

Introduction

What should you know

- Basic understanding of HTML
- Comfortable using text editor

1. Getting Started

HTML and CSS

HTML

- Structure of content
- Semantic meaning

CSS

- Styling
- Appearance

Referencing CSS

External

Using separate CSS file that you call in you HTML file.

There is two ways to call the CSS file:

- Add referenced (link) within <head> of the document.
- @import method **EX:** @import url('/styles/layout.css');

Inline

Example:

```
<p style = "color= red"> red paragraph </p>
```

Internal

Example:

```
<head>
<style>
    H1 {
        color: green;
    }
</style>
</head>
```

Optimizing images and retain displays

Retina Display

- Trademarked term by Apple.
- Retina = high pixel density

- Introduced in 2010 with iPhone 4, then in 2012 with the iPad 3, MacBook Pro, and iMac.
- Other manufactures have their own version

Pixel Density

- Refers to the number of pixels within a space.
- Measured in pixels per inch (PPI) or dots per inch (DPI)
- Retina displays have double the number of PPI/DPI

Absolute Paths

Absolute paths refer to a resource located on a server

Example:

``

Chapter Quiz

1. You have a large image that needs to fit into a 400 x 200 pixel area. What should you resize the image to if your users are using Retina displays?
Answer: 800 x 400 pixels
2. Which example of hotlinking should be avoided?
Answer: ``
3. Your CSS is defined in the 'global.css' file located under the 'css' folder under your project folder. How should you include this file in your HTML?
Answer: `<link rel="stylesheet" href="css/global.css">`
4. While HTML controls document structure, CSS controls _____.
Answer: content appearance

2. Core Concepts

CSS specifications and the W3C

W3C: World Wide Web Consortium

CSS syntax and terminology

Shorthand and Longhand properties

Padding-top: 10px; (longhand)

Padding: 10px 5px; (shorthand)

CSS Comments

- Comments in CSS start with a slash and asterisk and must close in opposite order.
Example: `/* Comment. */`
- Leave note for yourself and others
- Organize code blocks
- Comment out code to hide it temporarily

CSS values and units

Valid Examples:

Color: red;
Font-size: 10px;

Not Valid Examples:

Color: 10px;
Font0size: red;

Numeric Values

Width: 80%;
Height: 10em;
Border-width: 5px;
Animation-iteration-count: 5;

<length>

The <length> data types is used to specify sizing with two types of units: absolute and relative.

Absolute

- Fixed units, always the same size.
- Not affected by values in related elements
- **Example:** px, cm, mm, pt

Relative

- Relational sizing, not fixed
- Dependent upon values declared in parent and ancestor elements
- **Example:** em, rem, vw, vh
-

Unitless Values

- Some numeric values do not require a unit
- Key words do not require units like color: red;

CSS Function Values

- The syntax always includes the function names and parenthesis
- Example: transform: rotate (); (rotates an element)
Width: calc(); (calculates a computed value)
background-image: url(); (embedded an image to background)

Example:

```
/* Rotate an element */
transform: rotate(90deg);

/* Calculate a computed value */
width: calc(80% - 20px);

/* Embed an image to the background of an element */
background-image: url('myimage.png');
```

The color and property values

Basic Color Keywords

- Red
- Blue
- Green
- Black

Extended Color Keywords

- Mediumseagreen
- Oldlace
- Aliceblue

Hexadecimal

- Prefixed with a number followed by six characters (0-9 and A-F)
- Defines the red, green, blue values #rrggbb

Hexadecimal Shorthand and Longhand

#f00; (shorthand)

#ff0000: (longhand)

Hsl()

- The **hue** is specified as an angle
- The value can be declared with or without the degree unit

Example: hsl(270, 60%, 70%)

hsl(270deg, 60%, 70%)

- **Saturation** is represented with a percentage
- 100% is full saturation, 0% is gray
- **Lightness** is represented as a percentage
- 100% is white, 0% is black, and 50% is normal.

