Mã học phần: INT1334

Lập Trình Web

Giảng viên: Ths Trần Hoàng Nam

Lý thuyết: 30 tiết
Bài tập/Thảo luận: 08 tiết
Thực hành: 06 tiết
Tự học : 01 tiết

Goals

1. Nắm được khái niệm về kiến trúc và phát triển ứng dụng Web. Ngôn ngữ nền tảng.
2. Phát triển ứng dụng Web phía front-end với ReactJS.
3. Phát triển ứng dụng Web phía back-end với nền tảng NodesJS/ExpressJS.
4. Các kỹ thuật lập trình web trong thực tế.

Quy định

1 Điểm quá trình(30%)

- Chuyên cần(10%)
- Kiểm tra giữa kỳ(10%)
- Bài tập trên lớp(10%)

2 Điểm tổng kết(70%): Báo cáo dự án

- 1 nhóm 4 sinh viên
- Đăng ký đề tài
- Gửi báo cáo mỗi tuần
- Báo cáo dự án(thuyết trình)

Outline

1 - Tổng quan về lập trình web
- Ngôn ngữ HTML.

2 - Cascading Style Sheet(CSS)
- Lập trình JavaScript cơ bản

3 - Lập trình JavaScript nâng cao
- Document Object Model (DOM).

4 - Front-end programing
- ReactJS Component

Outline

5
- React State & Hooks
- React Form & Data Handling

6
- Lập trình backend
- Lập trình CSDL

7
- Xác thực và bảo mật
- Một số vấn đề nâng cao trong phát triển ứng dụng web

8
- Front-end programing
- ReactJS Component

9
- Hướng dẫn làm đồ án

Website categories - Projects

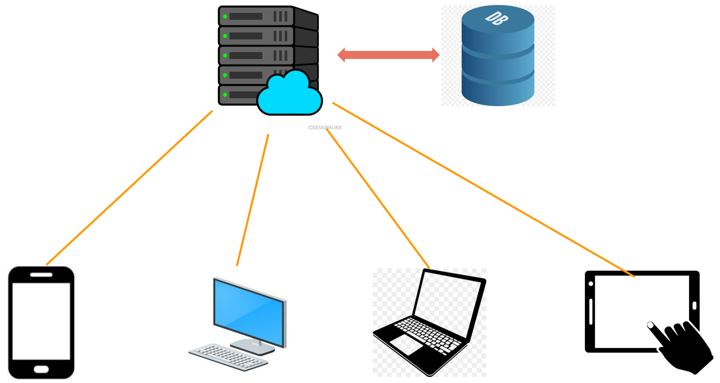
1. Personal Websites
   - Purpose: Showcase individual portfolios, blogs, or personal interests.
   - Examples: Blogs, vlogs, online resumes, or hobby-related sites.
2. Business Websites
   - Purpose: Represent a company, brand, or organization online.
   - Examples: Corporate websites, small business websites.
3. E-commerce Websites
   - Purpose: Enable online buying and selling of products or services.
   - Examples: Amazon, Shopify stores, Etsy.
4. Portfolio Websites
   - Purpose: Showcase creative or professional work.
   - Examples: Photographer, designer, or artist portfolios.
5. Educational Websites
   - Purpose: Provide educational resources, courses, or information.
   - Examples: Coursera, Khan Academy, university websites.
6. Entertainment Websites
   - Purpose: Deliver content for entertainment or leisure.
   - Examples: Streaming platforms (Netflix), gaming sites, or meme hubs.
7. News and Media Websites
   - Purpose: Share current events, news articles, and updates.
   - Examples: BBC, CNN, local news websites.
8. Community and Forum Websites
   - Purpose: Foster discussions and build online communities.
   - Examples: Reddit, Quora, niche forums.
9. Social Media Websites
   - Purpose: Enable social networking and interaction.
   - Examples: Facebook, Instagram, Twitter.
10. Government Websites
    - Purpose: Provide official information and services from governmental bodies.
    - Examples: National or state government portals, visa application sites.
11. Non-Profit or Charity Websites
    - Purpose: Promote charitable activities and gather donations.
    - Examples: WWF, UNICEF.
12. Health and Wellness Websites
    - Purpose: Provide medical information, advice, or services.
    - Examples: WebMD, health clinics, and fitness blogs.
13. Portfolio or CV Websites
    - Purpose: Highlight skills and achievements for professionals.
    - Examples: LinkedIn profiles, personal CV websites.
14. Membership Websites
    - Purpose: Offer content or services to registered users.
    - Examples: Subscription services, exclusive content platforms.
15. Web Applications
    - Purpose: Provide interactive services and tools online.
    - Examples: Google Docs, Trello, Canva.

