Part One - HTML Basics

01 – Concepts Related to Webpages

1. **URL (Uniform Resource Locator)**: The address we enter in the browser.
2. **Webpage**: Each individual page displayed by the browser.
3. **Website**: A collection of multiple webpages forming a single site.
4. **Web Standards**:

02 – A Taste of HTML

1. Step 1: Right-click => New => Text Document => Enter the following content and save.

```
<details>
    <summary>The secret to mastering programming:</summary>
    Subscribe to Codex and Start Coding Today!
</details>
```

2. Step 2: Change the file extension to `.html`, then double-click to open it.

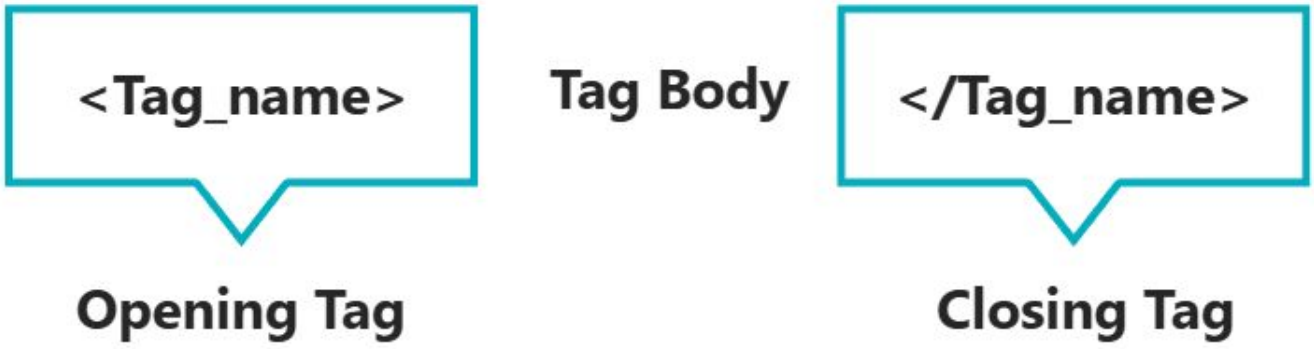
> The file extension `.htm` can also be used here, but the more standard `.html` is recommended.

3. Programmers write **source code**, which is handed over to the browser for rendering.
4. To view a webpage's **source code** using the browser, follow these steps:

> Right-click on a blank area of the webpage => Select **View Page Source**.

03 – HTML Tags

1. **Tags**, also known as **elements**, are the basic building blocks of HTML.
2. Tags are categorized into: **paired tags** and **self-closing tags** (the vast majority are paired tags).
3. Tag names are **case-insensitive**, but lowercase is recommended as it is more standard.
4. **Paired tags**:



Example Code:

```
<details>
    <summary>The secret to mastering programming:</summary>
    Subscribe to Codex and Start Coding Today!
</details>
```

5. **Self-closing Tags**:

Note: The / in self-closing tags can be omitted

Example Code:

```
<input>
```

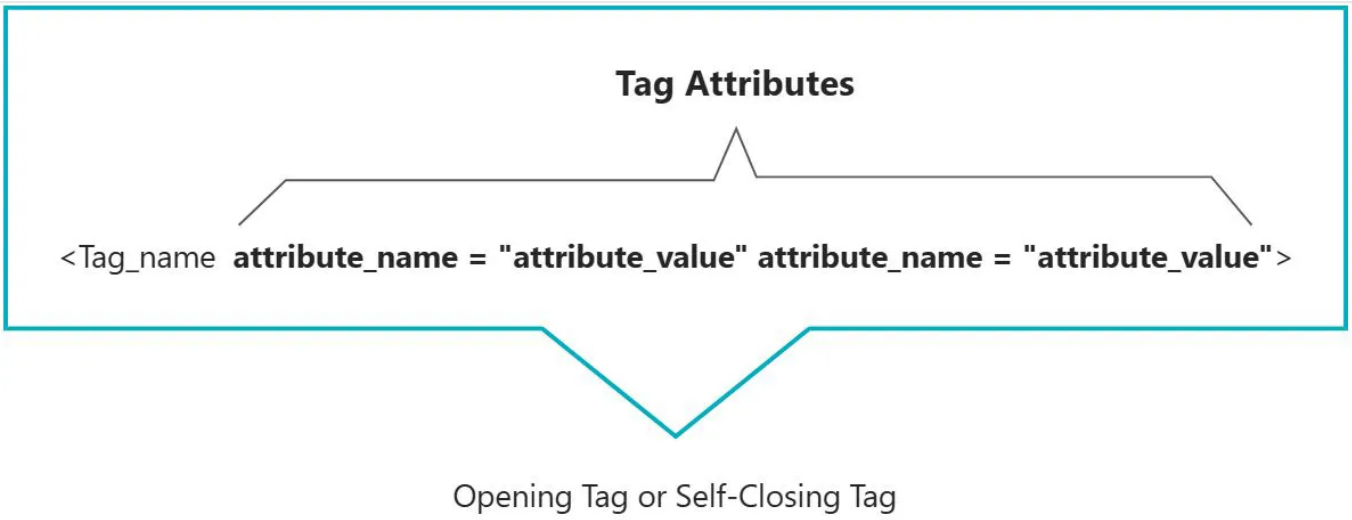
6. The relationships between tags: **parallel relationship** and **nested relationship**. Indentation can be done using the **Tab key**:

```
<details>
  <summary>The secret to mastering programming:</summary>
  Subscribe to Codex and Start Coding Today!
  <input>
</details>
<input>
```