Hsla() Alpha Channel

- Can be represented as a decimal or percentage

Example: hsl(270, 60%, 70%, .15)

hsl(270deg, 60%, 70%, 15%)

Type and universal selectors

Type Selectors

- The most basic kind of selector, simple matching pattern

Example: h1{}

Universal Selectors

- Matches to all elements, of any type

Example: * {

boder: 1px solid black;

}

Class and ID selectors

Class selectors

- Add the class attribute to the HTML element
- Reusable
- Can be applied to any element

ID Selectors

- Add the ID attribute to the HTML element
- You define the value
- The value is the selector starting with # (# container)

Descendent selectors

Document Object Model

The DOM represents the relationships between HTML elements

Descendent Selectors

Example:

Selects only links inside of paragraph inside the section
Section p a {.....}

Grouping Selectors

Example:

Applies to both h1 and h2 elements
h1, h2,

Inheritance and specificity

Inheritance

CSS styles can be inherited from the ancestors to the decedent elements

Specificity Calculation

A. Count the number of ID selectors.

B. Count the number of class, attribute, and psuedo-class selectors.

C. Count the number of type and pseudo-element selectors.

*	a=0	b=0	c=0	=	000
p	a=0	b=0	c=1	=	001
.class	a=0	b=1	c=0	=	010
#id	a=1	b=0	c=0	=	100
.class p	a=0	b=1	c=1	=	011
#id p	a=1	b=0	c=1	=	101
ul li a	a=0	b=0	c=3	=	003

The cascade and importance

Cascade

- The cascade in Cascading Style Sheets refer to how style rules are applied based on specificity and source order

Importance

- The !importance keyword overrides source order and specificity

Example: p{

```
    Font-size: 12px !importance;  
}
```

Chapter Quiz

1. Which CSS will cause your links to have a solid blue background that changes to semitransparent on hover?

Answer: `a:link { background: #0000ff; } a:hover { background: rgba(0,0,255,0.5); }`

2. Which CSS rule takes precedence over the others listed?

Answer: `div#sidebar2 p {}`

3. You can use the `_____` pseudo-class to set a different color on a link if it was clicked on.

Answer: `a:visited`

4. Which color will look the brightest on your screen, assuming the background is white?

Answer: `background-color: rgba(170,170,170,0.2);`

5. Which CSS selector can you use to select all elements on your page associated with the two classes `header` and `clear`?

Answer: `.header.clear {}`

6. A universal selector is specified using a(n) `_____`.

Answer: `*`

3. The Box Model

Inline, Block, and Display

Inline

- Takes up the same space as their content
- Elements are displayed in line from the left
- Elements will only wrap when items cannot fit
- `<a>`, ``, ``

Example:

`content`

`content content content`

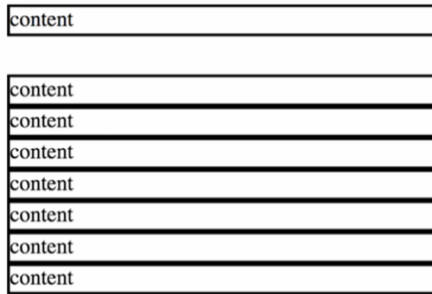
`content content content content content`
`content content`

Block

- Same height as content, same width as container
- Always starts on a new line

- <p>, <h1>, <article>, <section>

Example:

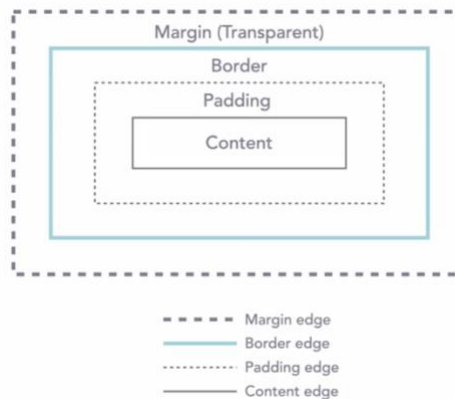


The Display Property

- Used to change the default behavior of inline and block-level element.

