CSS - Box Model and Display

By CODEMIND Technology

Contact us - 966 50 44 698
966 50 44 598

CSS Font Family

Generic Font Families

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy

Font Size

font-style: normal | italic

font-size: small | smaller | medium | large | larger

font-weight: bold | bolder | normal

④ Common CSS properties (by group)

TEXT:

color
font
font-family
font-size
font-weight
letter-spacing
line-height
text-align
text-decoration
text-indent
text-transform
vertical-align

LIST:

list-style
list-style-image
list-style-position
list-style-type

BACKGROUND:

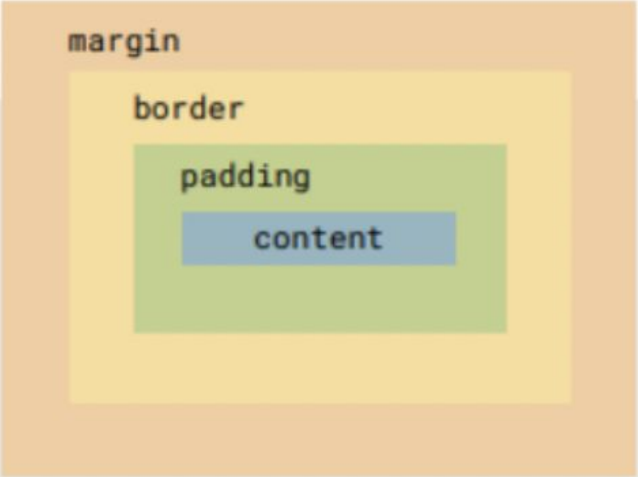
background
background-attachment
background-color
background-image
background-position
background-repeat

DISPLAY:

display
float
clear
overflow
visibility

OTHER:

cursor



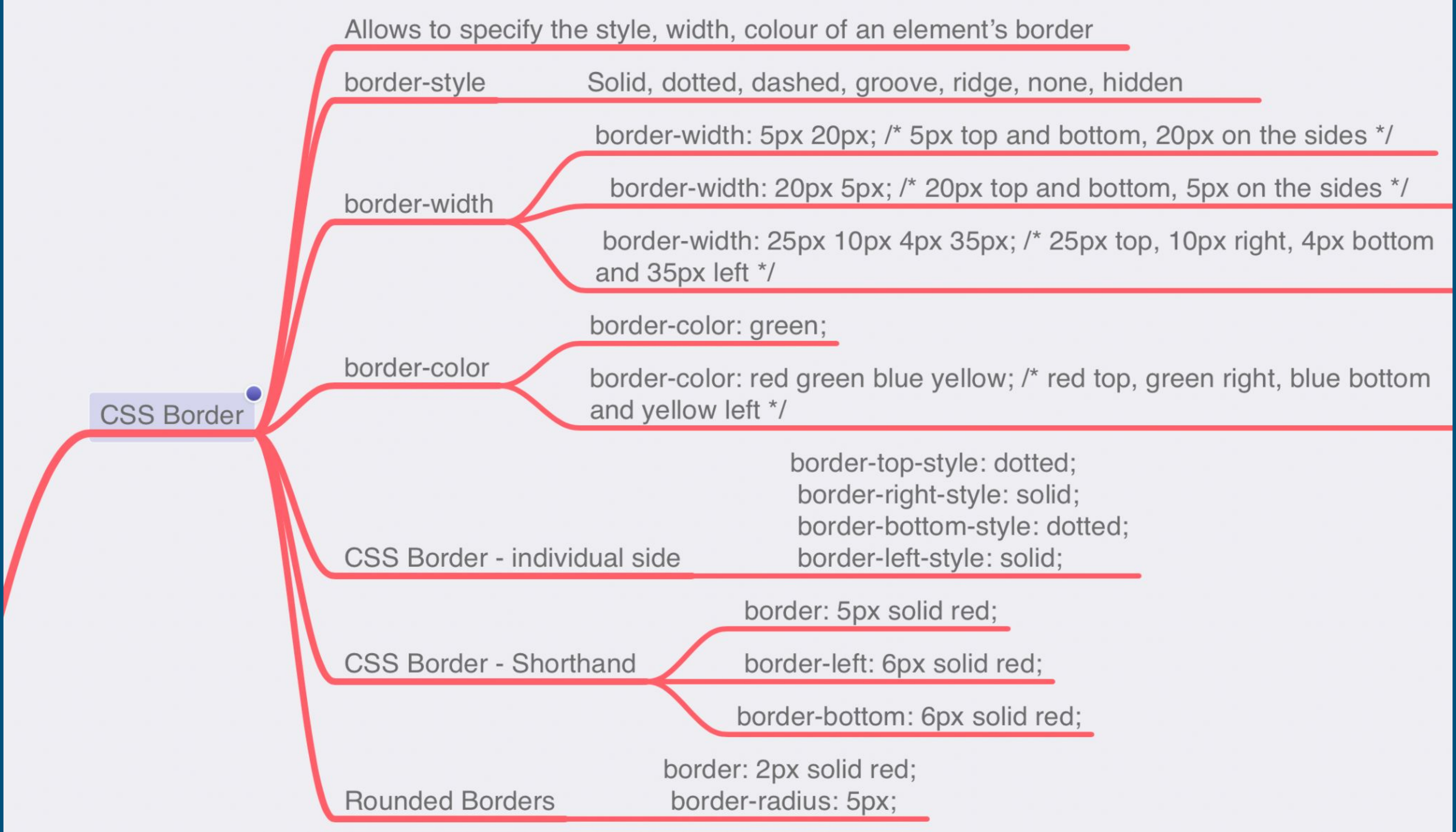
BOX:

