

Grouping and Box Model

Element structure

Shadi Lahham - Programmazione web - Frontend - HTML e CSS

Review - Anchors

Element: Anchor

Linking to anchors is very similar to normal links:

- Normal links always point to the top of a page
- Anchors point to a place within a page

To link to an anchor:

1. Create the anchor
2. Create a link pointing to the anchor

1.Anchor

- tag: `<a>`
- Name attribute: `"chapter4"`

2. Link to the anchor

- tag: `<a>`
- href attribute starts with #: `"#chapter4"`
- title attribute: `"The 4th Chapter"`

Element: Anchor - syntax

Anchor

```
<a name="chapter4"></a>
```

Link to the anchor

```
<a href="#chapter4" title="The 4th Chapter">Go to chapter 4</a>
```

Link an anchor in an external page

```
<a href="https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Selectors">MDN CSS selectors</a>
```

Link to the top of the page

```
<a href="#top">Back to top of page</a>
```

Note:

`href="#top"` or the empty fragment (`href="#"`) can be used to link to the top of the current page

Review - Inline vs Block

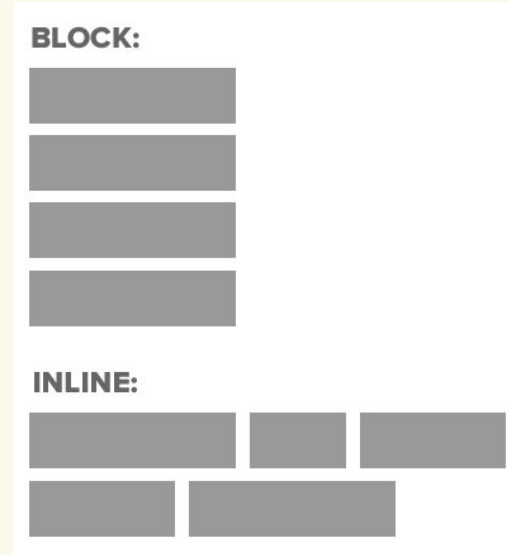
Inline vs Block

Block elements.

Appear on the next line, like paragraphs

Inline elements.

Appear on the same line that they are written on



Inline vs Block

- CSS divides HTML into two types: inline and block.
- After block elements, browsers render a new line.
- Example of inline elements:
 - **img, a, br, em, strong**
- Example of block elements:
 - **p, h1, ul, li, almost everything else**

Element: div

- Block level element. Each new div is rendered on a new line
- A division, or section of content within an HTML page
- Used to group elements to format them with CSS or manipulate with JS
- Apply IDs and Classes to divs to control their styles with CSS

Grouping elements - styling example

```
.special-quote {  
  text-align: right;  
  color: purple;  
  text-decoration: underline;  
}
```

```
<div class="special-quote">  
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit</p>  
  <p>Sed do eiusmod tempor incididunt ut labore et dolore.</p>  
</div>  
<p>Magna aliqua. Ut enim ad minim veniam.</p>  
<p>Quis nostrud exercitation ullamco.</p>
```

Note: Notice the new CSS properties used above, [text-decoration](#) and [text-align](#)

Element: span

- Inline element
- Each new span is rendered next to each other & only wraps when it reaches the edge of the containing element
- Can be used to apply styles to text inline so as not to break the flow of content

Styling text with spans - example

```
.highlight {  
  background-color: darkorange;  
  cursor: pointer;  
}
```

```
.highlight:hover {  
  background-color: yellow;  
}
```

```
.special {  
  color: darkorange;  
}
```

<p>Paragraph with highlighted text and special text.</p>

Note: Notice the new CSS property used above, [cursor](#)

Review - width and height

Property: Width

- Sets the width of a block-level element or `img`
- Doesn't work for inline elements (unless their `display` property is changed)
- Accepts a variety of length units

```
#sidebar {  
  width: 200px;  
  width: 20em; /* relative to font size */  
  width: 20%; /* relative to containing element width */  
  width: 20vw; /* relative to window: 1vw = 1% window width */  
}
```

A list of all CSS length units

[The Lengths of CSS](#)

[CSS Units](#)

The most used are: `px`, `rem`, `em`, `vw`, `vh`, `%` (percentage)

