### **Activity: Understanding the Internet and Hands-On HTML Basics**

**Objective:**By the end of this activity, learners will:

1. Understand basic concepts about how the internet works, including **IP Address**, **DNS**, and **Domain Names**.
2. Experiment with basic HTML tags like <h1>, <p>, <ul>, <li>, <a>, and bonus tags <img> and <ol>.
3. Develop foundational knowledge to build web pages while gaining theoretical insight into how the internet supports these technologies.

**Time:** 1 Hour  
**Structure:**

* **Section 1:** Theory (30 minutes)
* **Section 2:** Hands-On HTML Practice (30 minutes)

### **Section 1: Theory – How Does the Internet Work?**

**Instructions:**Answer the following questions in your own words. Use class notes, discussions, or research online (Google, W3Schools, etc.) to find answers. Write your responses in the space provided.

#### **Part A: Core Internet Concepts**

[How does the INTERNET work? | ICT #2](https://www.youtube.com/watch?v=x3c1ih2NJEg&ab_channel=SabinCivilEngineering)

1. **What is an IP Address?**
   * Explain what an IP Address is and its role in connecting devices over the internet.
   * Example: "Why do websites need IP addresses to work?"

An IP Address is a unique, numerical identifier given to servers, computers, and any device that connects to the internet. IP Addresses are assigned depending on their type (public, private, static, or dynamic), but the Internet Assigned Numbers Authority (IANA) allocates an IP address and its creation. IP Addresses play an important role in transporting data across the internet. Sabin Civil Engineering, in the provided video, compares IP addresses to shipping addresses in that it tells providers and requesters of information on the internet where to retrieve and send the data.

Websites need an IP address to work because without one a customer or user would not be able to find or access the website.

*Additional Source:*

*Fortinet:* [*https://www.fortinet.com/resources/cyberglossary/what-is-ip-address*](https://www.fortinet.com/resources/cyberglossary/what-is-ip-address)

1. **What is DNS (Domain Name System)?**
   * Describe how DNS works and why it’s important.
   * Example: "How does DNS help you visit a website like www.google.com instead of typing an IP address?"

Domain names are assigned to an IP Address to make it easier for humans to remember and search. The DNS contains a ‘list’ of domain names and their corresponding IP addresses. By using the DNS, a browser will be able to find the IP address for a domain name such as google.com and locate and access the appropriate server.

1. **What is a Domain Name?**
   * Define a domain name and explain how it is different from an IP address.
   * Example: "Why do people buy domain names for websites instead of using IP addresses?"

A domain name is assigned to an IP Address to make it easier for humans to remember and search. Someone would buy a domain name for a website because it is more marketable than an IP Address and for a company it is part of its branding. Additionally, IP Addresses may change over time and updating this change in a DNS behind the scenes is much easier than advertising and trying to let everyone who might use a website know that the IP Address has changed.

*Additional Source:*

*Wordpress.com: https://wordpress.com/blog/2025/03/21/what-is-a-domain-name/*

1. **How Does the Internet Work?**
   * Summarize how information travels from your computer to a website and back.
   * Include terms like **request**, **server**, and **browser**.

A screenshot of a computer

AI-generated content may be incorrect.

#### **Part B: Browsers and HTML**

1. **What is a Web Browser?**
   * Define what a web browser does.
   * Example: "How does a browser help users interact with the internet?"

A browser is software application that allows a user to interact with and retrieve information on the internet. Browsers translate the data and code that are retrieved from servers and display them in a way humans can understand and interact with.

*Additional Source:*

*BrowserStack:* [*What is a Browser? How does it Work? | BrowserStack*](https://www.browserstack.com/guide/what-is-browser)

1. **What is HTML?**
   * Write a short definition of HTML and its role in building web pages.
   * Example: "Why is HTML considered the backbone of web pages?"

Hypertext Markup Language (HTML) is the standard language used to structure the contents of webpages. HTML is used with CSS and JavaScript to make a complete webpage. HTML can be considered the frame that CSS styles and JavaScript makes interactive. Without the frame, there is nothing supporting CSS and JavaScript.

1. **What Happens When You Enter a URL into a Browser?**
   * Describe the steps involved, from entering a URL to seeing the web page.
     1. The browser figures out where to go.
        1. It uses the DNS to identify the IP Address it needs to request access to/data from
     2. The browser uses the IP address to establish and secure a connection with the server.
     3. The browser requests the homepage.
     4. The server responds and sends back a response
     5. The browser translates the server’s response and displays it so the user can understand and interact with it.

