# 2022.8.5 Welcome and First Steps

## 4. A High-Level Overview of Web Development

0:29，讨论客户端和服务器、前端和后端开发、静态和动态网站等内容，当然还有关于web开发的核心语言和核心技术

0:51，围绕我们在浏览器中打开网页的过程来解释这一切

4:12，组成网站的文件只是存储在web服务器上，然后按原样发送到浏览器，我们就说我们有一个静态网站，我们稍后讨论什么是动态网站时

4:26，静态网站基本上是一个网站，其中的文件只是从服务器发送到浏览器，所以没有经过任何的改造

10:08，HTML代表名词， CSS代表宾语，JS代表动词

### 浏览器打开网页过程

1. 访问页面时，浏览器向页面所在互联网上的托管服务器发送请求

a) 每个网站都存储在服务器上，服务器时一台连接到互联网的计算机，能够接收这样的请求

2. 当浏览到某个网站时，我们的web浏览器会向存储该网站的服务器发送请求，当服务器收到请求时，它将获取组成网站的所有文件，并将它们发送回浏览器

a) 我们说服务器向浏览器发送响应，其中基本上包含了构成网站的所有文件

b) 从这些文件扩展名中可以看到，这里有一些HTML、CSS和JavaScript代码，这些是任何浏览器可以理解的三种与语言

3. 浏览器从服务响应中接收到这些HTML、CSS和JavaScript文件，它将获取代码并呈现我们试图访问的网站

### 后端开发，动态网站

1. 第一步是向udemy所在的web服务器发送一个请求.com的主机

2. 像udemy的网站为什么和Omnifood静态网站有如此大不同？

a) 因为udemy网站时不断变化的，比如评论、评分、发布课程等功能，都是在不断发生变化的

b) com需要在服务器上运行一个完整的应用程序，它们还需要一个大型数据库，包含所有课程和所有评论，以及网站上显示的所有内容

c) 仅靠前端技术是远远不够的，因此我们还需要使用后端技术，比如node、php、python

2. 这些语言就是从数据库中取出数据，并将这些数据组装成最终的文件，然后将这些文件作为响应发送给浏览器，这整个过程被称为后端开发，基本上是一个网站的无形部分

3. 事实上，你访问undemy，一个新版本的网站将从他们的数据库生成，并发送到您的浏览器，如果udemy是一个静态网站，那么网站文件就已经在服务器上了，就等着别人来访问他们，这就是静态和动态之间的最大区别在静态中，文件已经完成，而在动态中，他们首先需要有服务器上运行的应用程序生成，现在剩下的流程其实和以前一样，这些文件现在可以作为响应发送到浏览器，然后转换为最终的网站，就像我们这里看到的udemy

## 6. Your Very First Webpack!

1．2:19，index.html基本上总是任何网站的入口点

2. 3:25，! + tab = HTML基础结构

## 8. Watch Before You Start!

1. 不要认为，你不适合编程

2. 每个单词都要自己编写，让大脑产生记忆

3. 记录笔记，不太多，起到注释与回忆作用

4. 独立完成挑战，没有思路是正常的

5. 勤学多练，引申出不同的例子去学习制作

# 2022.8.6 HTML基本原则

## 10. Introduction to HTML

0:23，HTML代表超文本标记语言，HTML是与CSS和JavaScript沿着的核心web技术之一，我们web开发人员可以用它来构造和描述任何网页的整个内容

0) 注意是标记语言，HTML实际不是一种编程语言

1:58，HTML的本质以及我们可以用它做什么

### HTML Element Detailed Introduction

1. 开始标记：元素的名称，包含在其中

2. 内容，有内容(文本)，包含子元素，无内容(图像)

3. 关闭标记，有内容或包含子元素时，关闭标记，没有内容时，将被省略

## 11.HTML Document Structure

1. <!DOCTYPE html> 基本上告诉浏览器这个文档使用HTML，每个HTML文档总是需要以这些所谓的doc类型开始

2. <head>这个头将包含页面标题，一些关于页面的附加信息，链接到CSS文件或其他东西

3. <body>实际上是用于页面上可见的所有元素，因此我们浏览器中看到的所有元素都将始终位于body元素内部

## 12. Text Elements

0:24，通常标题的目的是将大块的文本分成更有逻辑的部分，并基本上为每个部分添加一个标题

### <h1>~<h6> heading instroduction

1. 标题的目的是将大块的文本分成更有逻辑的部分，并基本上为每个部分添加一个标题

2. 标题有层次结构，我们可以在文本中建立层次结构

3. 一个HTML文档页面，最好只有一个H1，当然这不是强制的

4. <b></b>，<strong></strong>，代表文本加粗，b没有任何所谓的语义，strong元素意味着这是一个真正重要的元素，代表这是一个更重要的文本，这就是语义HTML的基本思想

5. <i></i>，<em></em>，代表文本斜体，i只是代表斜体，em代表强调文本的含义

## 13. More Text Elements:Lists

1. ol(ordered list)，代表有序列表

2. li(unordered list)，代表列表项

3. 我们不会有任何文本在主体内浮动而不在元素中，对吧？我们总是在创造一个很好的结构，每个元素都有一定的意义

4. 将所有内容包装在一个元素中的重要性，这个元素实际上赋予了内容一些意义

## 14. Images and Attributes

1. <img>，source

2. alt，可选文本

3. lang，代表语言

4. meta，它是描述文档的一段信息

5. UTF-8，基本上是我们英语中所有简单字符

## 15. Hyperlinks

1. 链接时真正使互联网称为万维网的东西，没有网页之间的链接，就不会有网络

2. 链接分为两类，第一类时指向我们自己网站内其他页面的链接，第二类指向我们网站外部的链接

3. target属性，\_blank代表新建一个标签页打开

## 16. Structuring our page

0:51，通过创建一些容器元素来改变这种情况，这些容器元素将把这些元素组合在一起

1. <nav>(navigation)

2. 将所有不同的部分放在自己的容器中时非常重要的

3. <header>页眉基本上是指网页文档的顶部，更小单元的顶部

4. 如果这些盒子什么都不做，那为什么我们不把它们都叫做导航，或者都叫做标题

5. <footer>，页面的底部

6. 将内容分组为逻辑部分非常重要，这对于语义HTML很重要，对于使用CSS进行样式设置也很重要

## 17. Note on Semantic HTML

1. 当我们谈论语义时，我们的意思是某些元素实际上由一个附加的意义或目的，当我们考虑某个HTML元素时，我们实际上不应该考虑该元素在页面上呈现时的外观

2. 文章、页眉、页脚

3. 搜索引擎优化，这基本上意味着谷歌这样的搜索引擎将能够理解你的内容结构，它们将能够更好地分析你的内容和你的网页是什么，编写语义HTML对于辅助功能也是极其重要的，尤其是对于那些依赖屏幕阅读器来浏览网页的人

4. 当我们考虑HTML时，我们不应该只考虑它在浏览器中的实际外观，而应该更多地考虑这些元素的实际含义和它们所代表的含义

## 18. Installing Additional VS Code Extensions

1. 一个网络服务器在你自己的电脑上运行，实际上在这个端口号上运行这个代码，这个端口基本上就像是web服务器的子地址

## 19. challenge #1 <aside>

1. <aside></aside>，aside元素通常用于补充页面主要部分信息的一些次要信息

2. 如果它是一个列表，我们应该在HTML中标记它