Hyperiondev

TASK

Introduction to Web Development

Visit our website

Introduction

WELCOME TO THE INTRODUCTION TO WEB DEVELOPMENT TASK!

In this task, you'll learn the basics of the Web, including where it originated, and some technical details on how it works. In this course, you will develop a Web application that runs on the Internet, so background knowledge on the Web is crucial to help you understand the context in which you will be working.

You won't learn new HTML in this task - we need to tame our enthusiasm for one task! As you progress from someone who *uses* Web applications, to someone that actually *designs and builds* them, we will peel back the layers of the Internet to show you what really lies beneath the surface.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to Discord at **https://discord.com/invite/hyperdev** where our specialist team is ready to support you.

Our team is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

THE WORLD WIDE WEB

What is the World Wide Web really? Well, it is a global information system, consisting of web pages linked to each other using **hyperlinks**. These are the links that allow us to navigate from one page on a website to another. They also allow us to navigate to pages from other websites from around the world. It is this linking technology that creates the effect of an infinite web of information that we navigate daily.

A note from the
HyperionDev Team

Even though we can't imagine our lives without it, the World Wide Web is a rather recent invention. It was invented by Tim Berners-Lee, an English computer scientist, in 1989. Since its invention, it has expanded exponentially until it has become intricately interwoven into every part of our lives. Our work, entertainment, communication, and even our culture is strongly influenced by this powerful technology.

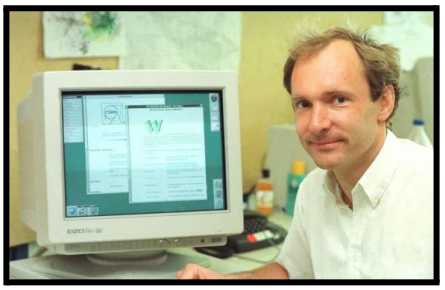


Figure 1: Tim Berners-Lee [1]

CORE COMPONENTS OF THE WORLD WIDE WEB

The World Wide Web consists of three key components. The first is called a **Universal Resource Identifier**, or URL. This is a unique identifier assigned to each page and resource on the web. It allows us to identify and retrieve the specific page, or video/audio/etc file that we want over the Internet.

The second technology is the language used to create web pages. As you know, this language is called **HyperText Markup Language (HTML)**. Unlike other programming languages that allow us to create programs that actually perform

[1] *"History of the Web – World Wide Web Foundation."* ***http://webfoundation.org/about/vision/history-of-the-web/****. Accessed 22 Dec. 2016.*

tasks, this language is used to create the format of the page - what the contents are and how they are placed on the page.

The third technology is the protocol used to request and transfer web pages from one location to another. The Internet is a frantically busy and complex medium of communication and, in order for us to ensure that we transfer things successfully from one location to another, we must have rules of transfer - a protocol - for devices to adhere to. This protocol is called **HyperText Transfer Protocol**, better known as **HTTP**.

HOW THE WEB WORKS

We all access the web using a **client**. A client can be a phone, laptop, desktop, etc - basically anything we can use to access the web. To get to a certain resource (web page etc) on the web, we often open a **browser** and use a **URL** to specify what we want to see.