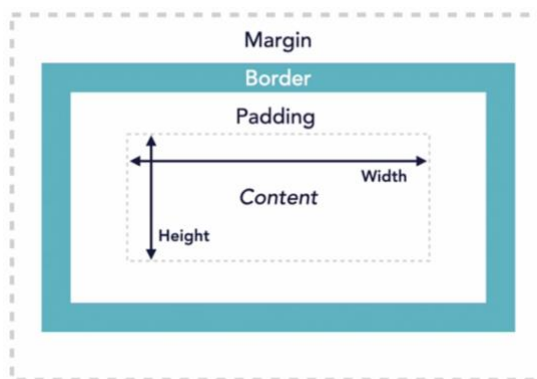
The Box Model Properties

- Content
- Padding Box
- Border Box
- Margin Box



Box Properties

- Width
- Height
- Padding
- Margin



Values and Units

For the box properties, the <percentage> and <length> data types are used

The Box Properties Syntax and Usage

Width and Height

The width and height properties change the size of the content box. Inline elements require the display property.

Example:

```

/* Block element */
div {
  width: 100px;
  height: 100px;
}

/* inline-element */
span {
  width: 100px;
  height: 100px;
  display: block;
  /* OR */
  display: inline-block;
}

```

Padding

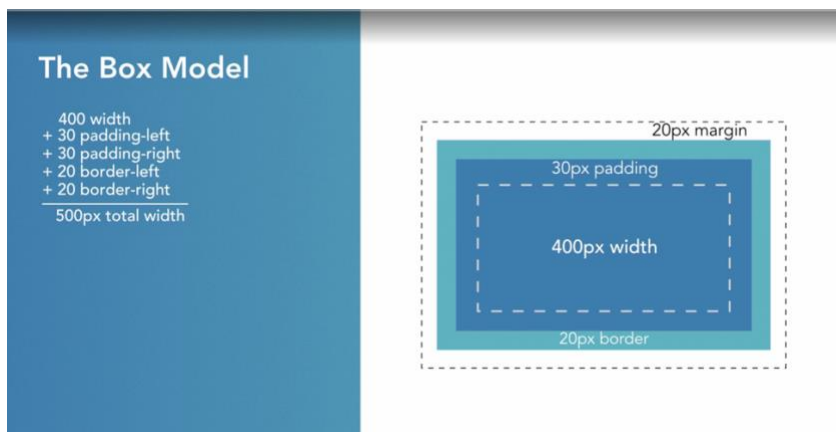
Padding adds/removes space inside of the element but around the content box

Margin

Margin adds/removes space around the element

Border

The border property displays border between the margin and padding.



Margin and Negative Values

By default, block elements stack in top of each other.

Use negative margin values to move elements outside of the stacking position.

Chapter Quiz

1. Which properties are included in the box model?

Answer: height, padding, and border

2. Which width specification is relative?

Answer: 10vw

3. You change a certain CSS rule in your browser's Developer Tools view. When is this change reflected inside the browser?

Answer: The change is seen immediately.

4. You have two consecutive span elements on your page, and you see a small gap between them. Is it possible to eliminate this gap?

Answer: The gap can be eliminated by changing the font size of the container element to zero.

5. What is a CSS reset?

Answer: a set of CSS rules that resets all styles to a clean and consistent baseline

5. Typography

Typography of the web

Typography: Arranging type for readability and to engage and communicate with the reader

Typeface: A set of fonts, designed with common characteristics composed of glyphs

Changing the font-family

Example:

```
body{  
font-family: Helvetica, Arial, sans-serif;  
}
```

Generic font-family Names

- Serif (serif fonts)
- Sans-serif (sans-serif fonts)
- Cursive (script or decorative fonts)
- Fantasy (decorative fonts)
- Monospace (monospace fonts)

Web-Safe Fonts

Fonts that are commonly pre-installed on majority of computers or devices (Examples: Arial, Times New Roman)

Fonts may vary among operating systems

Font-Weight and font-style

Font-weight

- Light
- regular
- Semi-Bold
- Bold
- Extra-Bold

Font-weight Keywords

- Normal is equal to 400 and is also default for body text
- Bold is equal to 700 and is default for headings

Number values will map to the nearest typeface available

Font-style

Used to add or remove italic style. There are three values: italic, oblique, and normal.

