

Web Developer

HTML, CSS e Strumenti di Digital Marketing
(SEO, SEM, SEA)

Docente: Shadi Lahham

Grouping & Box Model

Element structure

Shadi Lahham - Web development

Grouping & separating

div & span

Block vs inline

Block vs inline elements

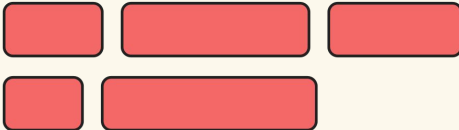
Block elements

Appear on the next line, like paragraphs



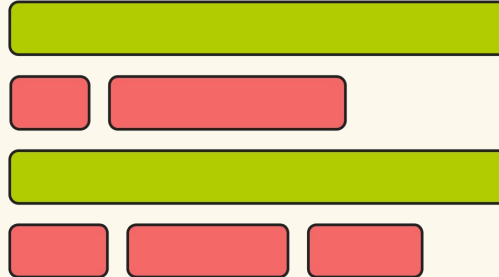
Inline elements

Appear on the same line that they are written on



Block and inline elements

Behave based on their default display value



Grouping for styling div

```
.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 CSS properties used above, [text-decoration](#) and [text-align](#)

Styling pieces of text with spans

```
.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 CSS property used above, [cursor](#)

CSS properties

width & 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 viewport: 1vw = 1% viewport width */  
}
```

A list of all CSS length units

[The Lengths of CSS](#)

[CSS Units](#)

[CSS values and units | MDN](#)

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 viewport: 1vh = 1% viewport 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 */  
}
```

[max-width](#) | [MDN](#)

Calc

- Property values can be determined by calculation using the CSS `calc()` function
- Different units like percentages, pixels, ems, and more can be combined
- Responsive design and dynamic layout adjustments are facilitated

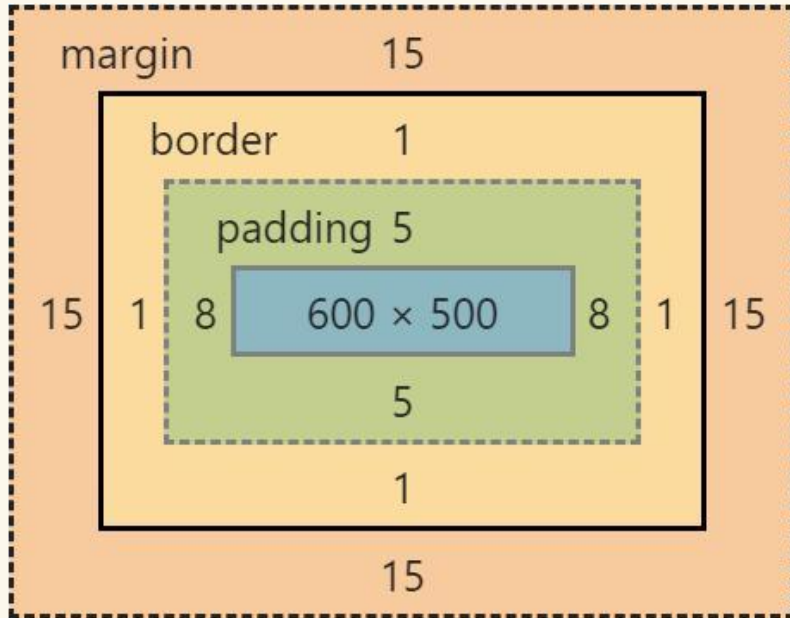
```
.container {  
  width: calc(50% - 20px);  
  padding: calc(10px + 5%);  
  margin-left: calc(2 * 10px);  
}
```

[calc\(\) | MDN](#)

