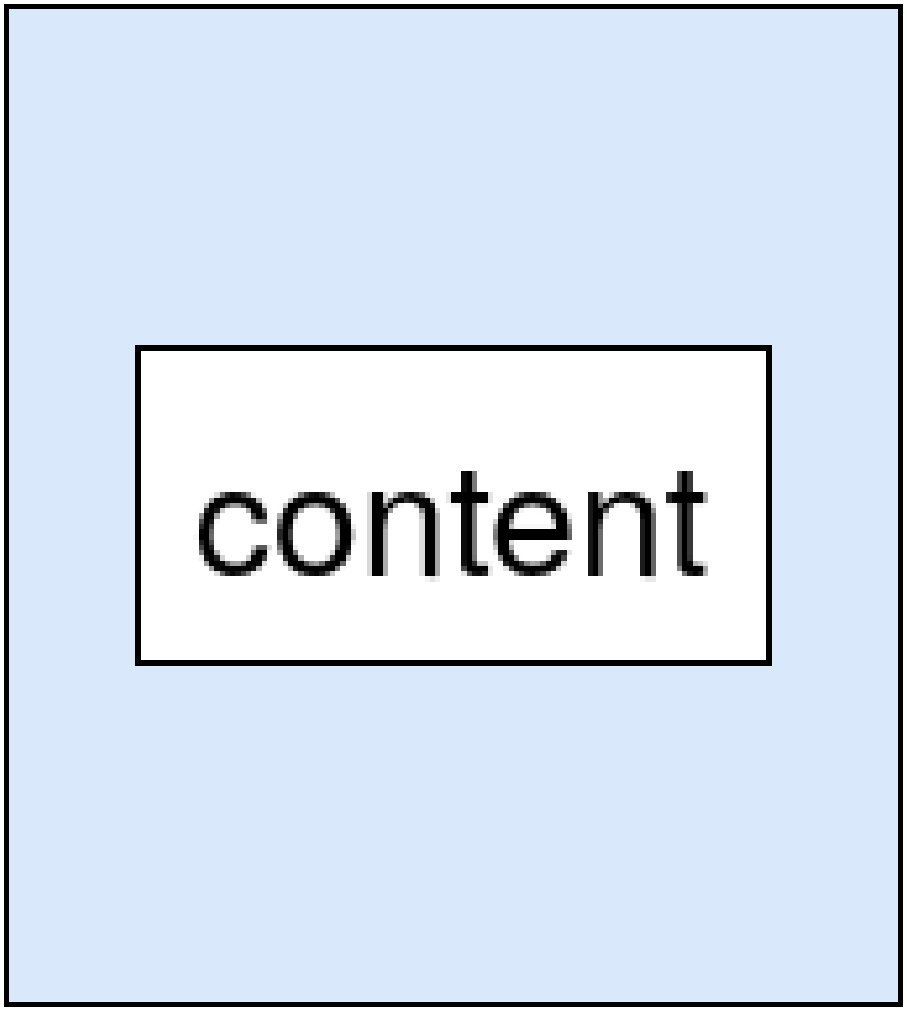
Examples in `box.html`.

