# CSS Notes (Intermediate)

## The Box Model

* Everything displayed by a CSS is a **box**.
* The behaviour of the box changes based on:
  + the display value
  + the dimensions set
  + the content inside
* **extrinsic sizing**: the size of the box is fixed
* **intrinsic sizing**: the size is determined by the browser, based on the size of the content
* Better to use intrinsic sizing by either:
  + unset the width, or
  + set width to be min-content (Other attributes includ max-content and fit-content)

### Areas of the box model



* **content**: where the content lives! This can control the size of the parent
* **padding**: surrounds the content box
* **border**: surround the padding, it’s the edge of the box
* **margin**: the space around the box

### Controlling the box model

* Every browser applies a user-agent stylesheet to HTML, defines how elements look in the absence of a style sheet. Differ between browsers.

NB **Block Level vs Inline Level** - Block level elements take up as much space as possible. Each starting a new line, also occupying as much horizontal space as possible E.g. p, ol, ul, li, headings, article, section, div - Inline elements display in a line, they do not force the text after them to a new line E.g. a, strong, em, b, i,q, mark, span

## Selectors

* E.g. a first paragraph might need to be larger than remaining paragraphs e.g.

<p>Here is the first paragraph and we want it to stand out in some way</p>  
<p>While the second and subsequent paragraphs can be made to conform to the standard styling</p>

* A **selector** can be used to find the specific element and apply a CSS rule:

article p:first-of-type {  
 color : #CC0000;  
 font-size: 1.5em;  
}

* The **universal selector**, \* can be applied to every element.
* A **type selector**, matches an HTML element e.g.

section {  
 padding : 2em;  
}

* A **class selector** matches any element with that class applied to it
* An **id selector** matches any element with that id applied to it
* An **attribute selector** matches any element with the given attribute e.g.

[data-type="primary"] {  
 color : green;  
}

<div data-type="primary"></div>

* This can be used to match a particular attribute even if no value is given.
* A **matching selector**(?) can be used to match a portion of text e.g. a link that starts “https”:

[href^='https] {  
 color : blue;  
}

* Use \* and it will match where that string occurs
* Selectors can be grouped, separated by commas before defining them.

### Pseudo-classes

* These focus on specific state e.g. when the mouse hovers over a button, or parts of an element:

/\* link changes when being hovered over \*/  
a:hover {  
 outline: 1px dotted blue;  
}  
/\* All even paras have a different background \*/  
p:nth-child(even) {  
 background-color : #e7e7e7;  
}

### Pseudo-elements

* Act a if they are inserting a new element with CSS.
* They use `::

.my-element::before {  
 content : 'Prefix -';  
}  
/\* can also use ::after \*/

* Also, e.g., highlight area selected by a user:

~~~~~{css} ::selection { background : black; color : white; }

(Read more here: <https://web.dev/learn/css/pseudo-elements>)