Introduction to Frontend Engineering

Oluwasetemi Ojo and Desmond Nyamador

Lesson Outline

- Building Blocks
- Tools (Vscode Introduction)
- Introduction to HTML
- Building out our first html page
- Semantic HTML
- Limitations of HTML
- Emmets

Introduction to Frontend Engineering (HTML)


HTML
CSS
JS

Building Blocks

Introduction to Frontend Engineering

- Hyper Text Markup Language (HTML)
- Cascading Style Sheets (CSS)
- Uniform Resource Locators (URLs)
- Hypertext Transfer Protocol (HTTP and HTTPS)
- JavaScript Programming Language (ECMAScript)
- Document Object Model (DOM)

Others include: DNS - Domain Name System, Web Browsers, Web Servers and Web Hosting, Internet (Understanding how internet works)

> *HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.*
>
> – *Wikipedia*

Building Blocks

Tools

Introduction to Frontend Engineering

Visual Studio Code

Download VS code

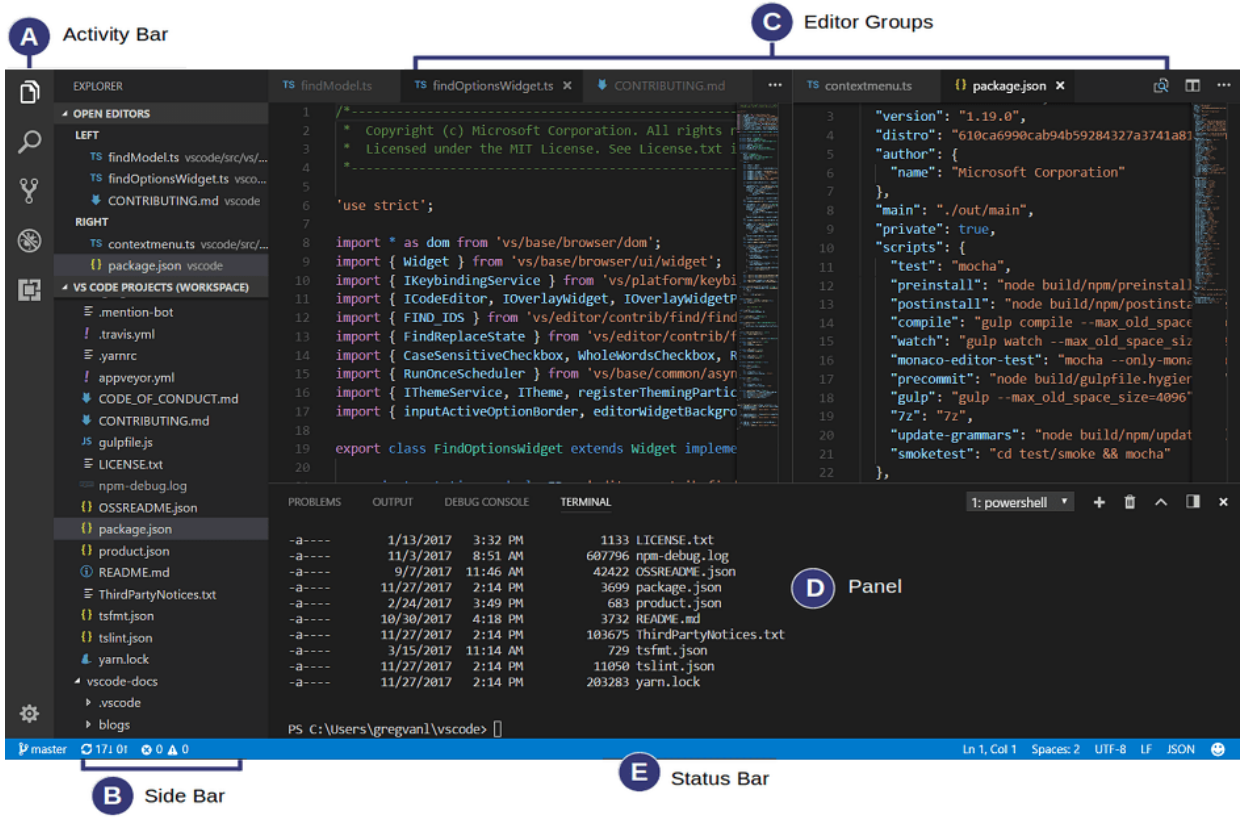
Get familiar with VS code Interface

Install the Live Server Extension

Install the Prettier Extension

Important tips and tricks
Know and learn important shortcuts, Windows or Mac OS

PS: Change the look and feel of your vscode with themes, install any preferred theme from the marketplace.