Example:

```
p {  
Font-style: italic;  
}
```

Web Fonts with @font-face

- Downloaded font files
- Included in your project files just like other files (i.e. images)
- Declare and link font files using @font-face

Example:

```
@font-face {  
  font-family: "My Font";  
  src: url(my-font.woff); /* relative path */  
  src: url(http://example.com/fonts/my-font.woff); /* absolute  
path */  
}
```

Relative File Paths



The font-size property

- Different units for specifying the font size
- Relative values are calculated based on the newest ancestor element.
- Absolute values are fixed and not affected by ancestor elements

Font-size: px

- Screens are measured in pixels
- Absolute value. Great for accuracy
- Use whole numbers, avoid decimals
- Browser default = 16px

Font-size: em

- Named after the letter "m"
- Relative unit
- Can use whole numbers or decimal points
- 1em = inherited font-size

Font-size: rem

- Root em
- Relative unit
- Only relatives to the root element (<html>)
- Not affected by parent or ancestor elements

Font shorthand

Property

- Font-style
- Font-size
- Font-weight
- Font-family
- Font -variant
- Line-height

Font Shorthand syntax rules

- Order matters! Font-style, font variant, and font weight
- Font-variant = normal, or small-caps only
- Font-size/line-height
- Font-family must be the last value specified
- Font: italic small-caps bold 24px/1.5 Helvetica, sans-serif;

Text-decoration, text-align, and line-height

Text-decoration

Example:

Text-decoration: none; /* removes underline */

Text-align

- The text-align property can be used to align content with the block element
- Add it to the HTML element itself or the parent element.

Line-height

- The line-height property sets the height of the space between two lines of text.
- Closely related to font-size
- Can use different value types (px, %, em, rems)

Chapter Quiz

1. You want your styling to be based on a font stack consisting of three fonts. Where should the generic font for your font family be specified?
Answer: It should be the last one on the list.
2. What is one disadvantage of using a web font service?
Answer: It is not always a free service.
3. Which shorthand font specification is invalid?
Answer: font: oblique 24px bold "my font", sans-serif;
4. What is the font size of an element in pixels, given that this element's parent has a font size set to 32px and the element's font size is set to 0.5625em?
Answer: 18px
5. Generally speaking, _____ typefaces are used for formal content, decorative details, and code sections, respectively.
Answer: serif, script, and monospace

5.Layouts: Float and Position

Float and collapsed container

Option 1: overflow

```
<div class="parent">
  <div class="floated">...</div>
  <div class="floated">...</div>
</div>

.floated {
  float: left;
}
.parent {
  overflow: hidden;
  /* OR */
  overflow: auto;
}
```

overflow: hidden;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Alias voluptates, enim excepturi debitis quibusdam deserunt, velit commodi ipsum doloribus, tenetur id culpa corrupti itaque, reprehenderit illum minima. Voluptas quos, mollitia! Lorem ipsum dolor sit

overflow: scroll;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Alias voluptates, enim excepturi debitis quibusdam deserunt, velit commodi ipsum doloribus, tenetur id culpa corrupti itaque, reprehenderit illum minima.

Option 2: "clearfix" Hack

CSS snippet added to the parent of floated elements.

```
<div class="clearfix">
  <p>floated element</p>
  <p>floated element</p>
</div>

.clearfix:after {
  content: "";
  display: table;
  clear: both;
}
```

Option 2: "clearfix" Hack

```
.clearfix:after {
  content: "";
  display: table;
  clear: both;
}

<div class="clearfix">
  <div>floated element</div>
  <div>floated element</div>
</div>

.another-name:after {
  content: "";
  display: table;
  clear: both;
}

<div class="another-name">
  <div>floated element</div>
  <div>floated element</div>
</div>
```

Option 3*: display

* not currently supported in all browsers

```
<div class="parent">
  <div>floated element</div>
  <div>floated element</div>
</div>

.parent {
  display: flow-root;
}
```

Position

- static: not positioned
- relative: relative to current position
- absolute: relative to containing element
- fixed: relative to the viewpoint
- sticky: relative to containing element and viewpoint