The box model

- **width** and **height** apply to the content of the box

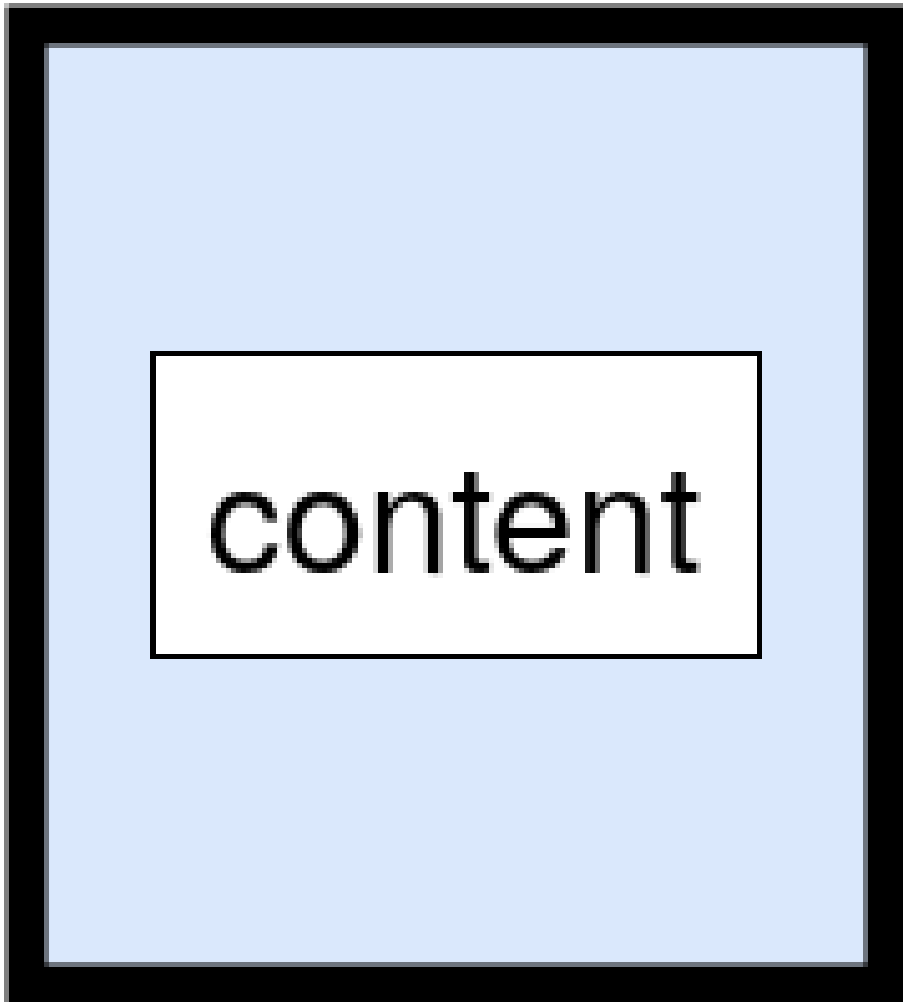
The box model

- **width** and **height** apply to the content of the box
- The **padding** is the distance from the content to the border