A Activity Bar
C Editor Groups
B Side Bar
D Panel
E Status Bar
EXPLORER
OPEN EDITORS
LEFT
findModel.ts vscode/src/vs/...
findOptionsWidget.ts vsco...
CONTRIBUTING.md vscode
RIGHT
contextmenu.ts vscode/src/...
package.json vscode
VS CODE PROJECTS (WORKSPACE)
.mention-bot
.travis.yml
.yarnrc
appveyor.yml
CODE_OF_CONDUCT.md
CONTRIBUTING.md
gulpfile.js
LICENSE.txt
npm-debug.log
OSSREADME.json
package.json
product.json
README.md
ThirdPartyNotices.txt
tsfmt.json
tslint.json
yarn.lock
vscode-docs
.vscode
blogs
findModel.ts
findOptionsWidget.ts
CONTRIBUTING.md
contextmenu.ts
package.json
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
PS C:\Users\gregvanl\vscode>
master
Ln 1, Col 1
Spaces: 2
UTF-8
LF
JSON

Getting Started with HTML

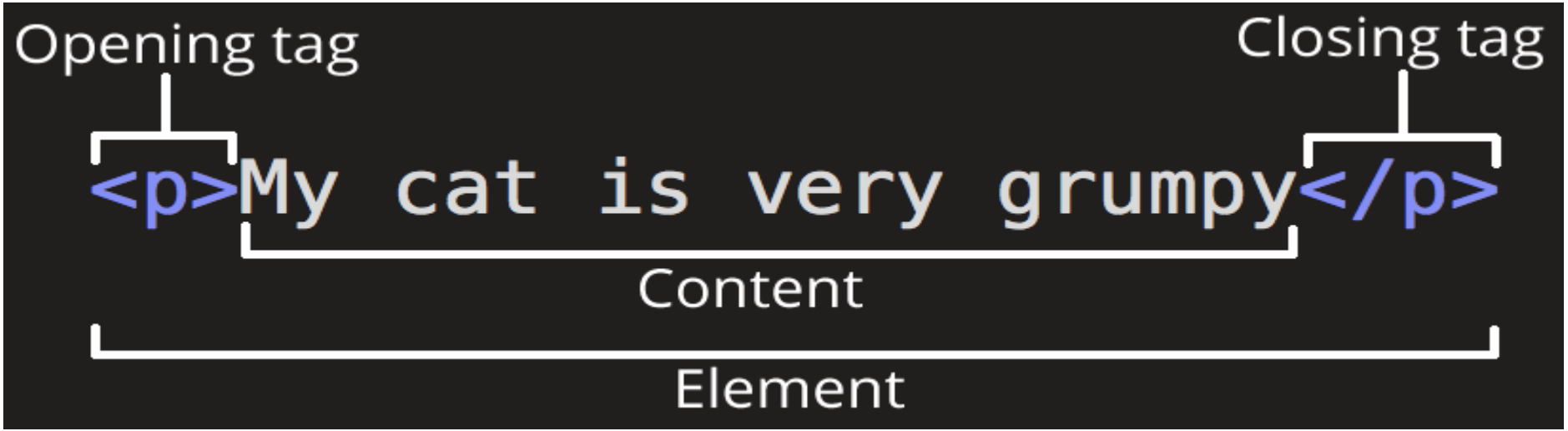
Introduction to Frontend Engineering

Getting Started with HTML

Few important things to note:

- Basic Structure
- Element or Tags
  - Opening and Closing Tags
  - Self closing Tags
  - Inline or block Tags
  - Meta tags
- Attributes
  - name=”value”
  - Single value attribute
- Reference for more about HTML Tags
  - MDN HTML reference
  - HTMLreference.io

Introduction to Frontend Engineering


Opening tag
Closing tag
My cat is very grumpy
Content
Element

Exploring HTML Tags

Few important tags to note:

- h1 to h6
- p, em, i, a, blockquote, sub, sup,
- ol, ul, dl, datalist
- Img, video, audio
- table, tbody, thead, tr, th, td
- form, input, select, button, textarea, fieldset
- div, span, link, style, meta, title, details, head, body, html
- For more on HTML Tags
  - MDN HTML reference
  - HTMLreference.io

Introduction to Frontend Engineering

Building our first HTML page

Introduction to Frontend Engineering

Building our first HTML page

Let’s get our hands dirty with code.