border
border-color
border-style
border-width
height
margin
padding
width
box-sizing

POSITION:

position
top
bottom
left
right
z-index

CSS Borders



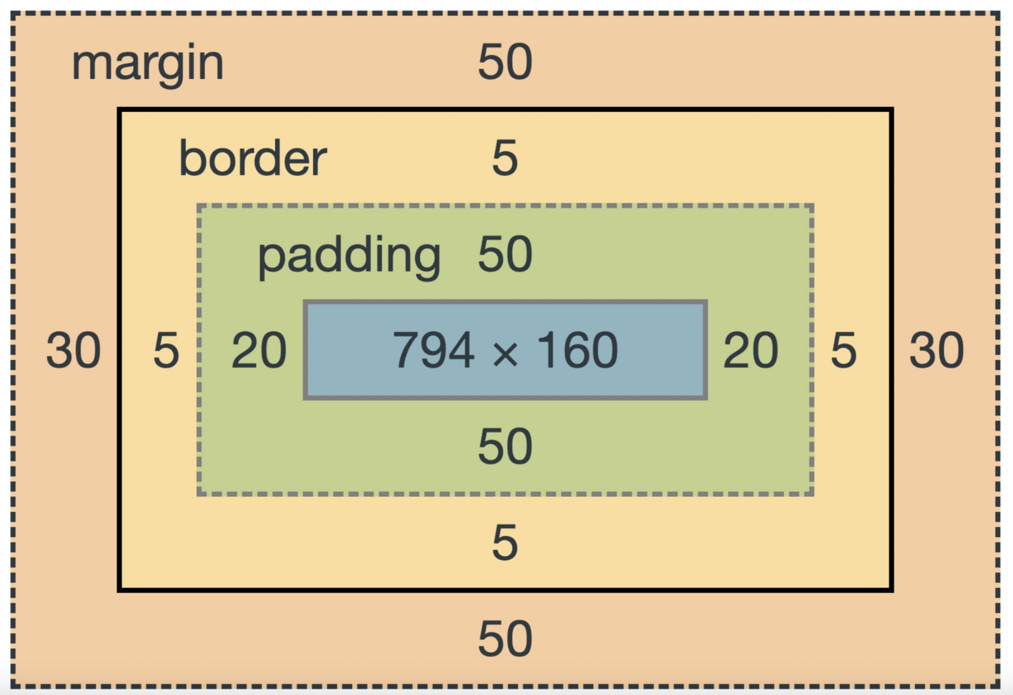
CSS Border

- Allows to specify the style, width, colour of an element’s border
- border-style
  - Solid, dotted, dashed, groove, ridge, none, hidden
- border-width
  - border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */
  - border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */
  - border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */
- border-color
  - border-color: green;
  - border-color: red green blue yellow; /* red top, green right, blue bottom and yellow left */
- CSS Border - individual side
  - border-top-style: dotted;
border-right-style: solid;
border-bottom-style: dotted;
border-left-style: solid;
- CSS Border - Shorthand
  - border: 5px solid red;
  - border-left: 6px solid red;
  - border-bottom: 6px solid red;
- Rounded Borders
  - border: 2px solid red;
border-radius: 5px;

CSS Box Model

CSS box model consists of

- Margins
- Borders
- Paddings and
- The actual content



IMP Property: box-sizing: border-box

This property set the how the total width and height of an element is calculated

Note: Default value for this property is content-box

CSS Margins

- Used to create space around elements, outside of any defined borders

CSS Padding

- Used to create space around an elements content, inside of any defined borders

CSS Margins

- Used to create space around elements, outside of any defined borders
- Margins - Individual sides
  - margin-top
  - margin-right
  - margin-bottom
  - Margin-left
  - Margin:auto → Horizontally center the element within its container
- Margin -Shorthand Property:
  - margin:20px 30px 15px 25px → top-20px, right-30px, bottom-15px, left-25px
  - margin:10px 20px 30px → top-10px, right and left - 20px, bottom-30px
  - margin:10px 15px → top and bottom 10px, left and right 15px
  - Margin: 5px → All four sides 5px each

CSS Margin Collapse

- Sometimes two margins collapse into a single margin
- Top and Bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.
- Note: This does not happen on left and right margins. Only top and bottom margins
- Example:

CSS Padding

- Used to create space around an elements content, inside of any defined borders
- Margins - Individual sides
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left
- Margin -Shorthand Property:
  - padding:20px 30px 15px 25px
  - padding:10px 20px 30px → top-10px, right and left - 20px, bottom-30px
  - padding:10px 15px → top and bottom 10px, left and right 15px
  - padding: 5px → All four sides 5px each

CSS Height and Width

- height and width properties are used to set the height and width of an element

- max-width property is used to set the maximum width of an element

CSS Height and Width Values:

- auto: This is default. The browser calculates the height and width
- length: Defined the height and width in px, cm
- %: Defines the height/width in the percentage of the containing block
- initial: Sets the height/width to its default value
- inherit: The height/width will be inherited from it’s parent

The ‘display’ Property

- Specifies how an element is displayed.

- Note: Every element has a default display value depending on what type of element it is.

- none
- block
- inline
- inline-block

Chrome Extension - Pesticide

→ Install and pin it on browser so that whenever we want to see the our html developed page element’s outline, we can click on it and see.

