Web Development HTML5 & CSS

...

Eng. Wael Salama
Cell: 0122 172 4503
eng.wsalama@gmail.com
http://wsalama.000webhostapp.com

Git

...

https://github.com/engwsalama/webdev_html_css.git

HTML5

...

HTML Formatting - Headings - Paragraph

•••

Image

•••

List

•••

Table

•••

Form

...

Document Object Model

...

CSS
Cascaded Style Sheet
...

Working with CSS

•••

Styles...

Font

...

Border

...

Styling table

...

Div- Span

...

Links

...

HTML4 Organization

...

-
-
-
-

HTML5 Organization

...

- <header>
- <nav>
- <section>
- <footer>

Form & Datalist

...

Multiple & descendant

…

Child

...

Attribute

...

hover

...

Pseudo-classes

- A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s).
- State of element based on user activity {page interacts with user activity}

Before & After

•••

Pseudo-Element

- A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s).

Adjacent

•••

Pseudo-Element

- A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s).

Selection

...

Pseudo-Classes & Pseudo-Element

nth-child

...

Combined Selectors

...

Cascade

...

-1-

Cascade

- Three things decide which styles get applied
  - Importance: normal (any style) or !important (color:red !important;)
  - Specificity:
  - Source order

-1-

Cascade

Specificity: means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied.

- Look at the element that is being styled. Add the total number of each category in the selector expression. Treat this like a software version number.:
  a. ID selectors (e.g., #example).
  b. Classes, pseudo-classes, attribute selectors (e.g., .example), attributes selectors (e.g., [type="radio"]) and pseudo-classes (e.g., :hover).
  c. Type selectors (elements and ::pseudo-elements)(e.g., h1) and pseudo-elements (e.g., ::before).

-1-

Cascade

Specificity Examples

- 0.4.2 = .red .big p.one.two span { }
- 1.1.1 = #simon p.first { }

The second version is more important and gets applied second (if these were pointing at the same element)

-1-

Cascade

Source Order

CSS declarations come from different origins:

- The user-agent (browser) style sheet; *each browser has own styles*
- the author style sheet; *each developer has own styles as different types below*
- and the user style sheet.

Within the author style sheet origin we also have:

- External stylesheet;
- Embedded <style> element;
- Inline style attribute.

Float

...

Text Shadow

...

Box Shadow

...

Google Fonts

...

@font-face

...

CSS Functions

...

- CSS functions are used as a value for various CSS properties.
- rgb() function to provide a color value
- attr() function to retrieve the value of an HTML attribute.

Relative Units

...

- Em
- Rem
- Viewport

-1-

Line-Height

...

- Unitless

Custom Properties
CSS Variables
...

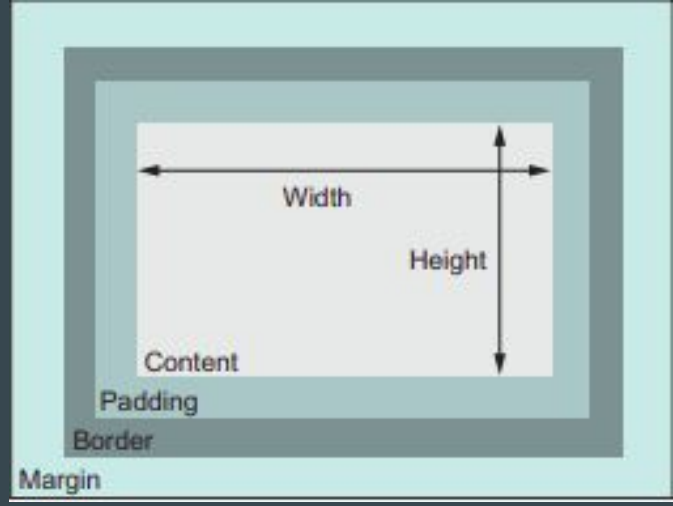
- Define one variable
- Var() function
- Variable scope

CSS Layout

...

-1-

CSS Layout



Why two columns did not sitting side by side?...
They line wrapped. that's because of the default behavior of the box model.