Property: Height

- Works like width, with all the same units

```
p.alert {  
  height: 50px;  
  height: 5em; /* relative to font size */  
  height: 10%; /* containing element MUST have specified height */  
  height: 10vh; /* relative to window: 1vh = 1% window height */  
}
```

Min max properties

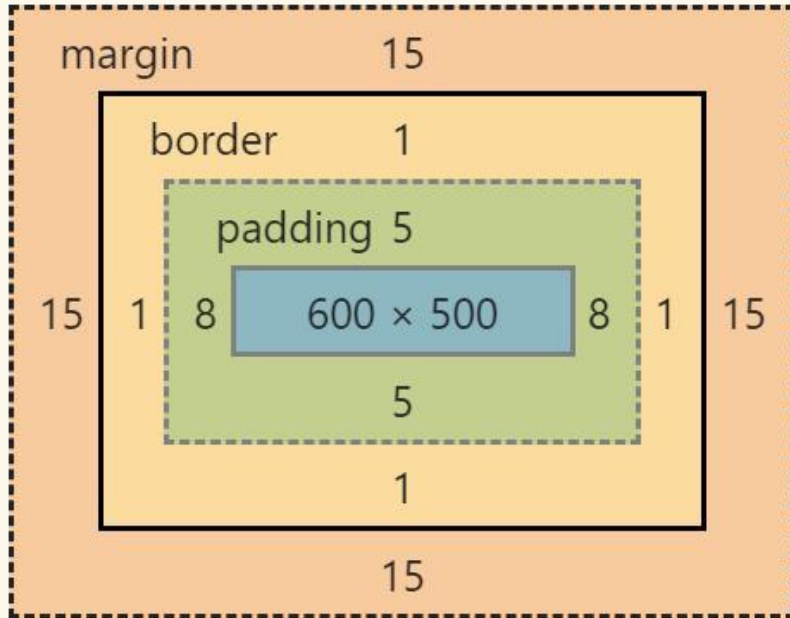
- Set upper or lower limits to the size of elements
- An element cannot be smaller than its min-width or min-height
- An element cannot be larger than its max-width or max-height

```
img {  
  max-width: 100%; /* may be no wider than the containing element */  
}  
  
#sidebar {  
  width: 30%; /* will be 30% of the width of the containing element */  
  min-width: 200px; /* but will stop shrinking with its parent at 200px */  
}
```

The box model

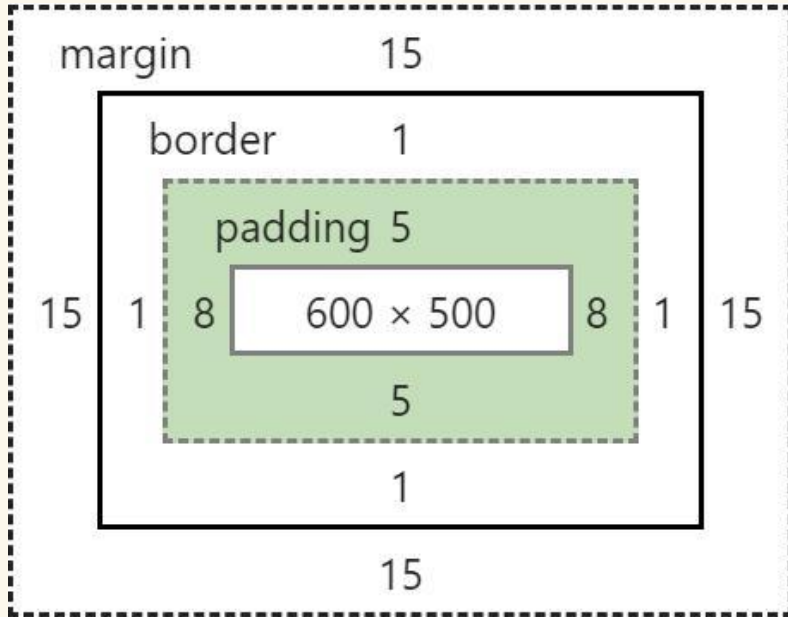
The box model: overview

Each element can have: padding, border, margin



Padding

Space between the border and the content



Padding values

It's possible to specifying the padding for each side of the element

```
.parent {  
  padding-top: 15px;  
  padding-right: 10%;  
  padding-bottom: 10em;  
  padding-left: 50vw;  
}
```

