www.chuwaamerica.com

# Chapter 1

Instruction to full-stack engineer & HTML/CSS Basics

*Instructors: Alan and Johnny*

www.chuwaamerica.com

## About Johnny and me(Alan)

I will focus on teaching(domain knowledge)
Johnny will take care of the project session

www.chuwaamerica.com

## We are in the same boat

Goal: Help you guys to launch career as a software engineer in a short period.

Estimated timeline: 2 months training + 2 weeks CV workshop + 1-month guided job hunting

www.chuwaamerica.com

## Please keep them in mind

1. Practice makes perfect
2. Always believe something wonderful is about to happen
3. Never never give up

www.chuwaamerica.com

## Zero Tolerance

Free is not equal to cheap

1. If you don’t submit your homework on time(miss homework deadline)

2. If you don’t complete your project on time(miss project deadline)

www.chuwaamerica.com

## What is the difference between fulltime and contractor interview

Typical fulltime interview
1. Zoom screen(1 hours)
2. Final round(4 – 6 hours, depends on the different companies )

Typical contractor interview
Phone screen(1 hours)
Final round(1 hour, 2 hours max, focus on coding, no system design and bq round)

www.chuwaamerica.com

## Trust yourself

Believe me and trust yourself, you could ace the contractor interview when you put effort in learning techniques no matter what background you are.

www.chuwaamerica.com

## Have Questions/Concerns ?

Any technical questions
1. Slack: @Alan Tang or @Yangjun Li
2. WeChat: @WindLover or @李逍遥(wechat id: yangjunlichat)

Any questions, feedback, suggestion, concern about our training/OPT/H1b
1. Drop an email to amitha.ng@chuwaamerica.com

www.chuwaamerica.com

## What is a typical day for a software engineer

1. Join the daily stand up meeting(30 minutes)
2. Write high quality code
3. Write unit test to make sure your functionality to be covered by unit test
4. Raise a PR(Pull Request)
5. Change your code based on your coworker’s comments
6. Review other’s PR
7. Merge your PR to the Repo and publish/bump up the version if it is necessary
8. Sync up with other engineers / PMs

www.chuwaamerica.com

## Agile Methodology

1. 2 weeks / Sprint
2. Sprint planning(UXs, PMs, software engineers, team lead, engineering manager)
3. Daily stand up
4. Additional meetings for syncing up/clarification if it is necessary
5. Sprint retro
6. Burndown chart will be used by your manager to monitor the whole process
7. Jira will be used in tracking your process

## 1.1 What is full stack engineer ?

A full stack engineer is an engineer who works with both the front and back ends of a website or application.

## 1.2 What is the difference between Front End and Back End ?

**Front-end developer:**

Utilize the data coming from the back-end to implement a web application

**Back-end developer:**

Provide data to front end via APIs (Application programming interface)

## 1.3 What skills do I need to become a full stack developer ?

1. HTML, CSS, JavaScript
2. Ideally, one or more third-party library like React or Angular
3. Programming languages and libraries for back end like Node JS and Java
4. Experience with databases, like MongoDB, Oracle, SQL, MySQL
5. Version control like Git
6. Knowledge of security concerns and best practices

Ideally, some knowledge of web or visual design

7. experience best practices

## 1.4 Techniques we should focus on learning for job searching

1. HTML, CSS, JavaScript
2. React
3. Node JS
4. MongoDB

## 1.5 Group the technologies

**Front end**

1. HTML
2. CSS
3. JavaScript
4. React(a JavaScript Library)

**Back end**

1. Node JS/ExpressJS
2. MongoDB

## 1.6 Prioritize the technologies in list

1. HTML -> (P0)
2. CSS -> (P1)
3. JavaScript -> (P0)
4. React -> (P0)
5. Node JS -> (P2)
6. MongoDB -> (P2)

TIPS: P0 is the highest priority and P4 is the least priority

www.chuwaamerica.com

## 1.7 The Three Musketeers in Front End Engineering

1. HTML
2. CSS
3. JavaScript

HTML provides the basic structure of sites (Body)

CSS is used to control presentation, formatting, and layout (Face)

JavaScript is used to control the behavior of different elements (Brain)

www.chuwaamerica.com

## 1.8 Some essential tools and online sandbox

a workman must first sharpen his tools if he is to do his work well

工欲善其事，必先利其器 -- 论语·卫灵公

1. Visual Studio Code
https://code.visualstudio.com/

2. NodeJS
https://nodejs.org/en/

3. CodeSandbox
https://codesandbox.io/

www.chuwaamerica.com

## 1.9 Download the VS Code and install the following plugins

1. Prettier - Code formatter
2. Rainbow Brackets
3. Code Runner
4. Open In Default Browser
5. Fira Code(could download and config it after class)

Talk is cheap, show me the code

-- Linus Torvalds

www.chuwaamerica.com

## 1.10 Introduction to HTML

**HTML** (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as :

<head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, < div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, < embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others