Introduction to Frontend Engineering

Building our first HTML page

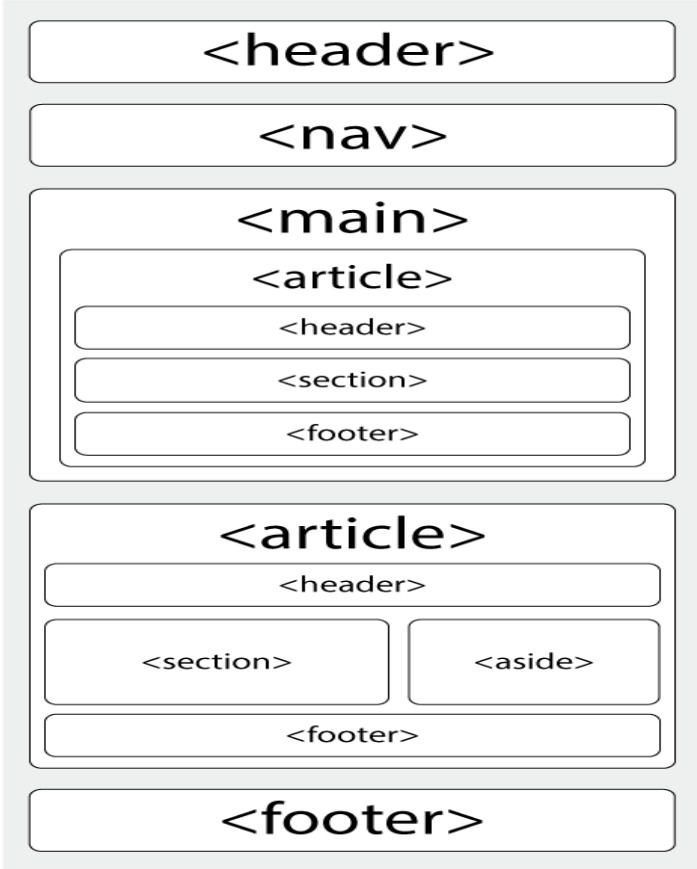
Open VScode and build something awesome

Introduction to Frontend Engineering

Semantic HTML

Introduction to Frontend Engineering

```
<header>
  <nav>
    <!-- main navigation in here -->
  </nav>
</header>
<main>
  <article>
    <!-- article content in here -->
  </article>
  <aside>
    <!-- aside content in here -->
  </aside>
</main>
<footer>
  <!-- footer content in here -->
</footer>
```



Semantic HTML

- header
- main
- article
- aside
- section
- footer
- figure
- mark
- figcaption
- time
- nav

Introduction to Frontend Engineering

Limitations of HTML

Introduction to Frontend Engineering

Limitations of HTML

- You can not create dynamic content
- It has limited designing capabilities
- Syntax errors are not identified or displayed by HTML
- Any type of calculations can not be done in HTML
- You can not create interactive web pages with HTML
- Complex HTML code is hard to read and understand

Introduction to Frontend Engineering

More Practical on HTML

Introduction to Frontend Engineering

Emmets

Introduction to Frontend Engineering

Emmets

- Abbreviations - #page>div.logo+ul#navigation>li*5>a{Item $}
- Syntax - class, id, siblings, child, climb up and multiplication
- Let stop here for now. For more information check the documentation.

