Width and Height Properties

- Box width →

  left + right padding, left + right border, left + right margin, content width

- **width property** → sets the content width
- Box height and width determined in the same way.
- **height property** → sets the content height
- **Example:** widthHeight.html

Block/Inline Elements

- **Block Element**
  - Displayed as a block of content starting and ending with a new line.
  - Examples: <p>, , <h1>→<h6>, <table>
  - Listed one after another VERTICALLY down the page.
  - They will stretch across the whole page unless constrained in some way.
  - **Block elements** generate a **principal block box.**
- **Inline Element**
  - Does not start new lines.
  - Contained within the flow of text.
  - Appear one after another in a line, HORIZONTALLY across the page.
  - **Inline elements** generate an **inline box.**
- **Example:** blockInline.html
- **display property –** Enables you to change the type of an element.
- **Example:** changingDisplayProp.html

POSITION Property

- The **position** property along with the **top**, **right**, **bottom**, **left** properties allow us to manipulate the position of elements.
- Four possible positioning schemes
  - ***static positioning*** – default positioning where each element is laid out one after another based ("normal flow")
  - ***fixed positioning*** – the element is fixed (does not move) and it is placed with respect to the viewport (e.g., browser window).
  - ***absolute positioning*** - the element is placed in an absolute position within the containing block (positioning context).
  - **relative positioning** – the element is placed in a position relative to the position where it will appear in static positioning.

Fixed Positioning

- **Example:** fixed.html
- top/right/bottom/left → Distance from appropriate side.
- If you resize the window the text does not move and sometimes you will not be able to see it.
- Useful when you want to have an element in a fixed position of the web page (e.g., menu or back button). For example, reduce the window size for the example until you see a scrollbar and then move the text.

Relative Positioning

- **CSS Normal Document Flow –** Placing of elements one after another or within another element based on the document structure and whether the element is an inline or block element.
- **relative positioning –**
  - You can move the element relative to its normal position in normal flow.
  - Space occupied by the element in normal flow is still retained.
- **Example:** relative.html

ABSOLUTE Positioning

- **Containing Block (Positioning Context) –**
  - Element with respect you are positioning the current element.
- Default containing block is the body block.
- **absolute positioning –**
  - Current element is placed in relation to the containing block.
- The containing block is not necessarily the immediate parent block.
- Rules for determining the containing block:
  - Nearest ancestor of the element that has a position property value set to something different from static.
  - If no ancestor has a position property set then the containing block is body block.
- Setting the position property of an element to relative and not providing any offsets (e.g., top, etc.) makes the element a containing block for its descendants.
- **Example:** absolute.html

float Property

- float property
  - A box is laid out according to normal document flow.
  - The box is then taken out of the flow and shifted to the left or right as far as possible.
- Values:
  - right – floats box to the right and content to the left.
  - left – floats box to the left and content to the right.
- **Example:** float.html, float.css (floating inline element img)
- The float property used instead of align attribute in img.
- The float property can be used with elements other than images.
- Provides an alternative to table-based layout.
- Float property and block elements
  - **Example:** floatBlock.html, floatBlock.css