## 1.11 Tags

1. The essence of HTML programming is tags

2. A tag is a keyword enclosed by angle brackets ( Example: <div> )

3. There are opening and closing tags for many but not all tags;  (for example: <hr /> )

www.chuwaamerica.com

## 1.12 Structure of a Web Page

1. All Web pages share a common structure

2. All Web pages should contain a pair of <HTML>, <HEAD>, <TITLE>, and <BODY> tags

```
<!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>Document</title>
  </head>
<body>
  <div>
    Test
  <div>
</body>
</html>
```

www.chuwaamerica.com

## 1.13 It is no need to remember all the Tags

**I will walk you through the following tags**

**BTW, please practice those tags after class**

<div>
<input>
<span>
<ol>, <ul>, <li>
<button>
<select> and <option>

## 1.13.1 <div> Tag

- The <div> tag defines a division or a section in an HTML document.
- The <div> tag is easily styled by using the class or id attribute.
- Any sort of content can be put inside the <div> tag!

## 1.13.2 <input> Tag

- The <input> tag specifies an input field where the user can enter data.
- The <input> element is the most important form element.
- The <input> element can be displayed in several ways, depending on the type attribute.

FYI: remember the following input types

```
<input type="text">
<input type="number">
```

## 1.13.3 <span> Tag

- The <span> tag is an inline container used to mark up a part of a text, or a part of a document.
- The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
- The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

www.chuwaamerica.com

## 1.13.4 <ol> and <ul> Tag

- The <ol> tag defines an ordered list. An ordered list can be numerical or alphabetical.

- The <ul> tag defines an unordered (bulleted) list.
- Use the <ul> tag together with the <li> tag to create unordered lists.

```
<!– start is an optional attribute -->
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

www.chuwaamerica.com

## 1.13.5 <li> Tag

- The <li> tag defines a list item.
- The <li> tag is used inside ordered lists(<ol>), unordered lists (<ul>), and in menu lists (<menu>).
- In <ul>, the list items will usually be displayed with bullet points.
- In <ol>, the list items will usually be displayed with numbers or letters.

## 1.13.6 <button> Tag

- The <button> tag defines a clickable button.
- Inside a <button> element you can put text (and tags like <i>, <b>, <strong>, <br>, <img>, etc.). That is not possible with a button created with the <input> element!
- <input type="button" value="Click Me">

## 1.13.7 <select> and <option>Tag

- The <select> element is used to create a drop-down list.
- The <select> element is most often used in a form, to collect user input.
- The id attribute is needed to associate the drop-down list with a label.

```
<label for="cars">Choose a car:</label>
 <select name="cars" id="cars">
   <option value="volvo">Volvo</option>
   <option value="saab">Saab</option>
   <option value="opel">Opel</option>
   <option value="audi">Audi</option>
 </select>
```

## 1.14 Nested Tags

Whenever you have HTML tags within other HTML tags, you must close the nearest tag first

Example:

```
<div>
  <button> New Button</button>
 </div>
```

## 1.15 Comment Statements

- Comment statements are notes in the HTML code that explain the important features of the code
- The comments do not appear on the Web page itself but are a useful reference to the author of the page and other programmers
- To create a comment statement, use the <!-- …. --> tags

## 1.16 Introduction to CSS

- CSS is stand for Cascading Style Sheets
- CSS is the language we use to style an HTML document
- CSS describes how HTML elements should be displayed.

## 1.17 CSS Advantages

- **Makes website more flexible**

CSS is reusable
Change stylesheet to change design of many pages

- **Easier to maintain**

Cleaner HTML code
Separates styles from HTML tags and page content
Consistent look across entire website that is easily maintained by changing styles in one place.

## 1.18 Adding style to HTML via CSS

**Three Ways to Insert CSS**

- Internal CSS
- Inline CSS
- External CSS

- **Question: What is the best approach to insert CSS ?**

www.chuwaamerica.com

## 1.18.1 Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

www.chuwaamerica.com

## 1.18.2 Inline CSS

- <h1 style="color:blue;text-align:center;">This is a heading</h1>
- <p style="color:red;">This is a paragraph.</p>

**FYI: inline CSS is not recommended to use**

## 1.18.3 External CSS

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<html>
  <head>
    <link rel=“stylesheet” href=“mystyle.css”>
  </head>
 <body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
 </body>
</html>
```

## 1.19 There are two aspects to adding style to a web page via CSS

- The CSS style "Declaration"
- The CSS style "Selector"
- Example:

```
.testClassName {
        font-size: 10px;
        background-color: #fff;
        color: #222;
        margin: 20px;
 }
```

## 1.20 The CSS selectors you need know/remember

- The CSS element Selector
- The CSS id Selector
- The CSS class Selector(most common)

## 1.21 The Examples for element, id and class Selector

-
  ```
  p {
    text-align: center;
    color: red;
  }
  ```

-
  ```
  #para1 {
    text-align: center;
    color: red;
  }
  ```

-
  ```
  .oneClassName {
    text-align: center;
    color: red;
  }
  ```

## 1.21.2 grouping selectors

- You can apply the same declaration to a group of selectors by listing all of the desired selector names separated by commas.