Shortcut to get random text → lorem5, loremN

Example of **block** elements

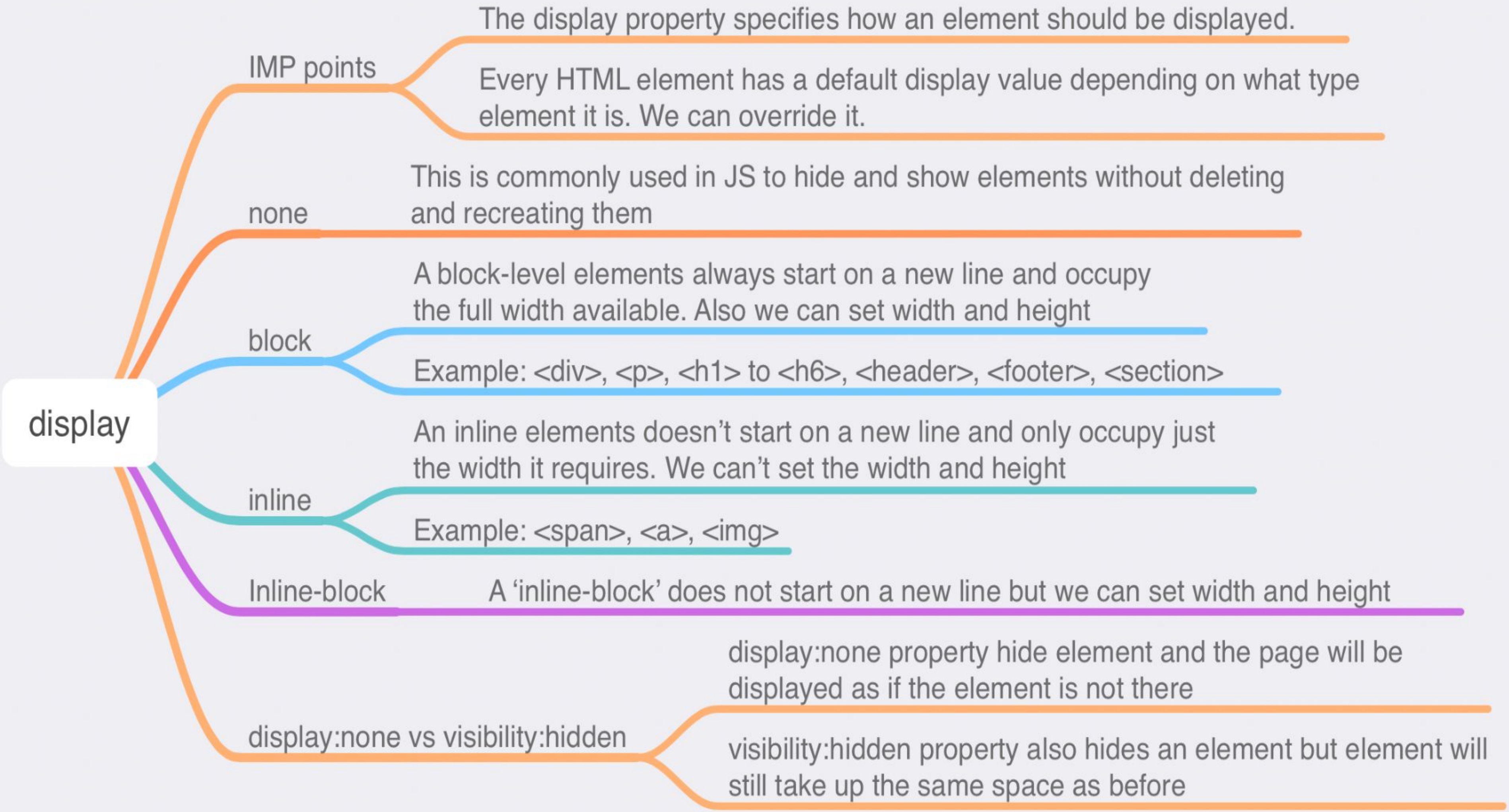
- Paragraph → <p>
- Heading → <h1> to <h6>
- List → <ol>, <ul> and <li>
- Table → <table> and <tr>
- Forms → <form>
- Semantic Tags → <nav>, <header>, <footer>

Example of **inline** elements

- Text Formatting tags → <strong>, <i>, <b>, <sup>, <sub>, <u>
- span → <span>
- Images → <img>
- Anchor → <a>
- Form → <label>, <input>, <button>, <textarea>
- Media Tags → <video>, <audio>