Tổng quan về lập trình web

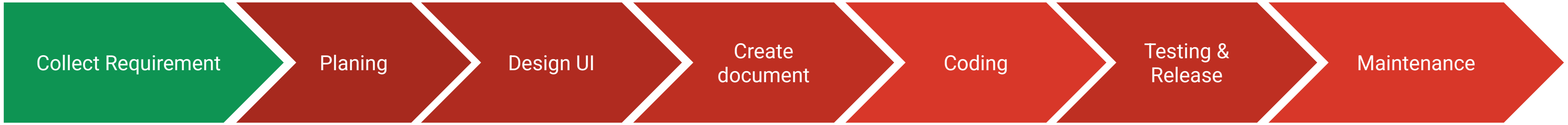
- Tạo ra các trang web và ứng dụng web bằng cách sử dụng các ngôn ngữ lập trình và các công cụ phát triển web.

  Front-end: Đây là phần của trang web mà người dùng cuối thấy và tương tác. Front-end thường bao gồm HTML, CSS và JavaScript.

  Back-end: Xử lý dữ liệu và logic của ứng dụng. Back-end thường được viết bằng các ngôn ngữ lập trình như PHP, Java, Python, Ruby, C# và nhiều ngôn ngữ khác.

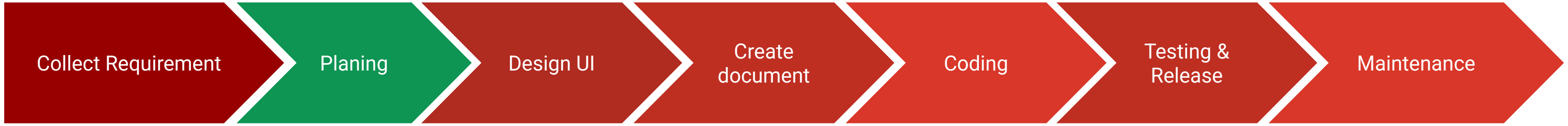


Tổng quan - quy trình phát triển 1 website

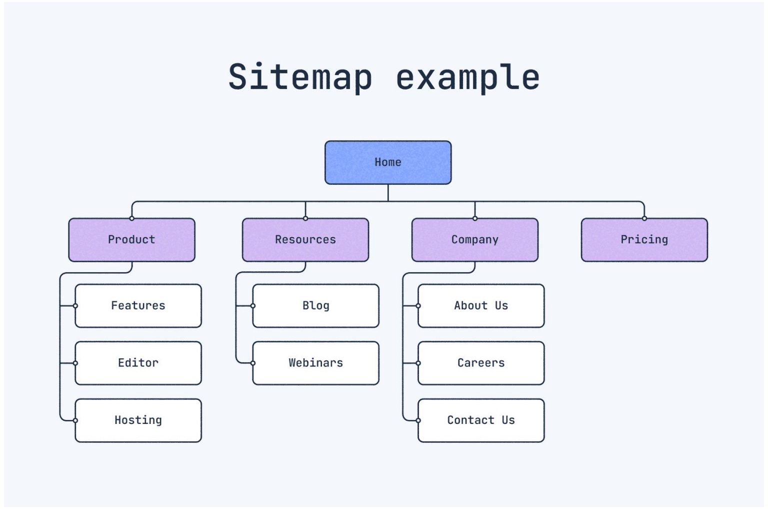


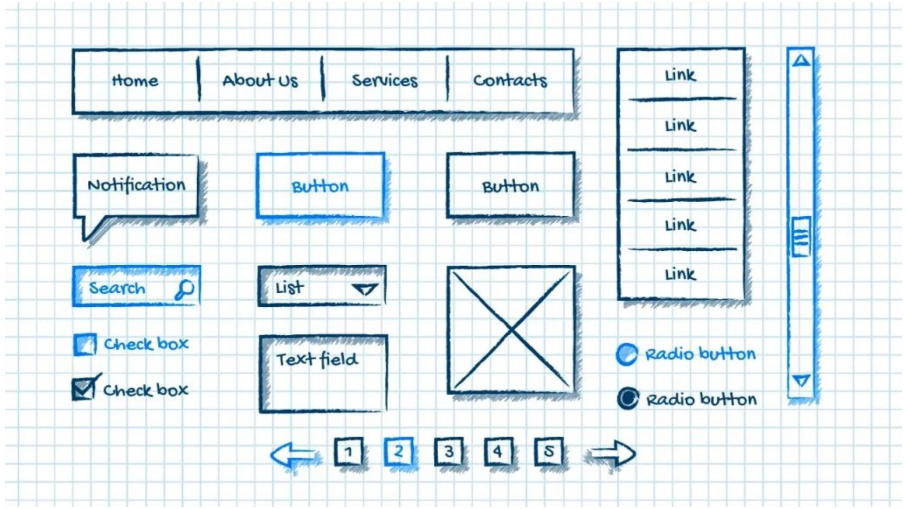
- Collect information from customer
- The purpose of website
- The user set
- Features
- Logo, color, images, article…

