CSS BOX MODEL

How Are Elements Displayed?

block-level elements occupy any available width, regardless of their content and begin on a new line

inline-level elements occupy only the width their content requires and line up on the same line, one after the other

Display

Every element has a default display property value, which determines how the element is displayed

that value may be overwritten

the most common are block, inline, inline-block, and none.

For eg. the following will display as para1, para2, para3

```
p {

display: inline;

}
```

Inline-block vs block and inline

display: inline-block

Compared to display: inline, display: inline-block allows to set a width and height on the element. The top and bottom margins/paddings are respected, but with display: inline they are not.

Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

Inline-block

Inline: A value of inline will make that element an inline-level element.

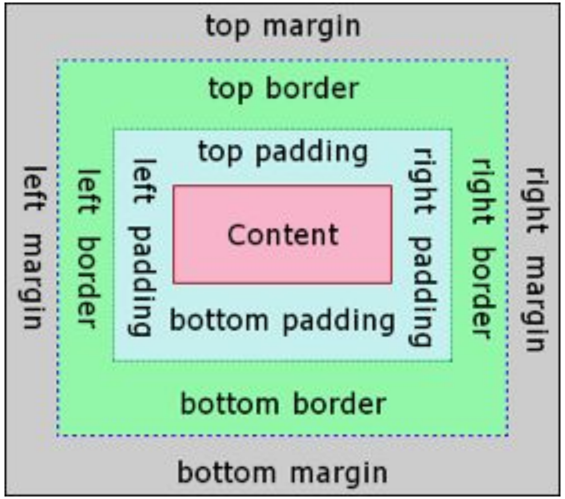
Inline-block: allow an element to behave as a block-level element, accepting all box model properties,but the element will be displayed in line with other elements, and it will not begin on a new line by default.

None: a value of none will completely hide an element and render the page as if that element doesn’t exist. Any elements nested within this element will also be hidden.

– Remember that there will be a space between inline elements

Box Model

Every element on a page is a rectangular box and may have width, height, padding, borders, and margins.



Box Model

Every element is a rectangular box, and there are several properties that determine the size of that box

width and height of an element

– may be determined by the display property, by the contents of the element, or by specified width and height properties.

padding and then border margin

Box Model

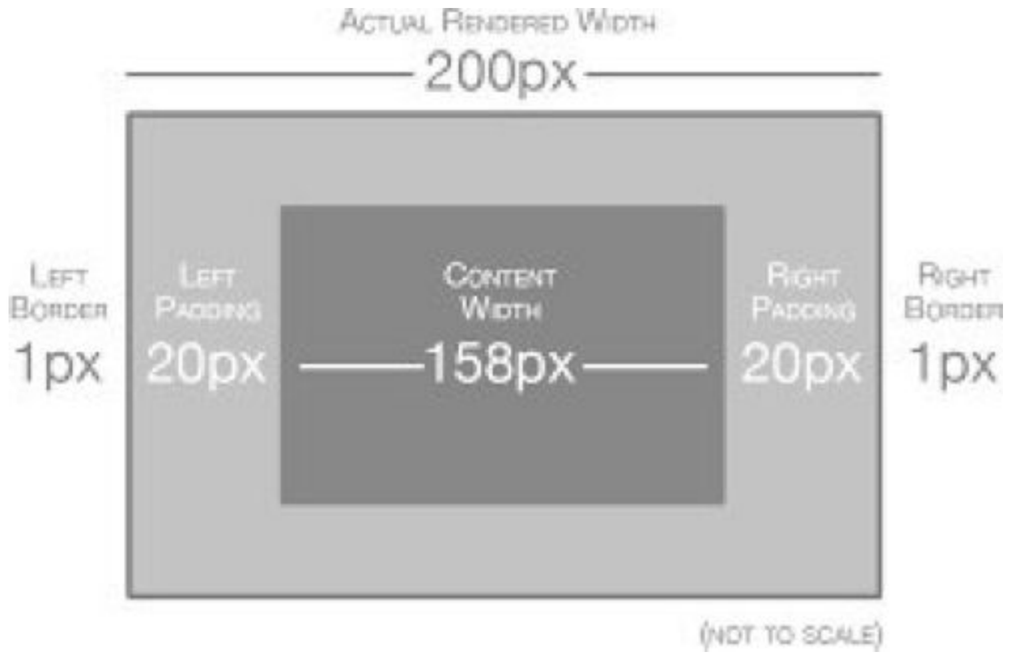
Width

margin-right + border-right + padding-right + width + padding-left + border-left + margin-left

Height

margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom

Box Model



Actual Rendered Width
200px
Left Border
1px
Left Padding
20px
Content Width
158px
Right Padding
20px
Right Border
1px
(Not to scale)

Box Model

To determine the actual size of a box we need to take into account padding, borders, and margins for all four sides of the box

Essentially box model is a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

Width & Height

Each element has default width and height

Default width of an element depends on its display value

Block-level elements have a default width of 100%

Inline and inline-block elements expand and contract horizontally to accommodate their content

Width

Inline-level elements cannot have a fixed size, thus the width and height properties are only relevant to non-inline elements

```
div {

width: 400px;

}
```

Height

Height is determined by the element content.

To set a specific height for a non-inline element, use the height property

```
div {

height: 100px;

}
```

Margin & Padding

Depending on the element, browsers may apply default margins and padding to an element

margins are outside of any border and are completely transparent

One oddity with the margin property is that vertical margins, top and bottom, are not accepted by inline-level elements, but they work for block-level and inline-block elements.

The margin and padding properties are completely transparent and do not accept any color values

Padding

Padding is similar to margin, but inside of the border rather then outside as margin

```
div {

padding: 20px;

}

div {

margin: 10px 20px;

}
```

Margin and padding

All four sides div { margin: 20px; }

Left and right div { margin: 10px 20px; }

top, right, bottom, and left div { margin: 10px 20px 10px 20px; }

Better div {

margin-top: 10px;

padding-left: 6px;

}

Borders

The border property requires three values: width, style, and color.

Common style values are solid, double, dashed, dotted, and none

```
div {

border: 6px solid black;

}
```

Borders

As with the margin and padding properties, borders can be placed on one side of an element at a time

```
div {

border-bottom: 6px solid black;

}
```

to change only bottom border width the above can be altered with:

```
div {

border-bottom-width: 12px;

}
```

Border Radius

Border-radius property, which enables us to round the corners of an element

The border-radius property accepts length units.

Assignment

Create a web page using as many of the HTML and CSS elements introduced in the course as possible. You can submit your work in Git as HTML and CSS text files.

For example: write about your home country, include a few images, a few links, a list of things you would suggest to a friend who was visiting, information about the country itself, directions to get to a famous place.

Use valid HTML and CSS throughout, taking care to ensure all tags and elements are correctly used.

Be as creative as you like, but ensure that the page is accessible and readable, and professional.

Some of you will be asked to present your page to the rest of the class next week and explain how and why you used HTML and CSS in the way that you did.