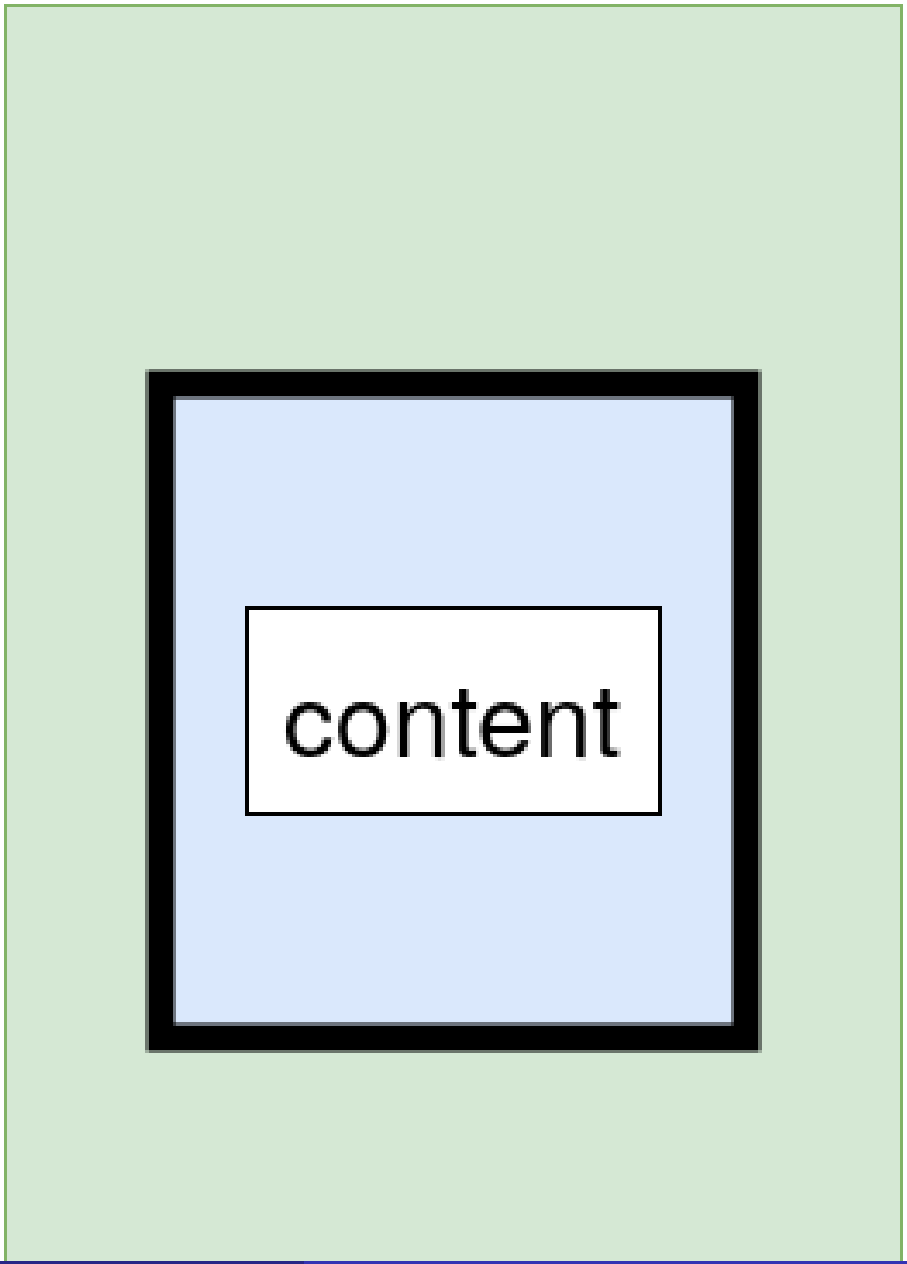
The box model

- **width** and **height** apply to the content of the box
- The **padding** is the distance from the content to the border
- The **border** sits between the margin and padding



The box model

- **width** and **height** apply to the content of the box
- The **padding** is the distance from the content to the border
- The **border** sits between the margin and padding
- The **margin** is the distance from the border to the other elements



The box model

```
background: #000;
width: 100px;
padding: 20px;
margin: 20px;
border: 5px solid orange;
```

some CSS rules that we can apply to boxes:

Display

the CSS display property describes how an element's box behaves

All HTML values have a default display property.

display - **block**

Block elements are those that start on a new line and fill their whole container, from left to right.

Some block elements are **`<div>`**, **`<li>`**

display - **block**

`display.html`

display - **inline**

Inline elements do not break the flow of text in a paragraph. Some examples of inline-by-default elements are **<a>**, and ****

display - **inline**

`display.html`

display - **inline-block**

We can create elements with `display: inline-block` that will:

- accept **`top`**, **`bottom`**, **`left`**, and **`right`**, like `display: block`.
- while allowing other elements to sit to their left and right, like `display: inline`.

display - **none**

`display:none` makes the element not to be displayed in the screen.

display - **none**

Let's clean up https://www.huffingtonpost.es

Horizontal centering

One of the first things we'll want to do when creating the layout of a webpage is centering the content horizontally.

This can be done using **margin** and **width**:

```
body {
  width: 800px;
  margin: auto;
}
```

Horizontal centering

but, what happens when we resize the window?

Horizontal centering

max-width allows us to express the maximum width we want for a box, so it resizes in case of smaller screens.

```
body {
  max-width: 800px;
  margin: auto;
}
```

Position

we use the **position** css property to make more complex layouts.

There are a number of different values for it:

Position - **relative**

`position: relative;` allows us to place the element related to where it would be placed by default.

We can move them with: **top, bottom, right,** and **left.**

Position - **fixed**

with the **`position: fixed;`** attribute we can make elements that stay in the same position. These are elements whose position doesn't change even when scrolling.

Position - **absolute**

with **`position: absolute;`** we can place elements like with **fixed** but relative to their *nearest positioned parent*

Practice

Investigate `position.html`