Tổng quan - quy trình phát triển 1 website

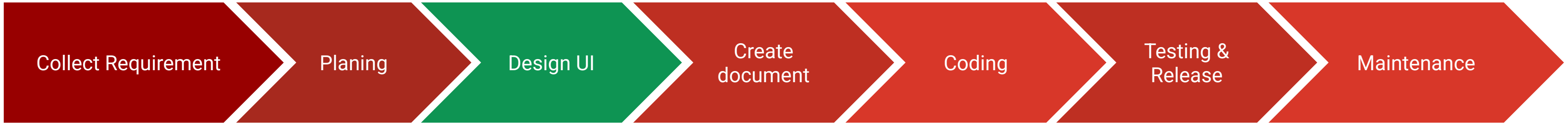


Sitemap & Wireframe

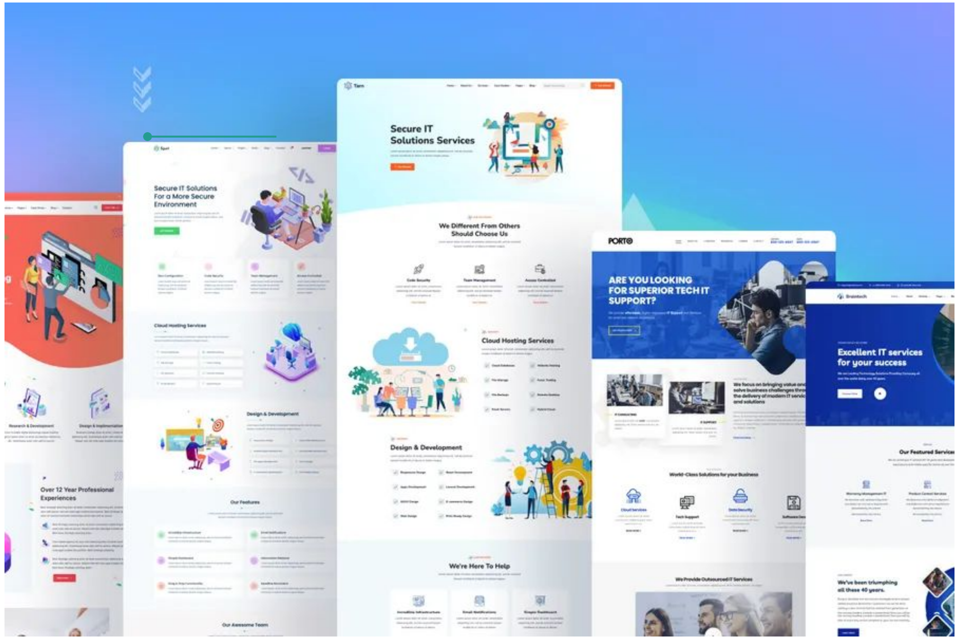


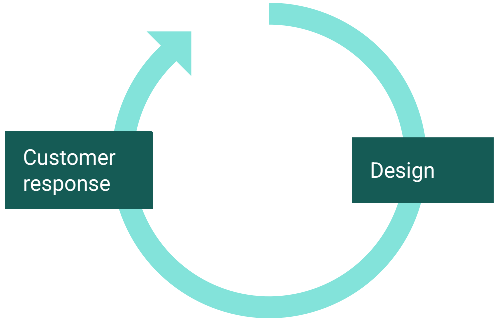


Tổng quan - quy trình phát triển 1 website

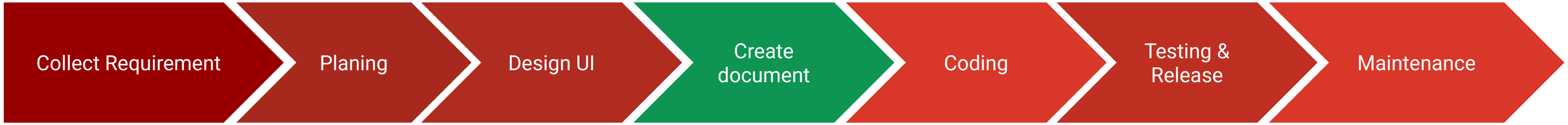


- Tool: photoshop
- Color
- Images
- Videos
- Font
- Update UI due to customer response





Tổng quan - quy trình phát triển 1 website



Create SRS (Software Requirement Specification)

- Include all feature spec (help customer understand about system)