- Example: h1, h2, h3, h4, h5, h6, .className, #id { color: red; font-family:sans-serif }

## 1.21 descendant selector and child selector

Example:

-
  ```
  <div class="outer">
   <div class="inner">
       <span>inner</span>
   </div>
  </div>
  ```

**What happen when we applied the CSS**

```
.outer > span { background-color: grey;  }
.inner > span { background-color: grey;  }
.outer  span { background-color: grey;  }
```

www.chuwaamerica.com

## 1.22 Specificity in the CSS

Specificity is the means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied. Specificity is based on the matching rules which are composed of different sorts of CSS selectors.

www.chuwaamerica.com

## 1.23.1 What is the background color for this div ?

```
<div id="testId"></div>
```

```
<style>
#testId {
  border: 1px solid black;
  width: 100px;
  height: 200px;
  background-color: red;
}

#testId {
  border: 1px solid black;
  width: 100px;
  height: 100px;
  background-color: green;
}
</style>
```

## 1.23.2 Answer

The color should be green

Equal specificity: the latest rule counts - If the same rule is written twice into the external style sheet, then the lower rule in the style sheet is closer to the element to be styled, and therefore will be applied:

www.chuwaamerica.com

## 1.23.3 What is the background color for this h1 ?

```
<h1 id=”content"></h1>
```

From external CSS file:

```
#content h1 {background-
color: red;}
```

In HTML file:

```
<style>
#content h1 {
  background-color: yellow;
}
</style>
```

## 1.23.4 Answer

The color should be yellow

Contextual selectors are more specific than a single element selector.

The embedded style sheet is closer to the element to be styled.

FYI: 内外有别

## 1.24 !important could override the CSS style

In fact, if you use the !important rule, it will override ALL previous styling rules for that specific property on that element!

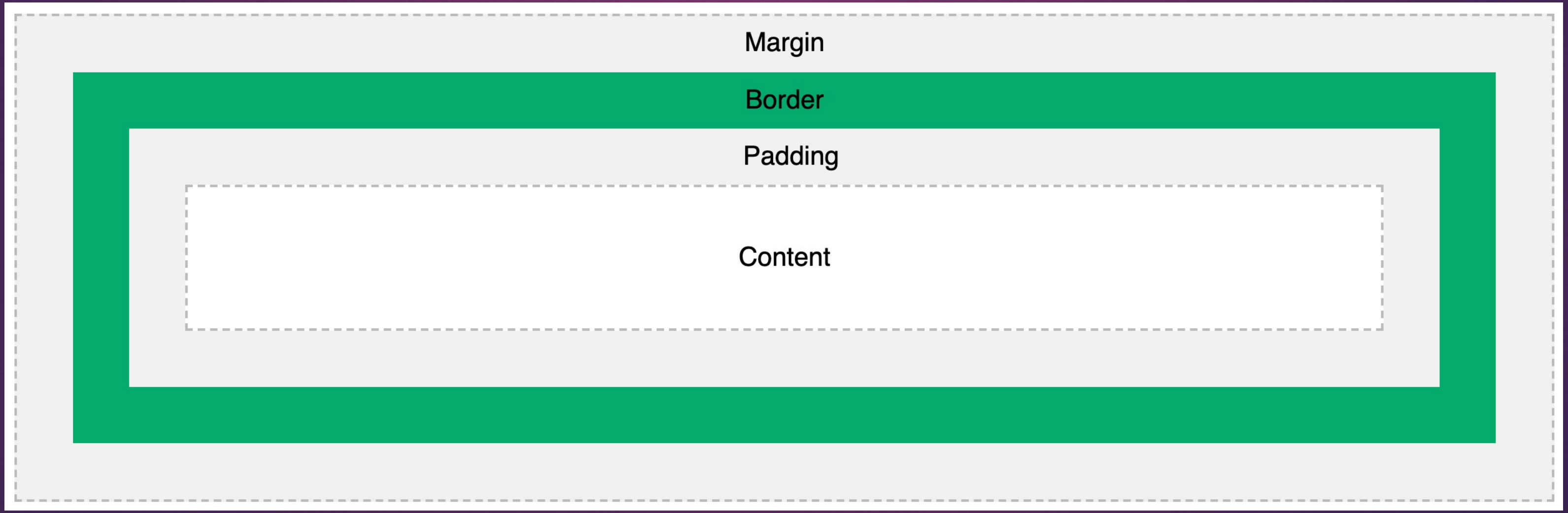
FYI: It is not recommended to use !important at all.

The only way to override an !important rule is to include another !important rule on a declaration with the same (or higher) specificity in the source code

www.chuwaamerica.com

## 1.25.1 Intro to CSS box model

- In CSS, the term "box model" is used when talking about design and layout.



## 1.25.2 element in box model

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

## 1.25.3 flexbox

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.
- .container { display: flex; /* or inline-flex */ }
- .container { flex-direction: row | row-reverse | column | column-reverse; }
- .container { justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right ... + safe | unsafe; }
- https://css-tricks.com/snippets/css/a-guide-to-flexbox/

www.chuwaamerica.com

## 1.25.4 Media query

- Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).
- utilize media query to make your app to be responsive
- Example:

```
@media (max-width: 768px) {
          ….your code here
  }
```