Note:

Background properties (such as background-color) apply to padding as well as content

Padding Shorthand

Four values (TRBL)

`padding: top right bottom left;`

Three values (T/RL/B)

`padding: top right/left bottom;`

Two values (TB/RL)

`padding: top/bottom right/left;`

One value

`padding: all sides;`

Note:

Padding adds to the total size of the box, unless `box-sizing: border-box;` is used

Padding examples

15 pixels on all sides

```
padding: 15px;
```

10 pixels on top only

```
padding-top: 10px;
```

10 on top, 5 on right, 3 on bottom, 5 on left

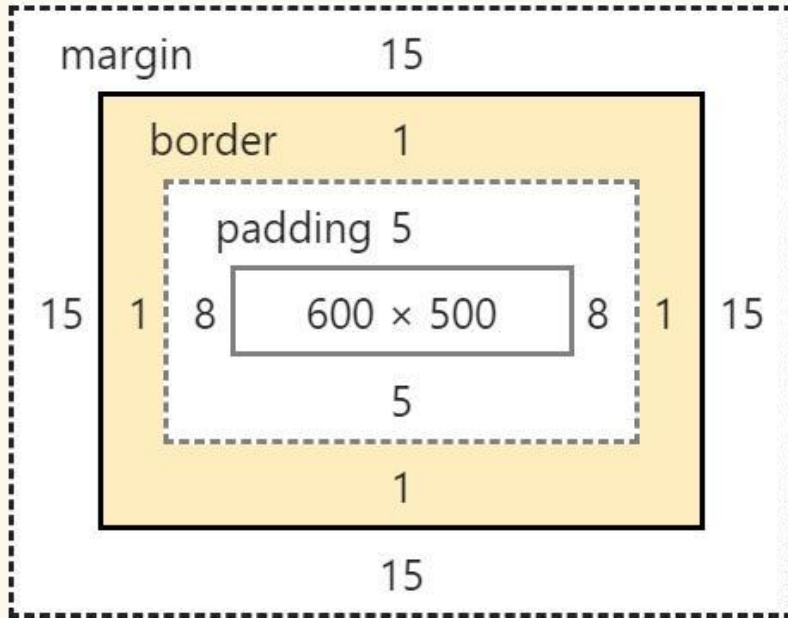
```
padding: 10px 5px 3px 5px;
```

Reference

[CSS Padding](#)

Border

The edge around the box



Border shorthand

Borders are specified as "thickness, style, color"

A solid red border

```
border: 1px solid red;
```

A thick dotted black top border

```
border-top: 4px dotted #000000;
```

Two different border styles

```
border-top: 1px solid #ff0000;
```

```
border-bottom: 4px dotted #00ff00;
```

Note:

Like padding, border adds to the total size of the box, unless `box-sizing: border-box;` is used

Border - separate properties

You can specify each property separately, or all three together

```
border-width: 10px;  
border-style: dashed;  
border-color: #666666;
```

OR

```
border: 10px dashed #666666;
```

Rounded Borders

```
border-radius: 5px;
```

Reference

[CSS Borders](#)

Property: box-sizing

The **box-sizing** property (introduced in CSS3)

Defines whether the width and height of an element should include padding and borders or not

content-box (sadly the default)

Width and height properties (and min/max) include only the content

Border and padding are not included

Border-box (most used)

Width and height properties (and min/max) include the content, padding and border

```
.example {  
  box-sizing: border-box;  
  width: 100%;  
  border: solid #5b6dcd 10px;  
  padding: 5px;  
}
```