Chapter Quiz

1. Use the CSS property on an element to prevent other elements from floating beside it.

Answer: clear

2. Where will you have little expectation to see that the float property is used?

Answer: on sites where the grid property is used

3. While an absolute element is positioned , a fixed element is positioned .

Answer: relative to its closest positioned ancestor; relative to the viewport

4. When would you put adjacent text elements in two containers and float them both?

Answer: when you want to show the elements in two separate columns

6.Layouts: Flexbox and Grid

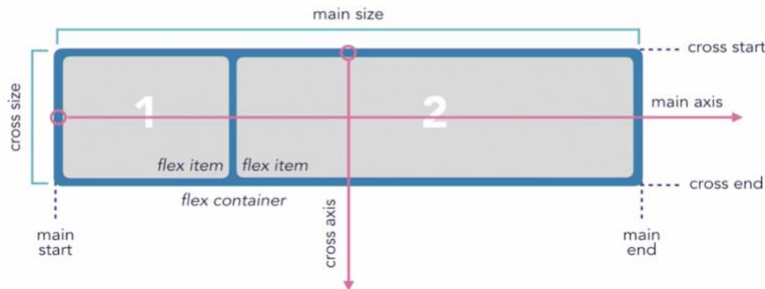
Introduction to Grid and Flexbox

Flexbox: Distribution across a single axis

Grid: Layouts with both rows and columns

Introduction to Flexbox

Flexbox Terminology

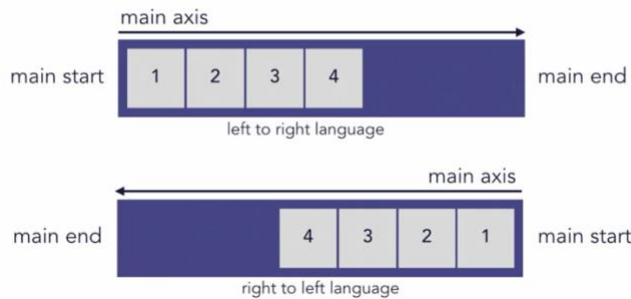


Flexbox: Orientation and ordering

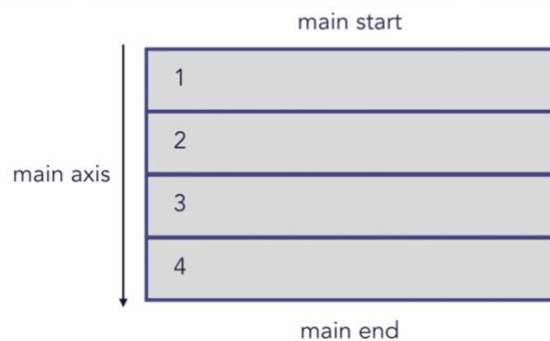
flex-direction: Determines the direction of the main axis.

There are four values: row, row-reverse, column, and column-reverse

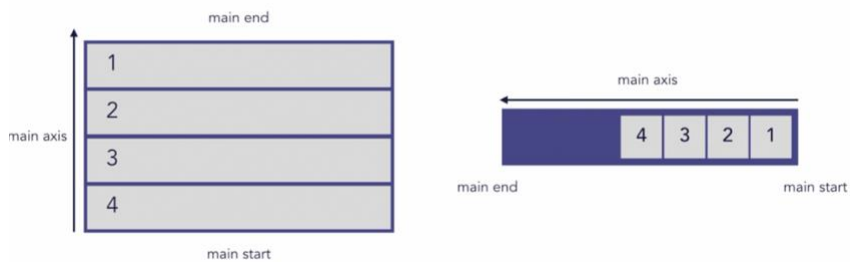
flex-direction: row



flex-direction: column



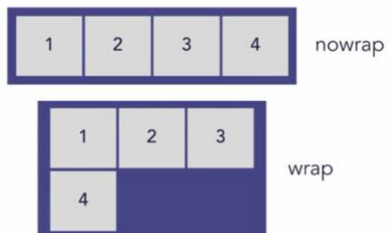
flex-direction: column-reverse | row-reverse



flex-wrap: nowrap



flex-wrap: wrap



flex-wrap: wrap-reverse



flex-flow Shorthand

Shorthand:

flex-flow: column wrap;

Longhand:

flex-direction: column;

flex-wrap: wrap;

Flexible sizing

flex- Properties

- flex-basis sets the initial size of the flex-items
- flex-grow: determines how items will expand if there is extra space in the container.
- flex-shrink: determines how items will shrink if there isn't enough space in the container.

flex Shorthand

flex: grow shrink basis; -> flex: 0 1 100px;

Flexbox: Alignment

Aligning Items in a Flex Container

- justify-content: align items on the main axis
- align-items: aligns items on the cross axis

