HyperionDev

Web Development Overview

Task

Visit our website

Introduction

In this task, you will learn the basics of the web, including where it originated, and some technical details on how it works.

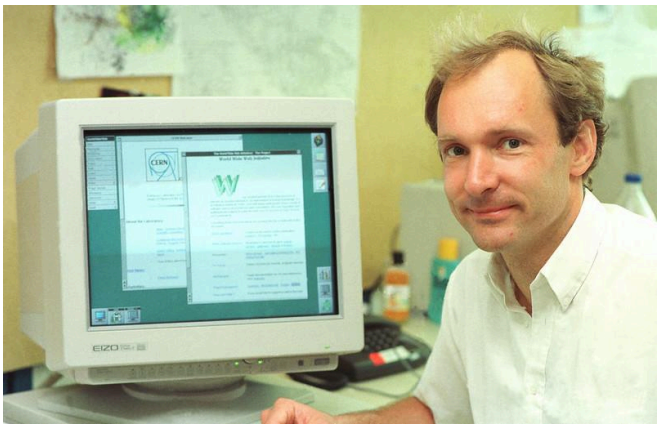
As you progress from someone who uses web applications to someone who actually designs, builds, and advises about them, we will peel back the layers of the Internet to show you what really lies beneath the surface.

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.

Did you know?

Even though we can't imagine our lives without it, the World Wide Web is a rather recent technology in the greater scheme of things! Tim Berners-Lee, an English computer scientist, invented it in 1989. Since then, it has expanded exponentially and is connected to nearly every part of our lives. Our work, entertainment, communication, and even our culture are heavily influenced by the World Wide Web.



Tim Berners-Lee (ITU Pictures, 1996)

Components of the World Wide Web

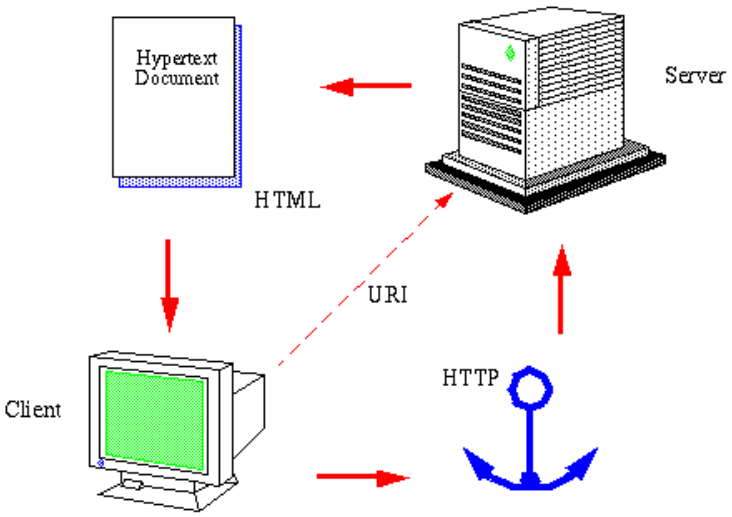
The World Wide Web consists of three key components. The first is called a **uniform resource locator**, or **URL**. This is a unique identifier assigned to each page and resource on the web. It allows us to identify and retrieve the specific page, video, audio, etc. file that we want over the Internet.

The second technology is the language used to create web pages, known as **Hypertext Markup Language** (HTML). Unlike programming languages that allow us to create programs that perform tasks, HTML is used to define the structure and layout of a web page, determining what the content is and how it is arranged 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 the **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.



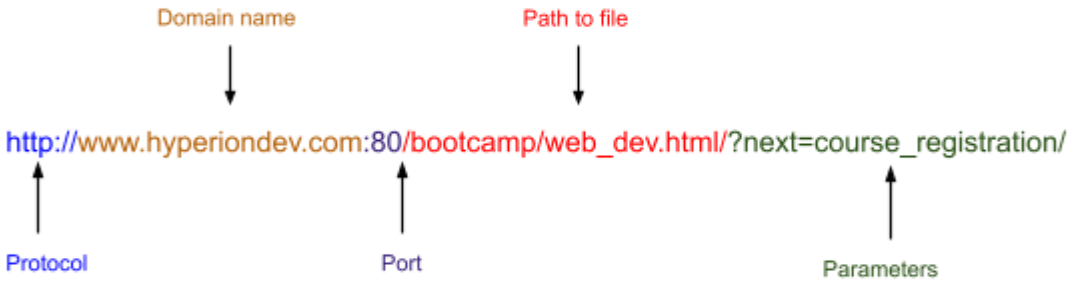
Basic model of the World Wide Web (Frystyk, 1994)

A **web browser** is an application program on your web-accessing device that allows you to view websites. Chrome, Firefox, and Microsoft Edge are all 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 will send 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 F#, Python, etc. Many back-end applications interact with databases that store large amounts of data.

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**.

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, and
- 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 X, we see information that is specific to our 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 who 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's much more to explore beyond this overview, and as you continue in the Bootcamp, you'll have opportunities to deepen your understanding. However, if you are interested in a little more detail on the fundamental protocols and infrastructure that make the web work, we highly recommend these additional readings:

1. **How Does the Internet Work?**
2. **Web Basics and Overview**

Spot check

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

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

Take note

The task below is **auto-graded**. An auto-graded task still counts towards your progression and graduation. Give it your best attempt and submit it when you are ready.

When you select "Request Review", the task is automatically complete, you do not need to wait for it to be reviewed by a mentor.

You will then receive an email with a link to a model answer, as well as an overview of the approach taken to reach this answer.

Take some time to review and compare your work against the model answer. This exercise will help solidify your understanding and provide an opportunity for reflection on how to apply these concepts in future projects.

In the same email, you will also receive a link to a survey, which you can use to self-assess your submission.

Once you've done that, feel free to progress to the next task.

Auto-graded task

Create a document titled **WebFundamentals.txt** 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 **Here's what you should know about Web 3.0** for more information.
2. What are the functional differences between the front end of a web application and its back end? Read **Developer roles: Front-end vs back-end vs full stack** 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 **What is the world wide web?** from TED-Ed to help understand this better.

Please ensure that you place the files for submission inside the task folder and click "Request Review" on the dashboard, even for auto-graded tasks.

Spot check answers

1. Question 1's answer: The uniform resource locator (URL), Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP).
2. Question 2's answer:
    a. Web 1.0: One-way communication channels where the web page presented content to the consumer and the consumer had no way of engaging. These were static web pages.

b. Web 2.0: The consumer can 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.

Share your thoughts

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

Do you think we’ve done a good job or do you think the content of this task, or this course as a whole, can be improved?

Share your thoughts anonymously using this **form**.

Reference list

ITU Pictures. (1994, July 11). *Tim Berners-Lee*. Flickr. **https://www.flickr.com/photos/itupictures/16662336315**

Frystyk, H. (1994, July). *The World-Wide Web*. W3C. **https://www.w3.org/People/Frystyk/thesis/WWW.html**