04 – HTML Tag Attributes

---

1. Used to provide **additional information** to tags.
2. Can be written in the **opening tag** or **self-closing tag**, as follows:




```
// Example Code:
<progress value="45" max="100">
  45%
</progress>;
<input type="password" />;
```

3. Some special attributes do not have an attribute name, only an attribute value, such as:

```
<input disabled />
```

4. Points to Note:

1. Different tags have different attributes; there are also some common attributes (which can be written in any tag, to be summarized in detail later).
2. Attribute names and values should not be written arbitrarily, as they are defined by W3C.
3. Attribute names and values are case-insensitive, but lowercase is recommended.
4. Double quotes can be written as single quotes, or even omitted, but double quotes are still recommended.
5. Do not use the same attribute name multiple times within a tag, as the later ones will take effect. For example:

```
<input type="text" type="password" />
```

05 – Basic Structure of HTML

---

1. How to view the specific code of a section in a webpage? — Right-click and select "**Inspect**".
2. Difference between "Inspect" and "View Page Source":

View Page Source shows: The source code written by the programmer.

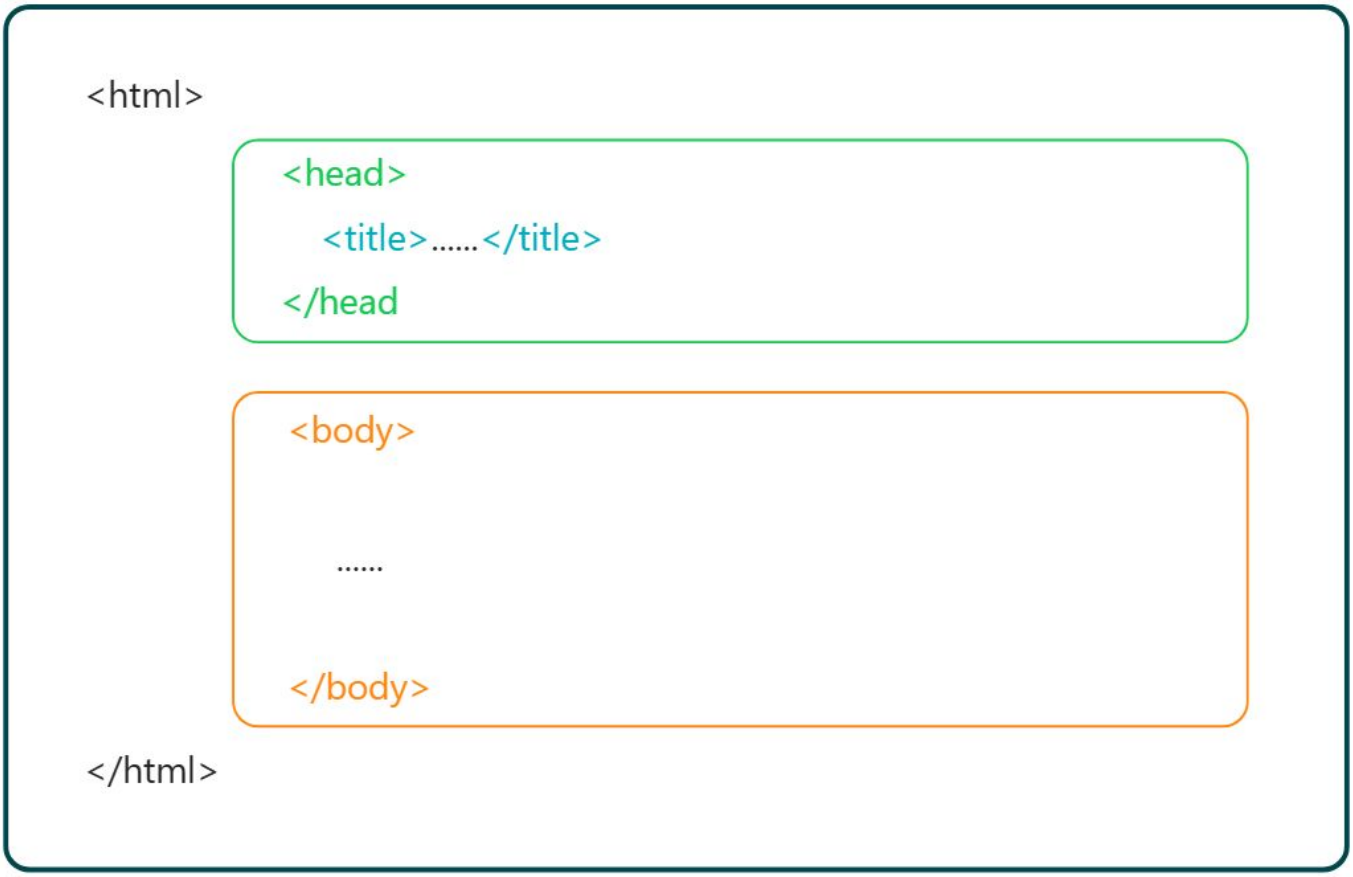
Inspect shows: The source code after being "processed" by the browser.