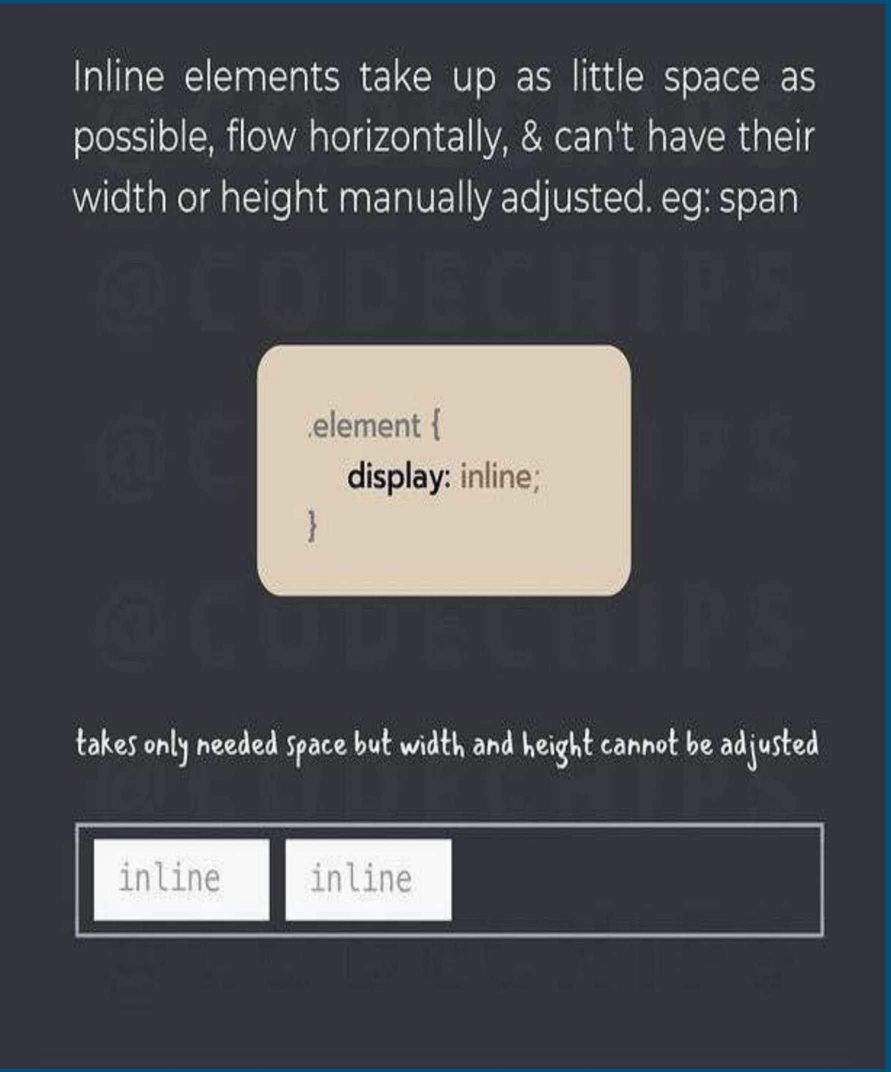
Example of **inline-block** elements

-



display

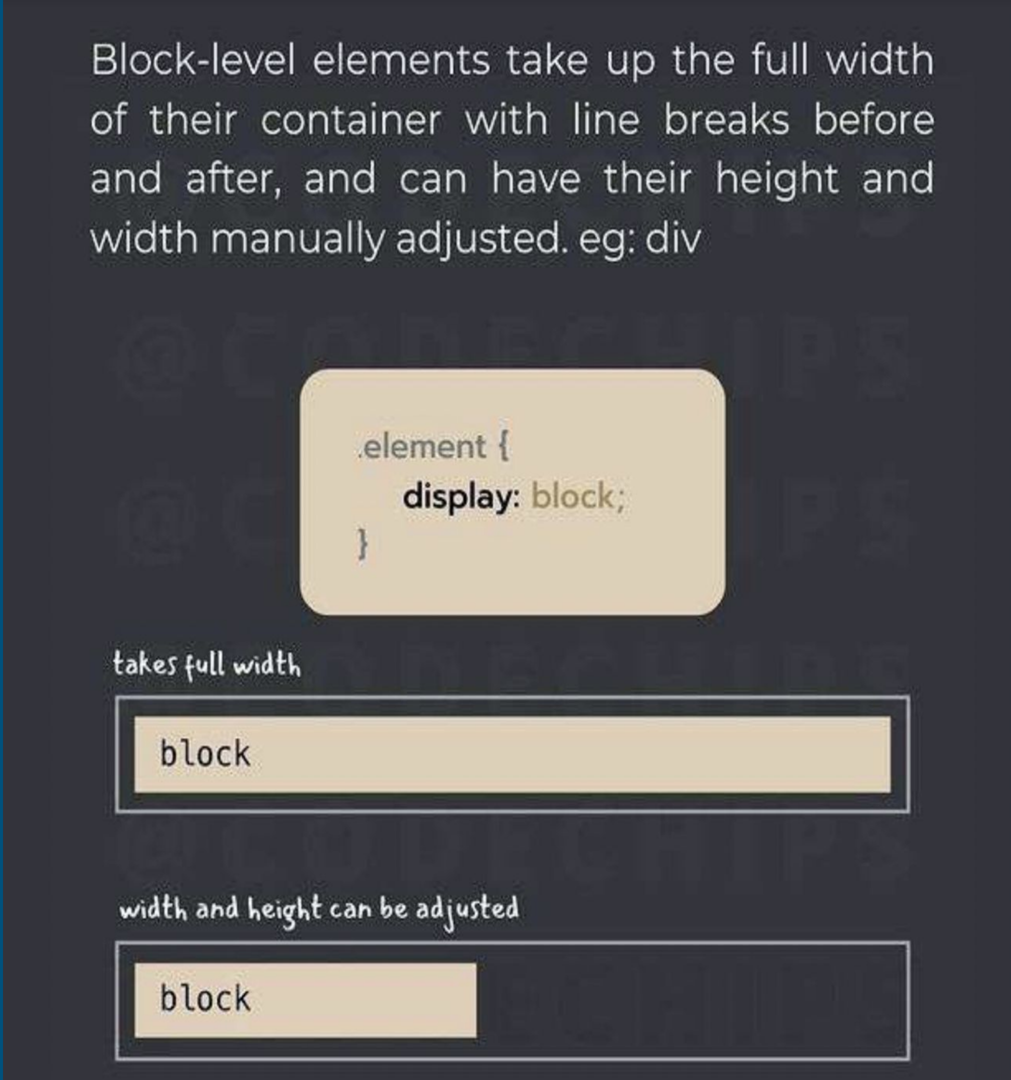
- IMP points
  - The display property specifies how an element should be displayed.
  - Every HTML element has a default display value depending on what type element it is. We can override it.
- none
  - This is commonly used in JS to hide and show elements without deleting and recreating them
- block
  - A block-level elements always start on a new line and occupy the full width available. Also we can set width and height
  - Example: <div>, <p>, <h1> to <h6>, <header>, <footer>, <section>
- inline
  - An inline elements doesn’t start on a new line and only occupy just the width it requires. We can’t set the width and height
  - Example: <span>, <a>, <img>
- Inline-block
  - A ‘inline-block’ does not start on a new line but we can set width and height
- display:none vs visibility:hidden
  - display:none property hide element and the page will be displayed as if the element is not there
  - visibility:hidden property also hides an element but element will still take up the same space as before



Inline elements take up as little space as possible, flow horizontally, & can't have their width or height manually adjusted. eg: span

```
.element {
    display: inline;
}
```