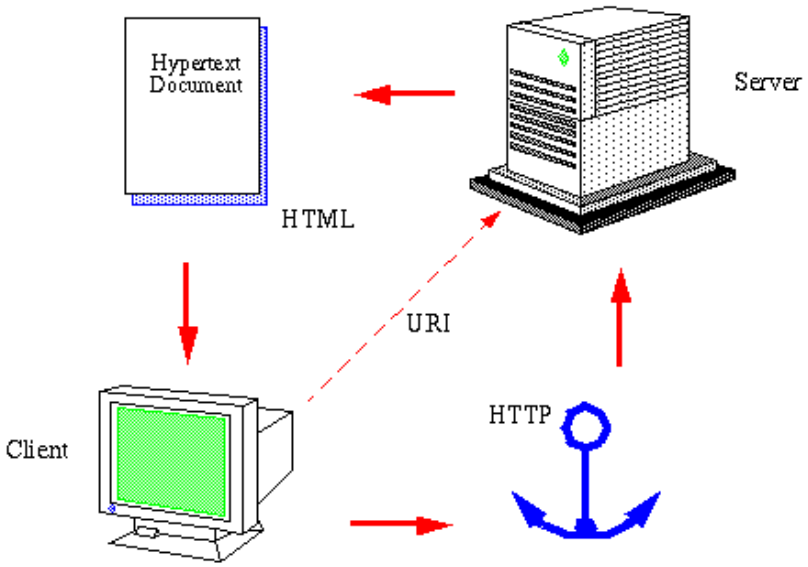
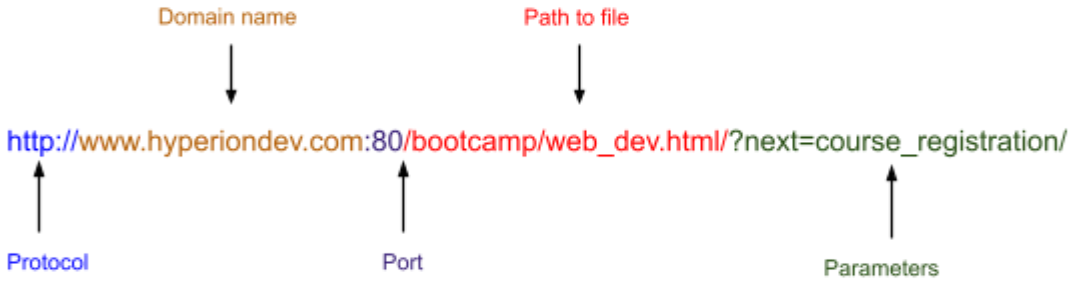


Figure 2: The World Wide Web Model by the W3C:
https://www.w3.org/People/Frystyk/thesis/WWW.html

A **web browser** is an application program on your web-accessing device that allows you to view websites. Chrome, Firefox, and Microsoft Edge are examples of web browsers. A browser takes a URL, which is the address of the website you want to visit, as input.

A URL is a type of **Uniform Resource Identifier**, or **URI**, which identifies the resource by specifying its location on the Web. This URL is a human-friendly address for a particular resource (for example a page called **index.html**) on a particular web server somewhere in the world.

Consider the following fictional URL:



As you can see, the URL contains a lot of information:

1. It identifies the protocol being used to send information. In the example above, the protocol being used is HTTP.
2. It identifies the domain name of the web server on which the resource can be found, e.g. www.hyperiondev.com.
3. It identifies the port on the server. In this example, the port number is given as port 80. In reality, if the default HTTP ports are used (port 80 is the default for HTTP, port 443 for HTTPS), they don't have to be given in the URL.
4. It gives the path to the resource on the web server, e.g. /bootcamp/web_dev.html
5. Parameters can be passed using the URL. Parameters are passed as key-value pairs (?key=value&key2=value2), e.g. ?next=course_registration

A web server is a computer that is set up to store and share many web resources including the HTML files you will create along with any images, videos, and CSS that you add to your HTML page. The function of the browser is to locate the server specified by the URL.

The browser will send an **HTTP** request for the web page to the server that stores it. The server receives requests from all over the world and sends an HTTP response with the requested resource back to the client's browser. After the server sends the page to the browser, the browser is then able to render the page on the screen for the user to view.

Web servers don't just contain static HTML pages. Due to the demand for more dynamic and responsive interaction with web applications, back-end systems have

been created. **Back-end** development has to do with writing code that sits on the server and dynamically builds resources that will be returned to the client. This code can be written using programming languages like JavaScript, Python, etc. Many back-end applications interact with databases that store large amounts of data. You will learn more about back-end development later in this Bootcamp. Your primary focus right now is on **front-end** development. Front-end development is all about using HTML, CSS, and JavaScript to determine what the user will see on their browser.