- When you set the width or height of an element, you're specifying the width or height of its content; any padding, border, and margins are then added to that width

Example:

- An element with a 300px width, a 10px padding, and a 1px border has a rendered with of 322px (width+padding+border)for both sides

-1-

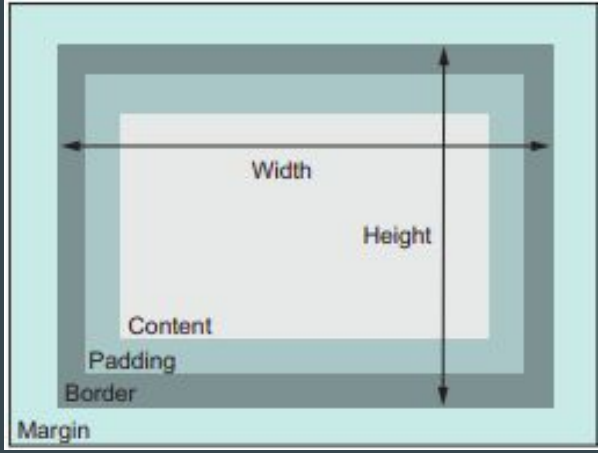
CSS Layout

To solve this problem

- Make the second column width 26%.
- Use calc(30% - 3em) in second column width
- Adjusting the box model

-1-

CSS Layout



Box Model.

- As the default, box model is not what you will typically want to use , because it will be equal to content + padding+border.

- Instead, you will want your specified widths to include the padding and borders.
- CSS allows you to adjust the box model behavior with its box-sizing property
- By default , box-sizing is set to the value of content-box, this means that any height or width you specify sets the size of the content box.
- You can assign a value of border-box to the box sizing instead.
- That way , the height and width properties set the combined size of the content , padding, and border,

CSS Position

- Static
- Relative
- Absolute
- Fixed

Combined absolute-relative

Flexbox

Flexbox

...

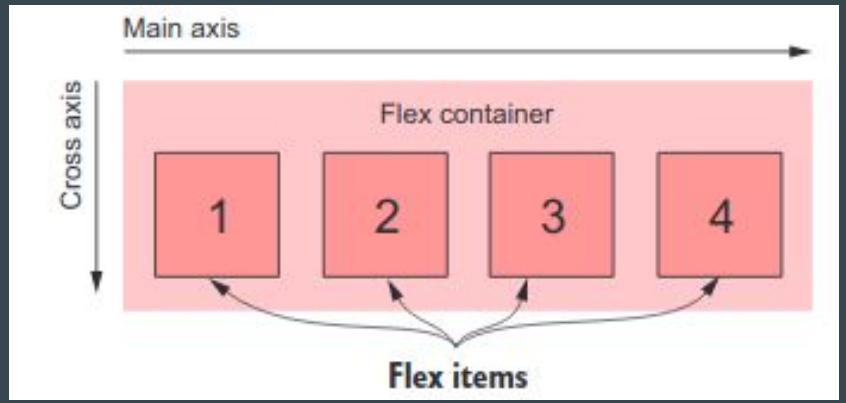
- Flexbox is one dimensional that deals with layout in one dimension at a time — either as a row or as a column.
- The flex-wrap property specifies whether the flexible items should wrap or not.
- If the elements are not flexible items, the flex-wrap property has no effect.

CSS Flexbox

- Flexbox begins with the familiar *display property*
- Applying display:flex to an element turns it into a flex container, and its direct children turn into flex items.
- By default,flex items align side by side, left to right, all in one row.

The flex container properties are:

- Flex-direction
- Flex-wrap
- Flex-flow
- Justify-content
- Align-items
- Align-content



CSS Flexbox- Properties for the parent

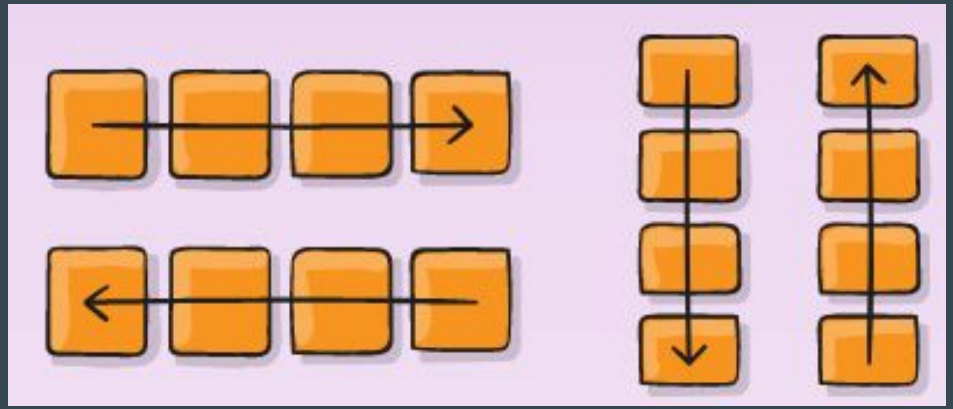
Flex Container

display:flex;