- Important for dev team (system analyst, business, analyst, code) and Tester
- Base on SRS => task scope => Time to finish and cost

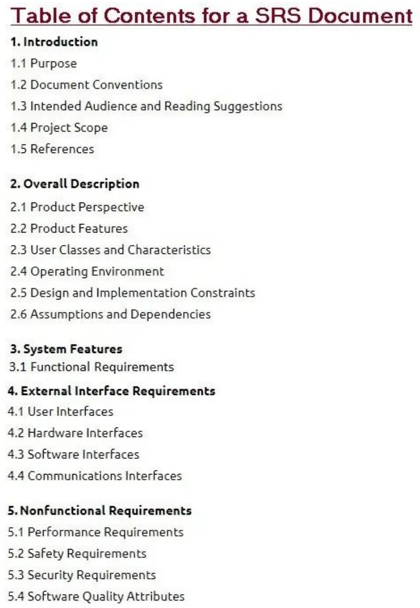


Table of Contents for a SRS Document

1. Introduction
1.1 Purpose
1.2 Document Conventions
1.3 Intended Audience and Reading Suggestions
1.4 Project Scope
1.5 References

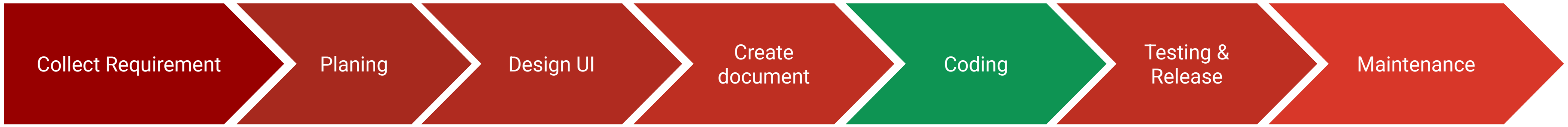
2. Overall Description
2.1 Product Perspective
2.2 Product Features
2.3 User Classes and Characteristics
2.4 Operating Environment
2.5 Design and Implementation Constraints
2.6 Assumptions and Dependencies