THE EVOLUTION OF THE WEB

The evolution of the Web can generally be divided into two phases - Web 1.0 and **Web 2.0**. The first generation of websites were basically one-way communication channels. The author would create a web page with some text and images, mostly to communicate information to customers. There was no means of interacting with the web page; all you could do was browse the pages contained in the website. Since there was no means of changing the web page which was being viewed without actually taking it off the internet and editing the HTML, we can think of these web pages as being static. In a Web 1.0 site, the user sits back and consumes the contents. For example, check out this classic Web 1.0 site **here**.

Web 2.0 is all about allowing the user to interact with and contribute content to the website. This is done by providing some means for the user to enter a comment, upload a picture, or "like" something that has been added by someone else. This transition from passive consumption to active contribution to the content of the web page characterises the evolution of the Web from 1.0 to 2.0. Because we can actually change the details of the web pages we access, we can think of these web pages as dynamic. Examples of ways to engage with a dynamic website include:

- Posting a comment on someone's Facebook wall.
- Creating a page on Wikipedia, or editing one that has already been added.
- Creating an investment account online using your bank's website.

Web 2.0 allows for the personalisation of our user experience for any given site. For example, after we log onto a social networking site like Facebook or Twitter, we see information that is specific to our own personal user account. This is possible because, instead of a single pre-written web page being sent to your browser when you request a page, your request is processed by a program behind the scenes. The program then extracts data relating to you, which is then used to personalise your page.

USER ACCOUNTS

This is where the concept of a user account comes in. The program that runs behind the scenes needs to know the person it is dealing with in order to personalise a template specifically for them. Each person that uses a web application has a user account with a unique username that identifies them. When you log in to Gmail, your email address is used to identify you and link you with the personal data that will be presented to you. The user account is a core concept in dynamic web development.

Extra resource

This task has provided a very basic overview of how the web works. There is a lot more to it than that! As you progress through the Bootcamp, your understanding of how the web works will increase. However, if you are interested in a little more detail regarding the fundamental **protocols** and **infrastructure** that make the web work, we highly recommend this additional reading:

1. **https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm**
2. **http://www.cs.kent.edu/~svirdi/Ebook/wdp/ch01.pdf**

SPOT CHECK 1

Let's see what you can remember from this section.

1. What are the three key components of the World Wide Web?
2. What were the main differences between Web 1.0 and Web 2.0?

Compulsory Task

Follow these steps:

- Create a simple web page called **WebFundamentals.html** in which you briefly answer the following questions:
    1. What is Web 3.0 and how is it different from Web 1.0 and Web 2.0? Read **this article** for more information.
    2. What are the functional differences between the front-end of a web application and its back-end? See **here** for more information.
    3. In your own words, explain the process that takes place from when you type a URL into the address bar in your browser until you finally view the page you have requested. Watch **the TED-Ed video** entitled "What is the World Wide Web?" to help you understand this better.

Feel free to use any HTML elements that you have learned about in the previous task to create your web page. Don't worry if this page looks a bit boring at the moment. In the next task, you will learn to add images and links to this task.

If you are having any difficulties, please feel free to contact our specialist team **on Discord** for support.

Completed the task(s)?

Ask an expert to review your work!

Review work

Things to look out for:

1. Make sure that you have installed and set up all programs correctly. You have set up **Dropbox** correctly if you are reading this, but **Visual Studio Code** may not be installed correctly.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.

SPOT CHECK 1 ANSWERS:

1. The Universal Resource Identifier, Hypertext Mark-Up Language (HTML) and HyperText Transfer Protocol (HTTP)
2.
    a. Web 1.0: one-way communication channels where the webpage presented content to the consumer and the consumer had no way of engaging. These were static web pages.
    b. Web 2.0: the consumer is able to interact with the website through communication channels like uploads, 'liking' and leaving comments. These are dynamic web pages. Dynamic web pages can be personalised to each individual consumer through user accounts.