Property: box-sizing Reset

Old method

```
* {  
  box-sizing: border-box;  
}
```

Universal Box Sizing

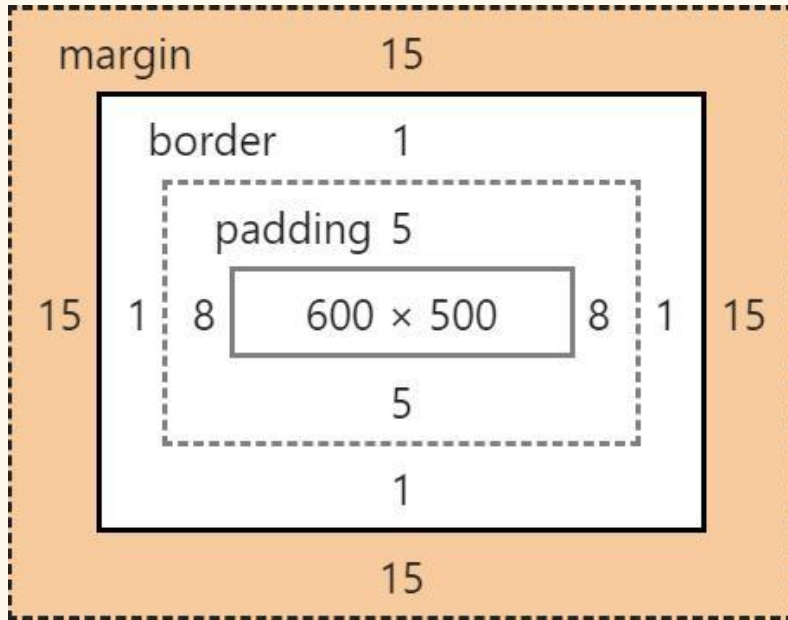
```
*, *::before, *::after {  
  box-sizing: border-box;  
}
```

Universal Box Sizing with Inheritance

```
html {  
  box-sizing: border-box;  
}  
*, *::before, *::after {  
  box-sizing: inherit;  
}
```

Margin

Transparent area around the box that separates it from other elements



Margin shorthand

15 pixels on all sides

```
margin: 15px;
```

10 on top, 5 on right, 3 on bottom, 5 on left

```
margin: 10px 5px 3px 5px;
```

10 pixels on top

```
margin-top: 10px;
```

Note:

Margins use the same syntax as padding

Reference

[CSS Margin](#)

Margin - Auto Margin

If a margin is set to auto on a box that has width it will take up as much space as possible

Centered:

```
margin: 0 auto;  
width: 300px;
```

Align box to the right:

```
margin-left: auto;  
margin-right: 5px;  
width: 300px;
```

Note:

To align text use [CSS text-align property](#)

Margin - Collapsing Margins

Top and bottom margins of elements are collapsed into a single margin that is equal to the largest of the two margins

```
.block-type-a {  
  height: 100px;  
  margin-bottom: 25px;  
  background-color: #c0ca33;  
}
```

```
.block-type-b {  
  height: 100px;  
  margin-top: 50px;  
  background-color: #fb8c00;  
}
```

```
<div class="block-type-a"></div>  
<div class="block-type-b"></div>
```

Your turn

1.CSS playground

- Spans
 - Wrap several words or phrases with span tags
 - Give them a class or id
 - Style them using different CSS rules; padding, border, margins, colors, backgrounds, fonts, text styles, etc
- Divs and semantic HTML5 elements
 - Use divs to create containers for your content
 - Divide the page into into different sections: a header, content area, sidebar, and a footer
 - Give them a class or id
 - Style them using different CSS rules; padding, border, margins, colors, backgrounds, fonts, content alignment, round corners, etc
- Images
 - Add a width & height to some img & div elements
 - Use CSS selectors to target specific images in div containers
 - Set a rule for all img tags so they can never outsize their parents

2.Styled news page

- Use the following HTML5 elements to create the structure of a news page:
 - `<header><nav><main><section><article><aside><footer>`
- Create several sections each containing several articles
- Create anchors in the nav element that link to the sections below
- Add borders, padding, margins to create the page structure
- Add content (news text and images) to flesh out the page
- Style the page
 - Use a different background for each HTML5 element
 - One section should be the 'headline-news' and styled differently from the other sections
 - Use font colors and sizes to further distinguish sections from each other
 - The page be centered in the browser window (not full width)

Bonus:

- Use an `<aside>` element
 - Position the aside element to one side of the page using float (see next lesson)

References

HTML elements

[The Anchor element](#)

CSS properties

[Text-decoration](#)

[Text-align](#)

[Cursor](#)

References

Box model

[CSS Padding](#)

[CSS Borders](#)

[CSS Margin](#)

References

Box model

[Mastering margin collapsing](#)

[Box-sizing - MDN reference](#)

["Inheriting box-sizing Probably Slightly Better Best-Practice"](#)

[CSS Tricks: Box Sizing](#)

[Learn About CSS box-sizing: CSS Border Box Explained](#)