3. System Features
3.1 Functional Requirements

4. External Interface Requirements
4.1 User Interfaces
4.2 Hardware Interfaces
4.3 Software Interfaces
4.4 Communications Interfaces

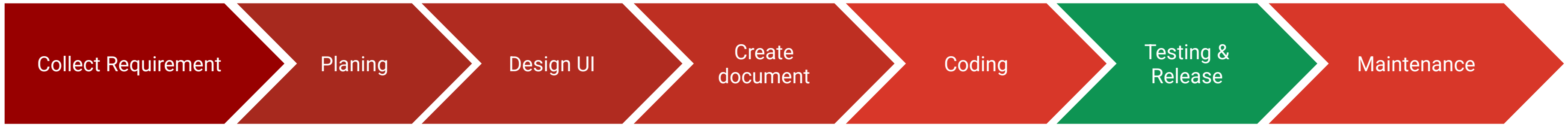
5. Nonfunctional Requirements
5.1 Performance Requirements
5.2 Safety Requirements
5.3 Security Requirements
5.4 Software Quality Attributes

Tổng quan - quy trình phát triển 1 website



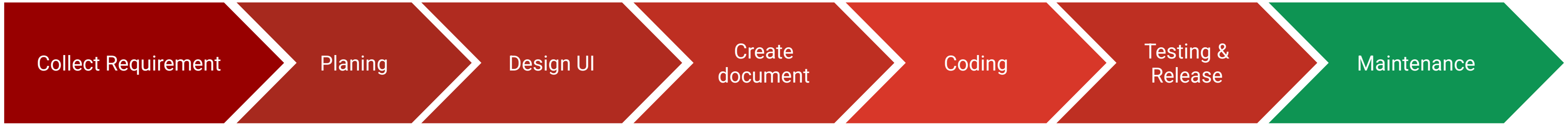
- Choose source code
- Programing language
- Framework
- Unit test

Tổng quan - quy trình phát triển 1 website



1. Testing of function (Functional testing)
2. Testing of software product characteristics (Non - Functional testing)
3. Testing of software structure/architecture (Structural testing)
4. Testing related to changes (Confirmation and regression testing)

Tổng quan - quy trình phát triển 1 website



- Fix bugs
- Update & upgrade libraries
- Optimize performance
- Upgrade features

Công việc của 1 web developer

- Designing user interfaces and navigation menus
- Writing and reviewing code for sites, typically HTML, XML, or JavaScript
- Integrating multimedia content onto a site
- Testing web applications
- Troubleshooting problems with performance or user experience
- Collaborating with designers, developers, and stakeholders

Tổng quan về lập trình web - setup environment

Front-end

HTML, CSS,
Javascript =>
ReactJS
Tool: Visual code

Backend

Code: NodeJS,
CSDL: MSSQL Express
Tool: Visual code

https://code.visualstudio.com

https://nodejs.org

Microsoft® SQL Server®

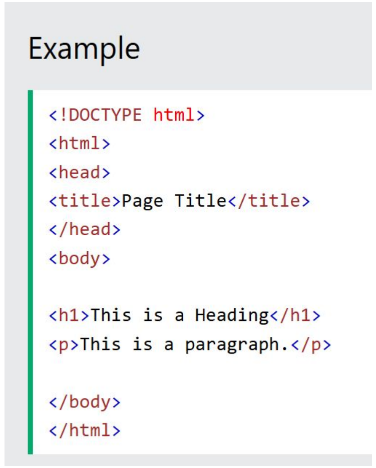
https://www.microsoft.com/en-us/sql-server/sql-server-downloads
(Developer or Express)