Introduction to Frontend Engineering


HTML
CSS
JS

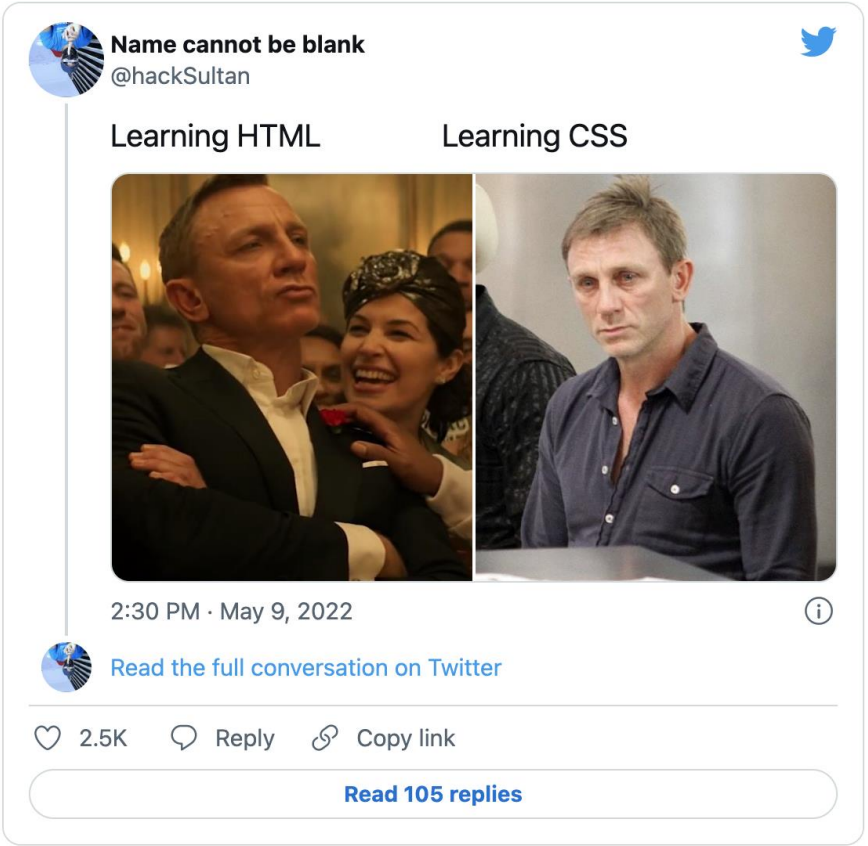
Cascading Style Sheets

Let’s talk about CSS

Lesson Outline

- CSS Basics
- Syntax
- Selectors
- Adding CSS to HTML
- Styling HTML pages
- CSS Variables and Comments
- Play CSS Game
- Emmets

Introduction to Frontend Engineering (CSS)


Name cannot be blank
@hackSultan
Learning HTML
Learning CSS
2:30 PM · May 9, 2022
Read the full conversation on Twitter
2.5K
Reply
Copy link
Read 105 replies

> *Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. Although most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.*
>
> – *Wikipedia*

CSS Basics

- selectors
- rule set
  - curl braces {}
  - properties
  - values
- adding css to html
- specificity
- cascading

Introduction to Frontend Engineering

Introduction to Frontend Engineering

CSS Syntax

```
selector { property: value; ...}
```

```
selectorlist {
  property: value;
  [more property:value; pairs]
}

... where selectorlist is: selector[:pseudo-class] [::pseudo-element] [, more selectorlists]

See selector, pseudo-class, pseudo-element lists below.
```

```css
/*
  this speak to a paragraph contained in an
  html to change the color from the default
  appearance to a red color.
*/
p {
  color: red;
}

strong {
  color: red;
}

types div.menu-bar li:hover > ul {
  display: block;
}
```

```css
div > form {
  display: inline-block;
}

.first {
  margin-top: 10px;
  color: #999;
}

#second {
  padding: 10px 30px;
  text-align: center;
  border-radius: 5px;
  border: 1px solid #ccc;
}
```

CSS Selectors

Introduction to Frontend Engineering

- Type selector - p { }
- id selector - #id { }
- class selector - .class {}
- universal selector - * {}
- descendant selector - div p {}
- child combinator - ul > li {}
- adjacent sibling - div + p {}
- general sibling - div ~ p {}
- attribute selector img[src]
- pseudo class :class
- pseudo element ::first

CSS - Specificity, Inheritance and Cascading

Rules to Note on Specificity

- Universal selector (*), combinators (+, >, ~, ") and negation pseudo-class (:not()) have no effect on specificity. (The selectors declared inside :not() do, however.)
- Inline styles added to an element (e.g., style="font-weight:bold") always overwrite any styles in external stylesheets, and thus can be thought of as having the highest specificity.
- ID selectors are highly specific.
- Class selectors, attributes selectors and pseudo-classes are less specific compared to ID selectors and Inline styles.
- Type selectors and pseudo-elements.

