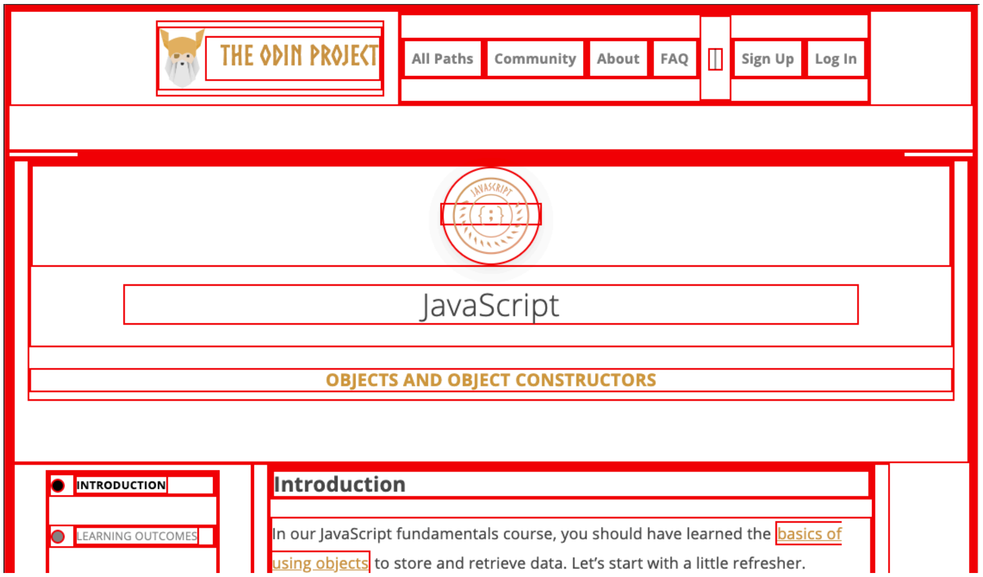
Box Model:

- Every single thing on a webpage is a rectangular box.
- These boxes can have other boxes in them and can sit alongside one another.



Introduction

In our JavaScript fundamentals course, you should have learned the basics of using objects to store and retrieve data. Let's start with a little refresher.

Can manipulate the size of these boxes with: **PADDING, BORDER, AND MARGIN**

Padding:

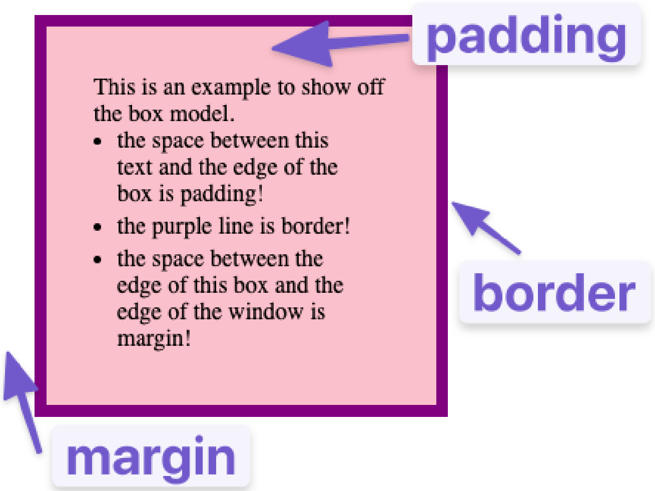
- Increases space between border and contents of box

Border:

- Adds space between margin and padding

Margin:

- Increases space between borders of box and borders of adjacent boxes



This is an example to show off the box model.

- the space between this text and the edge of the box is padding!
- the purple line is border!
- the space between the edge of this box and the edge of the window is margin!

The innermost box (actual contents!!!) can be adjusted using the **height** and **width** attributes.

In the box model, margins collapse between 2 elements next to each other, so 2 boxes next to each other with “margin:60px” will only be 60px apart, rather than 120px.

- To Note: If margins aren’t the same, the larger of the 2 margins will be used, e.g. B1: 60px, B2: 99px, margin ends up being 99px.

When using Chrome Web Tools the DEFAULT height and width of the box isn’t just what is declared for the content, but also includes: padding AND border of both top and bottom (individually, as they can have different values despite being the same ‘family’ of attributes), to create the ‘whole box’.

In a 180px x 180px with content that is 100px x 100px, this could look like:

- Padding: 10px (so 10 * 2 = 20px)
- Border: 30px (so 30 * 2 = 60px)

Margin ISN’T INCLUDED, as it isn’t a part of the element!!

If you want the height and the width of the ‘whole box’ to be the height and width normally attributed for just the content you need to include this line of code in your css class/id, etc. :

- ***box-sizing: border-box***
  - This makes the height and width account for padding and border of the element.
  - This reduces the size of the content rather than padding and border.

THIS IS EXTREMELY COMMON TO INCLUDE IN THE UNIVERSAL SELECTOR “ * { }” BECAUSE IT MAKES STYLING EASIER AS IT MAKES THE HEIGHT AND WIDTH THE ACTUAL HEIGHT AND WIDTH OF THE WHOLE ELEMENT [changes it from default of content-box].

MDN NOTES:

Boxes have inner and outer display type, and you can generally set the values for the display type using: “display: ____”.