Setup environment

- Download and install Visual Code
- Download and install NodeJS
- Download and install MSSQL Express
- Github.com(git clone,git commit,git push, git pull, git fetch) +2

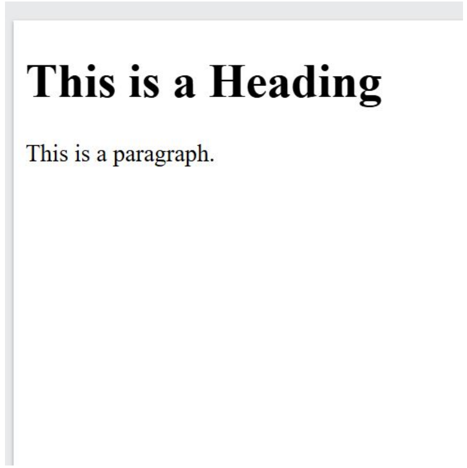
Introduction to HTML

- HTML stands for **HyperText Markup Language**.
- HTML is the standard markup language for Web pages.
- It is used to create the structure of web pages.
- HTML is easy to learn



Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a Heading

This is a paragraph.

History of HTML

- 1991: HTML introduced by Tim Berners-Lee.
- 1993: HTML 1.0 is released. Not many developers are creating websites at this time.
- 1995: HTML 2.0 standardized.
- 1997: HTML 3.0 was invented.
- 1999: HTML 4.01 becomes widely used.
- 2014: HTML5 released with modern features.
- Present: HTML5 is the current standard.

Key Features of HTML

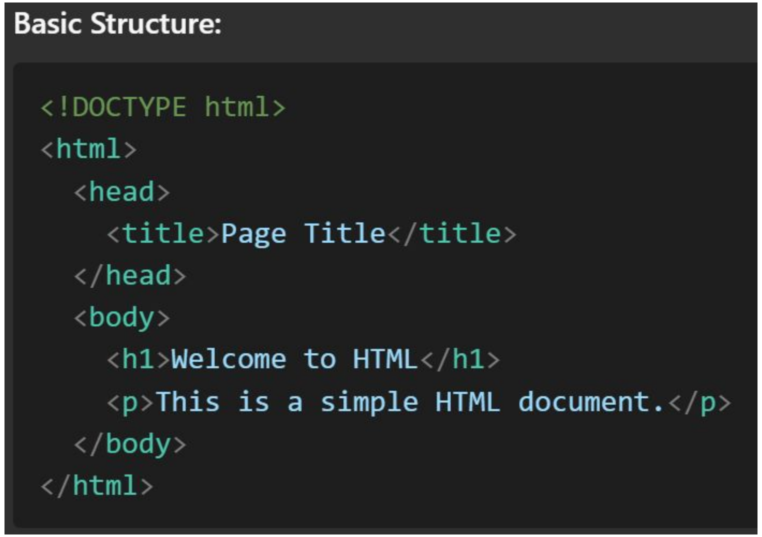
- Simple Syntax: Easy to learn and use.
- Platform Independent: Works across all devices and browsers.
- Flexible: Supports multimedia, links, forms, and more.
- Extensible: Integrates with CSS and JavaScript.

Safari
Apple
MacOS, iOS
Firefox
Mozilla
MacOS, MS Windows, Linux OS Android OS
Chrome
Google
MacOS, MS Windows, Linux OS, Android OS, Chrome OS
Edge
new
Microsoft
MS Windows, MacOS, iOS Android OS
Opera
Opera Software
MacOS, MS Windows, Linux OS, Android OS

HTML Document Structure

Components:

- <!DOCTYPE html>: Declares document type.
- <html>: Root element.
- <head>: Metadata, title, links.
- <body>: Visible content.