More on CSS Specificity

The amount of specificity a selector has is measured using four different values (or components), which can be thought of as thousands, hundreds, tens and ones — four single digits in four columns:

1. Thousands: Score one in this column if the declaration is inside a style attribute (such declarations don't have selectors, so their specificity is always simply 1000.) Otherwise 0.
2. Hundreds: Score one in this column for each ID selector contained inside the overall selector.
3. Tens: Score one in this column for each class selector, attribute selector, or pseudo-class contained inside the overall selector.
4. Ones: Score one in this column for each element selector or pseudo-element contained inside the overall selector.

The image in the next slide shows a few isolated examples to get you in the mood. Try going through these, and making sure you understand why they have the specificity that we have given them.

More on CSS Specificity

Selector	Thousands	Hundreds	Tens	Ones	Total specificity
`h1`	0	0	0	1	0001
`#identifier`	0	1	0	0	0100
`h1 + p::first-letter`	0	0	0	3	0003
`li > a[href*="en-US"] > .inline-warning`	0	0	2	2	0022
No selector, with a rule inside an element's `style` attribute	1	0	0	0	1000

CSS Cascade

The cascade is an algorithm that defines how to combine property values originating from different sources. It lies at the core of CSS, as emphasized by the name: Cascading Style Sheets.

The CSS cascade algorithm's job is to select CSS declarations in order to determine the correct values for CSS properties.

The cascading algorithm determines how to find the value to apply for each property for each document element:

1. It first filters all the rules from the different sources to keep only the rules that apply to a given element. That means rules whose selector matches the given element and which are part of an appropriate media at-rule.

1. Then it sorts these rules according to their importance, that is, whether or not they are followed by !important, and by their origin. The cascade is in ascending order, which means that !important values from a user-defined style sheet have precedence over normal values originated from a user-agent stylesheet.

1. In case of equality, the specificity of a value is considered to choose one or the other.

Adding CSS to HTML

Introduction to Frontend Engineering

3 Ways To Add CSS To HTML

Inline Styles

Uses the style attribute, comes with limitations.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <title>Getting Started with CSS</title>
</head>
<body>
  <p style="color: red;">This is red text</p>
</body>
</html>
```

Internal Styles

Uses the <style> tag, should be used carefully.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <title>Getting Started with CSS</title>
  <style type="text/css">
    p {
      color: red;
    }
  </style>
</head>
<body>
  <p>This is red text</p>
</body>
</html>
```

3 Ways To Add CSS To HTML

External Styles

Allows writing of css rules in an external stylesheet using the <link> tag.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="
width=device-width, initial-scale=1.0">
  <title>Getting Started with CSS</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <p>This is red text</p>
</body>
</html>
```

```
p {
  color: red;
}

#id {
  color: blue;
}

.class {
  color: green;
}
```

Styling HTML Pages

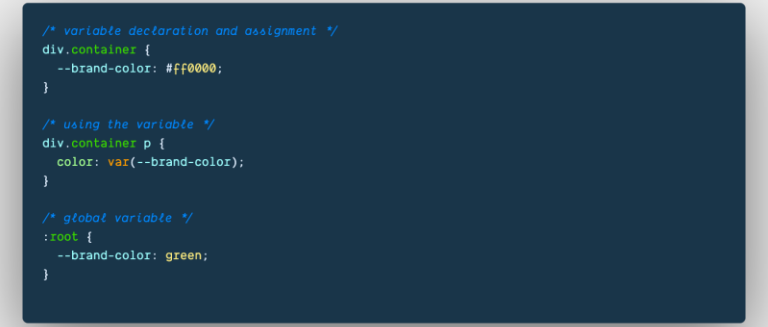
Introduction to Frontend Engineering

Lets STYLE our HTML page

Let’s get our hands dirty with code.

Introduction to Frontend Engineering

CSS Variables and



```
/* variable declaration and assignment */
div.container {
  --brand-color: #ff0000;
}

/* using the variable */
div.container p {
  color: var(--brand-color);
}

/* global variable */
:root {
  --brand-color: green;
}
```

Introduction to Frontend Engineering

CSS Selectors Game

Check out CSS Diner - https://flukeout.github.io/

Introduction to Frontend Engineering

Emmets

Introduction to Frontend Engineering

Writing CSS faster with Emmets

- CSS Abbreviations
  - Value alias
  - Color values
  - Unit less values
- Fuzzy search
- Writing gradient with the help of emmets
- Let stop here for now. For more information check the documentation.

Introduction to Frontend Engineering

Thank You!