- This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

Flex-direction:row | column

- Think of flex items as primarily laying out either in horizontal rows or vertical columns.
- flex-direction:reverse-row;
- flex-direction:reverse-column;

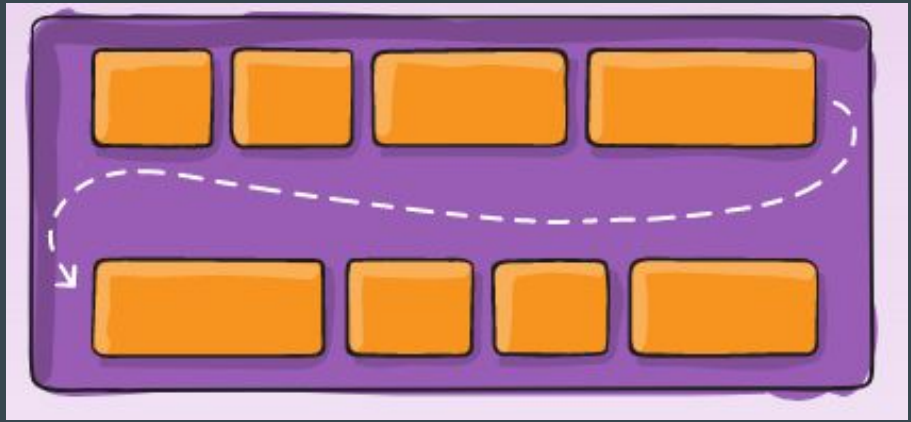


CSS Flexbox- Properties for the parent

Flex Container

- Flex-wrap: nowrap | wrap | wrap-reverse;

By default, flex items will all try to fit onto one line. You can change that and allow the items to wrap as needed with this property.



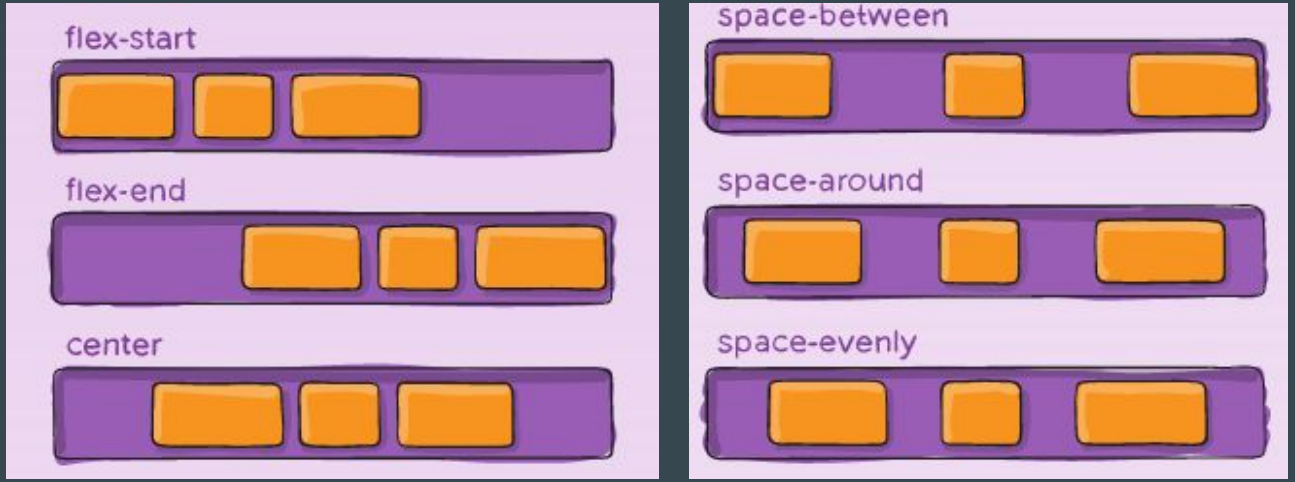
flex-flow

- This is a shorthand for the flex-direction and flex-wrap properties, which together define the flex container's main and cross axes.
- The default value is row nowrap.