Block:

- The box will break onto a new line.
- The `width` and `height` properties are respected.
- Padding, margin and border will cause other elements to be pushed away from the box.
- If `width` is not specified, the box will extend in the inline direction to fill the space available in its container. In most cases, the box will become as wide as its container, filling up 100% of the space available.

Some HTML elements, such as `<h1>` and `<p>`, use `block` as their outer display type by default.

Inline

- The box will not break onto a new line.
- The `width` and `height` properties will not apply.
- Top and bottom padding, margins, and borders will apply but will not cause other inline boxes to move away from the box.
- Left and right padding, margins, and borders will apply and will cause other inline boxes to move away from the box.

Some HTML elements, such as `<a>`, `<span>`, `<em>` and `<strong>` use `inline` as their outer display type by default.

Block and Inline are “Outer Display” types, affecting how the box is laid out in relation to the other boxes around it.

Boxes also have “Inner Display” types, such as flex (display: flex;). This still uses the block display for the outer, but have children of the box become flex items and behave accordingly.

- grid is another example of an inner value that boxes can have.

Margin & Padding*:

- margin-[top, bottom, left, right]
- padding-[top, bottom, left, right]

Borders*:

- border-[top, bottom, left, right]
- border-[style, color, width]
- CAN COMBINE BOTH, e.g.: `border-right-style`

* All of it applies to block boxes, some can apply to inline boxes, like those created by <span>

Inline Block

- Special value of display, providing a middle ground between inline and block
    - Use it if you don’t want an item to break onto a new line, but you do want it to respect width and height

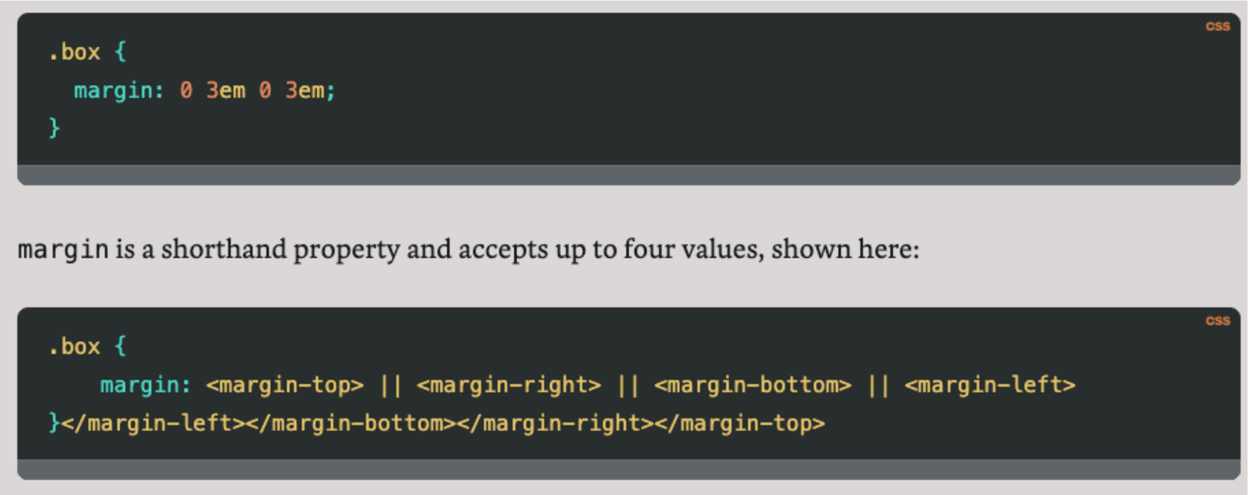
- Width and height properties are respected
- `padding`, `margin`, and `border` will cause other elements to be pushed away from the box.
- Doesn’t break onto new line and will only become larger than its content if you explicitly add width and height properties

- Helpful when you want to give a link a larger hit area by adding padding. (use with <a>)
    - Frequently used in nav bars

CSS-TRICKS NOTES

Margins:

- Set using lengths, percentages, or the keyword “auto” and can have negative values.

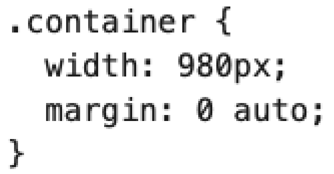


```css
.box {
  margin: 0 3em 0 3em;
}
```

margin is a shorthand property and accepts up to four values, shown here:

```css
.box {
    margin: <margin-top> || <margin-right> || <margin-bottom> || <margin-left>
}</margin-left></margin-bottom></margin-right></margin-top>
```

- If fewer than 4 values are set, the missing values are assumed based on the ones that are defined (if top and right are defined as 0px and 2px, then bottom and left are assumed to be 0px and 2px)
  - If 3 are declared it is taken as “`margin: [top] [left-and-right] [bottom];`”

- Each of the margin properties can also accept a value of auto
  - In most cases this auto will be equivalent to a value of 0 or whatever space si available on that side of the element
  - Handy for horizontal centering as seen below:
  - 

    ```css
    .container {
      width: 980px;
      margin: 0 auto;
    }
    ```

    - Centered in available space by giving element specific width and setting margins (relevantly left and right) to auto.

- AUTO CANNOT BE USED FOR CENTERING ELEMENT VERTICALLY (TOP AND BOTTOM MARGINS CAN’T BE AUTOED TO DO THIS)

- COLLAPSING MARGINS HAPPEN ONLY ON VERTICAL MARGINS!
- Negative margin will either pull the element itself in that direction, or pull other elements toward it