The 'display' property

This property has many possible values but we will only consider these values :

- block
- inline
- inline-block

It influences how elements with these display values are laid out on the page and how they react to 'width' and 'height' settings. Each element type has a specific default value for this property.

Block elements can never have other elements next to them.

We can set their width and height explicitly. If we do not set their width, they will take up all the horizontal space inside their parent.

Even if we make their width smaller than the available space, no other elements will be placed next to them.

We can set margins and paddings on all 4 sides.

Some examples of block element types are

- p
- ul, ol, li
- h1, h2, h3, etc.
- all of the grouping elements (div, nav, article, section, aside, etc).

Inline elements have an automatic width based on their content. They are placed next to each other on an invisible line until there is no more horizontal space left. The next inline element will be placed on the next line. So they are laid out just like words in a paragraph, when the end of the line is reached any further words will be placed on the next line.

Inline elements can not be given an explicit width nor height.

We can give them horizontal margin and padding, but not vertical ones.

Some examples of inline element types are

- a
- span

The vertical position relative to their line can be set using the **vertical-align** property. Have a look here to learn about that property. The interesting values are

- top, middle, bottom
- baseline (the default), i.e. the bottom of letters like a,x, o (but not g, y or p which descend below the baseline)

Inline-block elements have a combination of inline and block qualities.

- like block elements
  - we can set their width and height (just like block elements)
- like inline elements
  - they are placed next to each other until the end of the line is reached
  - their vertical position relative to their line can be changed with vertical-align

Although all elements have a default 'display' value, we can freely change it (and we will!). For example, to get the layout you want you might have to turn a <section> into an inline-block element. We will do this in the next section 'simple column layout with inline-block elements'

Some potential pitfalls when using inline-block elements

- Any whitespace in the HTML code between inline(-block) elements will be reduced to a single space.
  - to avoid this, eliminate such whitespace in your HTML
  - For example, if the <div> elements below are turned into inline-block elements with

    ```
    div {display:inline-block;}
    ```

    You should write

    ```
    <div>…</div><div>…</div>
    ```

    instead of

    ```
    <div>
      …
    </div> ← there's a newline here that will be turned into an empty space
    <div>
      …
    </div>
    ```
- The baseline is not the bottom of the line, the descender area is between them and based on the font size (even if the actual text has no letters with descenders!). If you leave vertical-align set to its default value (baseline), you might get some empty space below it that you do not want.

There is an example in 'demonstration pitfalls of inline-block elements.zip' that demonstrates these issues and lets you experiment with them.