CSS Flexbox- Properties for the parent

Flex Container

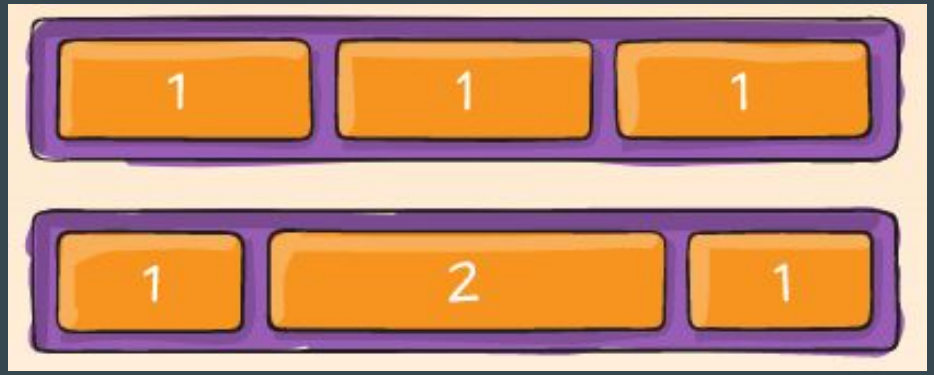
Justify-content: This defines the alignment along the main axis.



justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly

CSS Flexbox- Properties for the Children

Flex items



Flex-grow

- This defines the ability for a flex item to grow if necessary
- Negative numbers are invalid.
- flex-grow: <number>; /* default 0 */

flex-shrink

- This defines the ability for a flex item to shrink if necessary.
- Negative numbers are invalid.
- flex-shrink: <number>; /* default 1 */

CSS Flexbox- Properties for the Children

Flex items

Flex-basis

- This defines the default size of an element before the remaining space is distributed.
- It can be a length (e.g. 20%, 5rem, etc.) or a keyword.
- The auto keyword means "look at my width or height property"
- flex-basis: <length> | auto; /* default auto */

CSS Flexbox- Properties for the Children

Flex items

Flex

- This is the shorthand for flex-grow, flex-shrink and flex-basis combined.
- flex-shrink and flex-basis are optional.
- The default is 0 1 auto, but if you set it with a single number value, it's like <number> 1 0.
- flex: none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]

CSS Grid

Grid

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
- A grid layout consists of a parent element, with one or more child elements.
- Display Property should be grid or inline-grid

CSS Statements

There are two kinds of statements:

- Rulesets (or rules) that, as seen, associate a collection of CSS declarations to a condition described by a selector.

- At-rules that start with an at sign, '@' followed by an identifier.

- Each type of at-rules, defined by the identifier.

- They are used to convey metadata information (like @charset or @import), conditional information (like @media or @document), or descriptive information (like @font-face).

CSS Statements

Nested Statements

- These are statements that can be used in a specific subset of at-rules.
- @media at-rule content is applied only if the device on which the browser runs matches the expressed condition
- @document at-rule content is applied only if the current page matches some conditions

CSS Structure

...

-1-

How CSS is Structured?

1. Applying CSS to your HTML
- External stylesheet
- Internal stylesheet
- Inline stylesheet

How CSS is Structured?

2. Selectors
    a. Universal Selector [Example: * will match all the elements of the document.]
    b. Type selector
    c. Class selector
    d. ID selector
    e. Attribute selector
    f. Grouping selectors
        i. Selector list

How CSS is Structured?

h. Combinators
   a. Child Combinator
   b. General sibling combinator
   c. Adjacent sibling combinator
   d. Column combinator
i. Pseudo
   a. Pseudo classes
   b. Pseudo elements

Refer to css selector at : https://github.com/engwsalama/webdev_html_css

How CSS is Structured?

3. Specificity & Cascade
4. Properties & Values
5. Functions
6. @rules...[@import]
7. Shorthands
8. Comments /* */

HTML References

- https://developer.mozilla.org/en-US/docs/Web/HTML
- https://www.w3schools.com/html/default.asp
- https://www.w3schools.com/css/
- https://flatuicolors.com/
- http://www.webestools.com/
- https://www.fontsquirrel.com/tools/webfont-generator