The box model

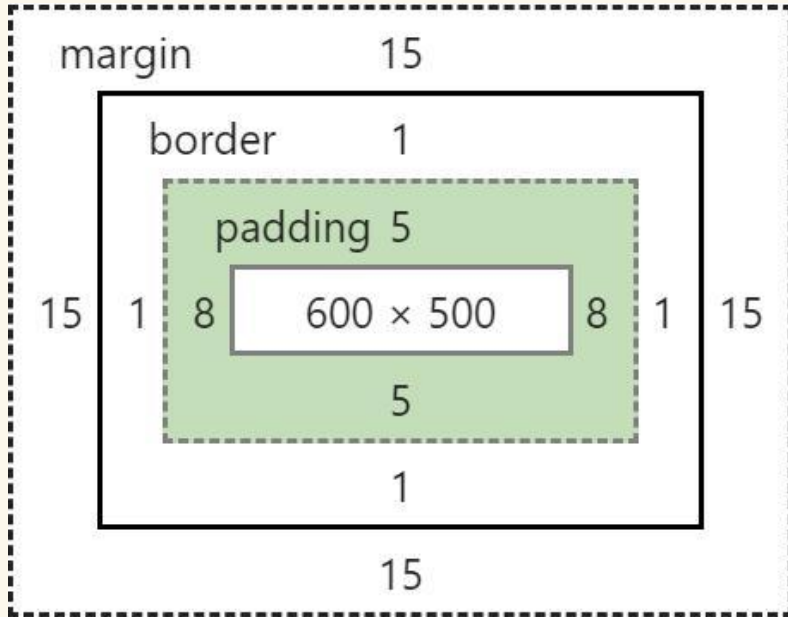
The box model: overview

Each element can have: padding, border, margin



Padding

Space between the border and the content



Padding values

It's possible to specify 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;
```

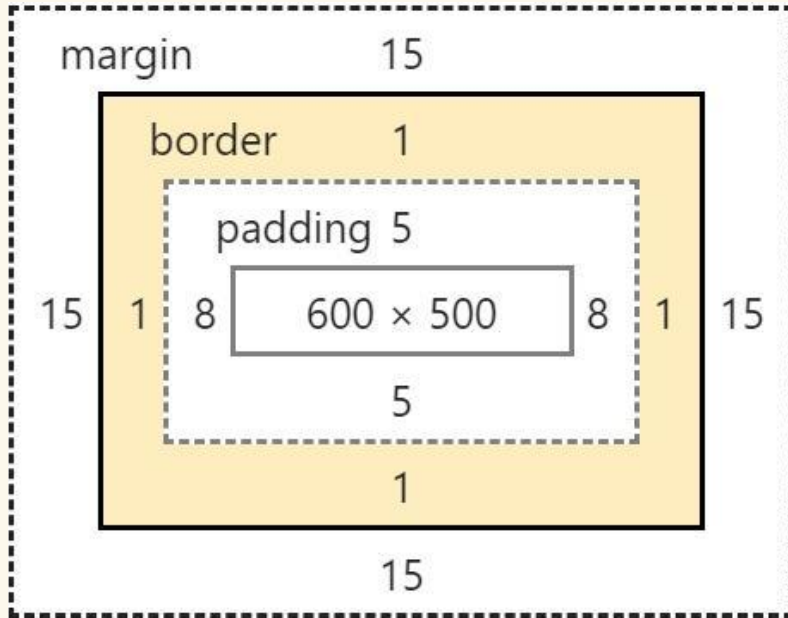
Reference

[CSS Padding](#)

[Padding | MDN](#)

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 properties

you can specify each property separately, or all three together

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

shorthand

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

edge specific properties

```
border-bottom-color: rgb(170, 50, 220, .6);
```

rounded Borders

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

Reference

[CSS Borders](#), [CSS border | MDN](#)

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;  
}
```

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;  
}
```

Box-sizing reset example

style.css

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

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

.box, .box2, .measure, .special {
  width: 100px;
}

.box {
  border: 4px solid orange;
  padding: 1px;
}

.box2 {
  border: 12px solid red;
}
```

```
.measure {
  background: pink;
}

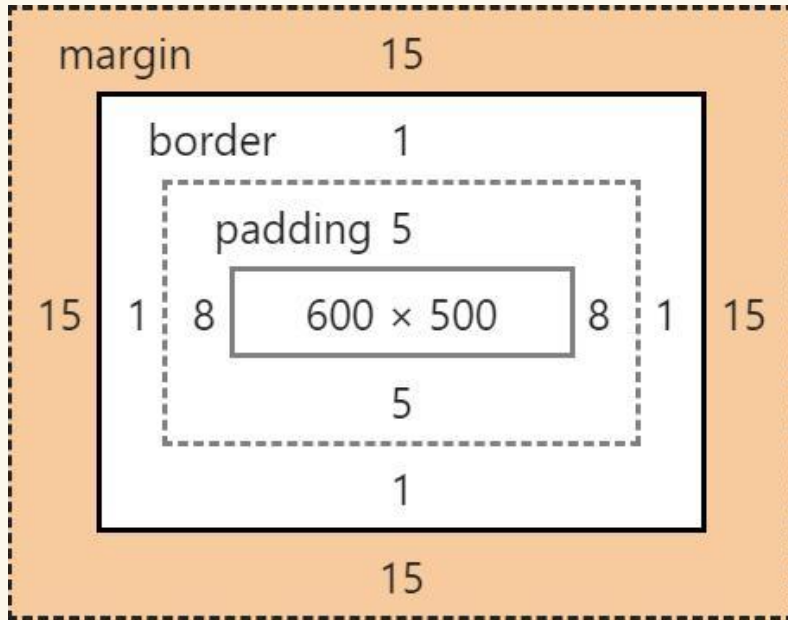
.special {
  background: yellow;
  border: 20px solid blue;
  box-sizing: content-box;
}
```

index.html

```
<div class="box">normal</div>
<div class="box2">variant</div>
<div class="measure">m</div>
<div class="special">i am different</div>
```

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 | MDN](#)

Margin: Auto margins

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](#), [text-align | MDN](#)

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>
```

Margin: Collapsing margins

- Only vertical margins collapse
 - left and right margins don't
- Vertical margins on nested elements collapse under certain conditions
 - only when the parent element doesn't have height, border or padding set

More details: [The Rules of Margin Collapse](#)

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 where necessary
 - 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 future lesson)

References

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: Better Best-Practice](#)

[CSS Tricks: Box Sizing](#)

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

[The Rules of Margin Collapse](#)