Example:

```
.box{  
display: flex;  
align-items: center;  
justify-content: center;  
}
```

Introduction to CSS Grid

CSS Grid

- grid container: the parent element
- grid items: the child element within the grid container

grid and display

```
.grid-container{  
display: grid;  
/* or */  
display: inline-grid;  
}
```


The explicit grid

Creating an Explicit Grid

- grid-template-columns
- grid-template-rows

Example (3 columns and 2 rows):

```
.grid-container{
display: grid;
grid-template-columns: 100px 100px 100px;
grid-template-rows: 100px 100px;

}
```

The Fraction Unit: fr

fr: represents a fraction of available of the space in the grid container

Example:

```
grid template-columns: 1fr 1fr 1fr;
grid template-columns: 1fr 2fr 1fr;
```

repeat()

repeat([number of tracks]). [size of tracks])

grid template-columns: 1fr 1fr 1fr; = grid template-columns: repeat(3, 1fr);

gap-row and gap-column

shorthand:

gap: 10px 20px;

longhand:

row-gap: 10px;

column-gap: 20px;

Gap for Flexbox and Grid

The original grid layout specification included the **grid-gap**, **grid-row-gap** and **grid-column-gap** properties.

It has been updated to **gap** to be used for both grid and Flexbox.

Currently, **gap** for Flexbox is only supported in Firefox.

```
/* shorthand*/
gap = grid-gap
```

```
/* longhand */
row-gap = grid-row-gap
column-gap = grid-column-gap
```

The implicit grid

Explicit vs. Implicit grid

Explicit Grid



Implicit Grid

```
.grid-container {  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-template-rows: 1fr 1fr 1fr;  
}
```

Grid Placement properties

Defining the grid Structure

```
.grid-container {  
  display: grid;  
  grid-template-columns: 100px repeat(2, 1fr);  
  grid-template-rows: 100px;  
  grid-auto-rows: 200px;  
  grid-auto-columns: 50px;  
}
```

Positioning grid Items

```
grid-column = grid-column-start  
              grid-column-end
```

```
grid-row = grid-row-start  
           grid-row-end
```

Chapter Quiz

1. What gets reversed when using the `flex-wrap: wrap-reverse` option?

Answer: cross start and cross end

2. How does the `display: inline-flex` option differ from `display: flex`?

Answer: It sets the flex container width to be like its content.

3. A flex container is set with a width of 200px and `flex-wrap: wrap`. It has 3 flex items set with `flex: 1 0 100px`. What is the effective width of the third flex item?

Answer: 200px

4. You want your flex items to have a maximum width of 300px. Which flex rule will set this?

Answer: `flex: 0 1 300px`;

5. What is the benefit of troubleshooting grid issues in Firefox compared to other browsers?

Answer: Firefox has a dedicated grid visualization tool.

6. Which flexbox alignment option distributes the items evenly across the main axis, with the first item at the start of the axis and the last item at the end?

Answer: `justify-content: space-between;`

7. What is the outcome of setting a grid container as an inline-grid?

Answer: It will display inline next to other inline elements.

8. Your flexbox container is set with the `justify-content: space-between;` property. Should you be concerned about increasing the padding of the flex items?

Answer: Yes, large padding values will cause the items to overflow the container.

9. Which statement is true regarding flexboxes?

Answer: They are used for space distribution across a single axis.

10. When will you want to use an explicit grid and an implicit grid together?
when you know the minimum number of grid items, but not the maximum

7. Advanced Selectors

Relational Selectors: Combinators

Descendant Selectors

Creates matching patterns based on the relationship between nested elements

```
<section>                section a { ... }
  <a href="#"></a>         .example p { ... }
</section>

<div class="example">
  <p></p>
</div>
```

Child Combinators (>)

Only matches to direct child element

- descendent selector
parent child{}
ancestor descendant{}
- child combinator
parent > child {}

Sibling Combinators (+, ~)

```
<parent>
  <sibling></sibling>
  <sibling></sibling>
</parent>
```

Adjacent Sibling Combinators (+)

```
h1 + p {}
```

```
<section>  
  <p>Sibling to heading.</p>  
  <h1>Heading</h1>  
  <p>Sibling to heading</p><!-- adjacent sibling -->  
  <p>Sibling to heading.</p>  
</section>
```