*Additional Source:*

*The Dev Learnings:* [*What Really Happens When You Type a URL in Your Browser*](https://thedevlearnings.com/what-really-happens-when-you-type-a-url-in-your-browser/)

### **Section 2: Hands-On HTML Practice**

**Instructions:**

1. Open a text editor (e.g., Notepad, VSCode, Sublime Text).
2. Create a new file and save it as index.html.
3. Add the following **basic structure** to your HTML file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My First Web Page</title>

</head>

<body>

<!-- Write your code here -->

</body>

</html>

1. Complete the tasks below by adding the requested tags inside the <body> section of your HTML file. Save the file and open it in a browser to see the results.

#### **Part A: Basic Tags Practice**

1. **Add a Heading**:
   * Use an <h1> tag to create a large heading with the text "Welcome to My Web Page".
2. **Add a Paragraph**:
   * Use a <p> tag to write a short introduction about yourself.
3. **Create a List**:
   * Use <ul> and <li> tags to create an unordered list of your favorite hobbies.
4. **Add a Link**:
   * Use an <a> tag to add a clickable link to your favorite website (e.g., Google, YouTube).

#### **Part B: Bonus Tags**

1. **Add an Image**:
   * Use an <img> tag to add a picture of your choice. You can use a placeholder image like:  
     https://via.placeholder.com/150
2. **Create an Ordered List**:
   * Use <ol> and <li> tags to create an ordered list of the top three places you’d like to visit.

#### **Questions After Experimenting**

Answer these questions based on your hands-on practice:

1. **What happens if you forget to close a tag like <h1>?**

If the <h1> tag is not closed, the entire webpage will be in the header format.

1. **What is the purpose of the <a> tag’s href attribute?**

The href attribute indicates the link’s destination.

1. **How does saving your file as .html affect how your browser opens it?**

Saving the file as .html allows the browser to open it.

### **Wrap-Up and Submission**

1. Submit your completed theory answers and HTML code to the shared class platform (e.g., Google Drive or email).
2. Be ready to share your web page and explain your experience in class.

**Key Takeaway:**This activity combines theory and practice to help you understand how the internet works and how basic HTML tags create web pages. These concepts are fundamental as you begin your journey into web development! 🚀

## **🔧 Bonus Activity: Build Your Own Mini Web Project**

**Objective:** Apply the theory and HTML skills you’ve learned by building a simple, creative, multi-section web page.

**Time:** 30–45 minutes (Can be done in class or as homework)

### **💡 Overview:**

You’ll create a simple personal or themed webpage with **multiple sections** using HTML. This project helps you explore **semantic HTML tags**, learn about **HTML structure**, and express yourself creatively.

### **🛠 Instructions:**

Open your index.html file (or create a new one if needed) and add the following:

#### **Part A: Use Semantic Tags**

Include **at least three** of the following semantic tags inside your <body>:

* <header> – Add a site title or welcome message
* <nav> – Add a simple navigation with links (they can link to other sections using #)
* <section> – Use this to group related content (e.g., About Me, Hobbies)
* <article> – Create a short blog-style entry about a topic you like
* <footer> – Add your name, email, or a thank-you message

#### **Part B: Style It with Inline CSS (Optional Challenge)**

Add some simple **inline styling** to make your page more attractive:

* Change the color of headings using the style attribute
* Add a background color to a section
* Change the font size of a paragraph

Example:

<h1 style="color: navy;">Welcome to My Web Page</h1>

<p style="font-size: 18px;">I'm excited to learn HTML!</p>

#### **Part C: Add More Content!**

You can enhance your page with:

* An **additional image**
* A **second list** (unordered or ordered)
* More links that use target="\_blank" to open in new tabs

### **🧠 Bonus Questions – Reflect & Discuss:**

1. What did you enjoy most about building your own page?  
   The challenge of making a simple webpage look ok and finding ways to implement some visual techniques I have used before such as transparency (opacity).
2. Which tag was new to you in this activity? How did it work?

The <section> and the <style> tags were new.

The <section> tag grouped similar information together into another ‘container’. Doing this allowed me to apply a ‘style’ to the whole section.

The <style> tag is added before the body of the code and was used to add a background image to the whole page.

1. What would you like to add to your webpage if you had more time or knew more HTML?

I would continue to like to make it look more cohesive, like add a halo or shadow to text to make it stand out a little more from the background. I could have continued to search these effects, but it would be nice to know the best way to add multiple effects before I just throw more in there.

1. Why do you think semantic tags (like <header>, <footer>, <section>) are helpful for websites?

I think semantic tags are helpful in keeping code organized and for helping the browser read it and know what to do.

### **📦 Submission:**

* Save your updated index.html file.
* Submit it with your answers to the bonus questions.
* Be ready to present your mini-project and walk through your code.

### **🧭 Key Takeaway:**

Creating a full mini-page helps you see how different parts of a webpage come together. You’ve now worked with both **structure** and **style**, setting a strong foundation for the future web.