Note: In daily development, **"Inspect"** is used the most.

3. The **basic structure** of a webpage is as follows:

1. The content you want to display on the webpage is written inside the `<body>` tag.
2. The content inside the `<head>` tag does not appear on the webpage.
3. The `<title>` tag inside the `<head>` tag can specify the title of the webpage.

4. Illustration:



5. Code:

```
<html>
  <head>
    <title>Website Title</title>
  </head>
  <body>
    ......
  </body>
</html>
```

06 – VSCode

1. Quick Introduction
2. Two methods to open a folder in VSCode
3. Adjust the font
4. Set a theme
5. Install the Live Server extension
    a. Makes it more convenient to open webpages

b. Provides a way to view webpages that how they would look after deployment

c. Automatically reloads the page when changes are made to the code.

d. Configure VSCode's auto-save settings based on your preferences

Note 1: Ensure you open a folder in VSCode; otherwise, the Live Server extension will not work properly!

Note 2: The opened webpage must follow the standard HTML structure; otherwise, auto-refresh will not function!

07 – Doctype

---

1. **Purpose**: To inform the browser about the version of the HTML.
2. **Syntax**:

- **Old Syntax**: The syntax depends on the version of HTML being used, with multiple variations.

For details, refer to the W3C website – Document Declaration (for understanding only, no need to memorize!).

- **New Syntax**: Everything has become simpler! The W3C recommends using the HTML5 syntax.

```
<!DOCTYPE html>
```

3. The doctype declaration must be placed on the very first line of the webpage and should be outside the `<html>` tag.

08 – HTML Character Encoding

---

1. **Computer Operations on Data**:

- During storage, data is **encoded**.
- During retrieval, data is **decoded**.

2. **Encoding and decoding follow certain standards — called character sets.**

3. **Common character sets** (for reference):

- **ASCII**: Includes uppercase letters, lowercase letters, numbers, and some symbols, totaling 128 characters.
- **ISO 8859-1**: Expands on ASCII by including some Greek characters, totaling 256 characters.
- **GB2312**: Further expands to include 6,763 commonly used Chinese characters and 682 symbols.
- **GBK**: Includes more than 20,000 Chinese characters and symbols, supporting Traditional Chinese.
- **UTF-8**: Contains all characters and symbols of all languages in the world — widely used.

4. **Principles of Usage**:

Principle 1: Ensure the appropriate character encoding is used during storage.
Otherwise: Data cannot be stored, and it may be lost!

Principle 2: Use the same encoding for retrieval as was used for storage.
Otherwise: Data corruption or garbled text may occur!

Example

Below text includes: English, Hindi, Japanese and Chinese.

I love you

मुझे तुमसे प्यार है

愛してます

我爱你

If ISO8859-1 encoding is used for storage, issues will arise the moment data is stored because ISO8859-1 only supports English!

To ensure all input can be stored and read properly, **UTF-8 encoding** is now almost universally adopted.

When writing HTML files, it's standard practice to use **UTF-8 encoding** for consistency.

5. Summary:

- When writing code, always use UTF-8 encoding (the safest choice).
- To ensure the browser renders HTML files correctly, you can specify the character encoding using the `<meta>` tag with the `charset` attribute.

```
<head>
<meta charset="UTF-8"/>
</head>
```