takes only needed space but width and height cannot be adjusted

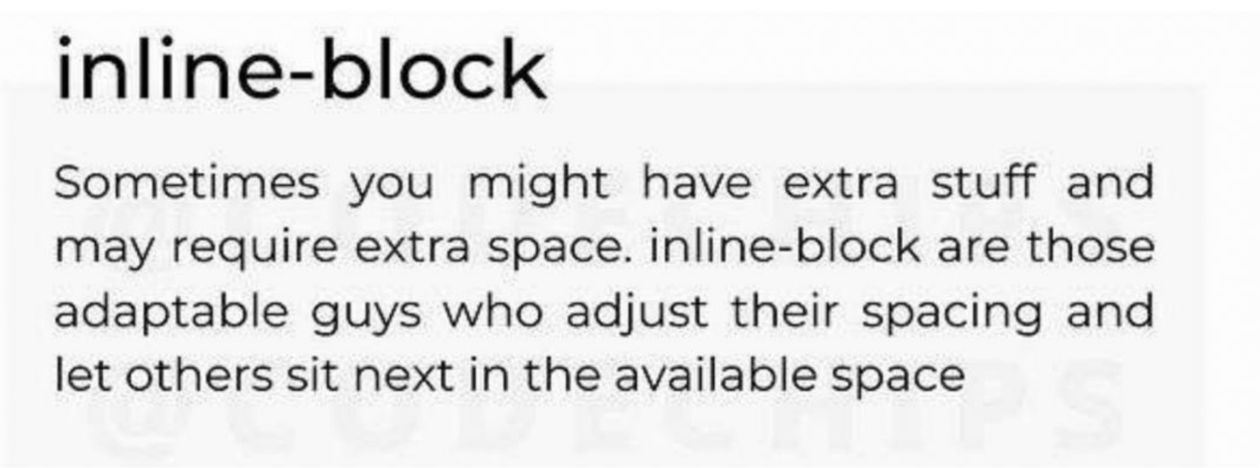


Block-level elements take up the full width of their container with line breaks before and after, and can have their height and width manually adjusted. eg: div

```
.element {
    display: block;
}
```

takes full width

width and height can be adjusted

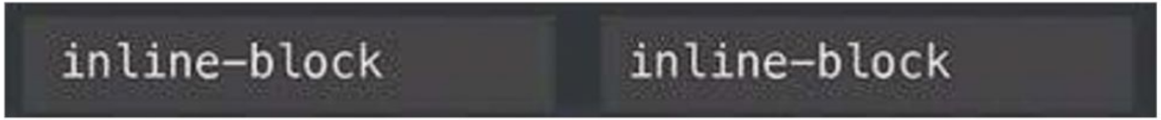


inline-block

Sometimes you might have extra stuff and may require extra space. inline-block are those adaptable guys who adjust their spacing and let others sit next in the available space

```
.element {
   display: inline-block;
}
```

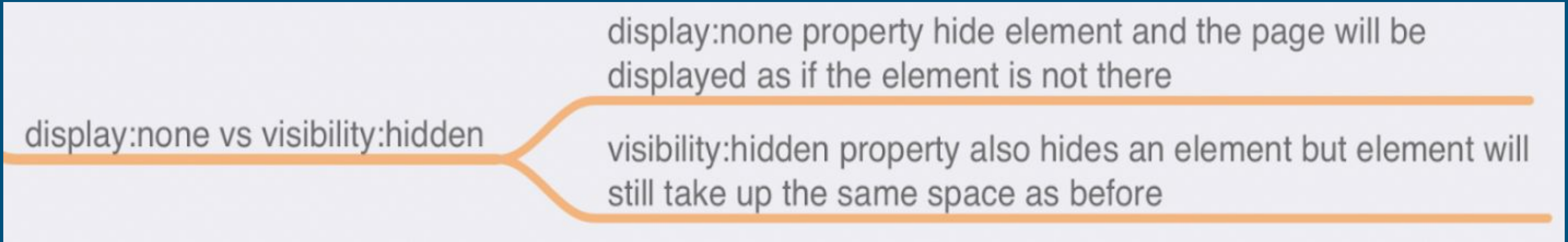
Note: These elements appears next to each other and it's allowed to set the width and height



The visibility property

visibility:hidden

display:none



display:none vs visibility:hidden

display:none property hide element and the page will be displayed as if the element is not there

visibility:hidden property also hides an element but element will still take up the same space as before