Basic Structure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Welcome to HTML</h1>
    <p>This is a simple HTML document.</p>
  </body>
</html>
```

Tools for HTML Development

- **Text Editors:** VS Code, Sublime Text, Notepad++.
- **Browsers:** Chrome, Firefox, Safari.
- **Debugging Tools:** Developer Tools in browsers.
- **Online Resources:** MDN Web Docs, W3Schools.

HTML Elements and Tags

Elements: The HTML element is everything from the start tag to the end tag.

Tags: Enclosed in angle brackets (< >) and often come in pairs.Example: <p>This is a paragraph.</p>

Attributes: Provide additional information about elements.Example: <img src="image.jpg" alt="Description">

<h1>My First Heading</h1>
<p>My first paragraph.</p>

Nested HTML Elements: HTML elements can be nested

Never Skip the End Tag

HTML is Not Case Sensitive

Common HTML Tags

Body tags: display content only in body tag

Headings: <h1> to <h6>

Paragraphs: <p>

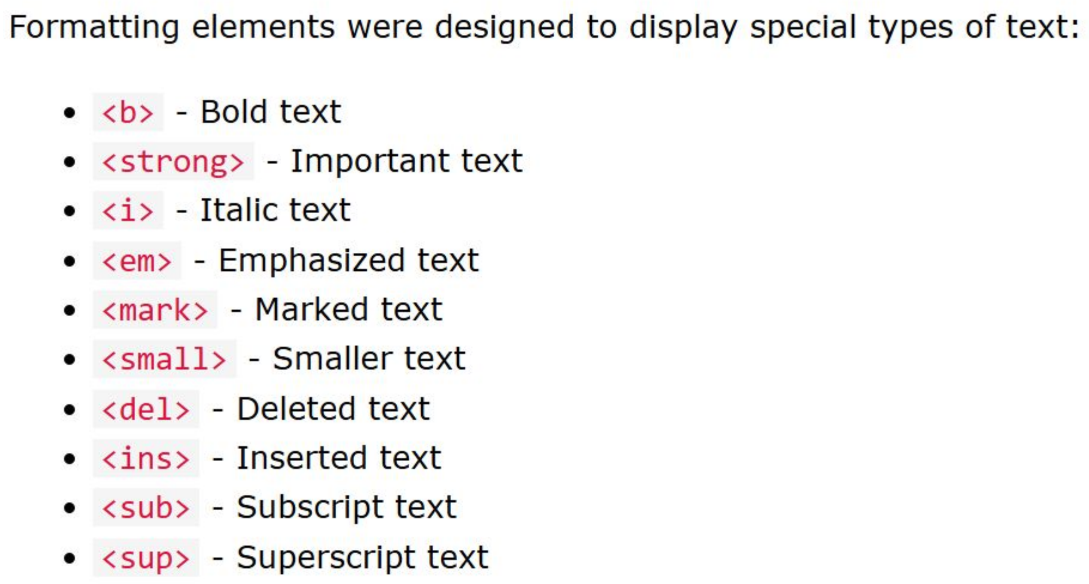
Links: <a href="url">Link</a>

Images: <img src="image.jpg" alt="Description">

Lists: <ul>, <ol>, <li>

Common HTML Tags

HTML Formating



Formatting elements were designed to display special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

Common HTML Tags

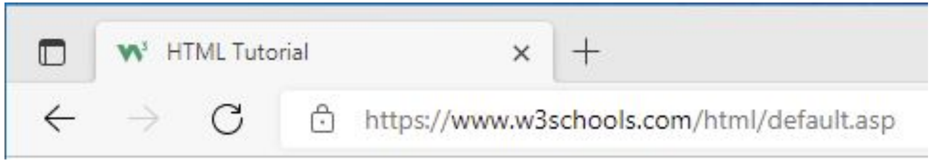
HTML CSS

CSS can be added to HTML documents in 3 ways:

- Inline - by using the `style` attribute inside HTML elements
- Internal - by using a `<style>` element in the `<head>` section
- External - by using a `<link>` element to link to an external CSS file

Common HTML Tags

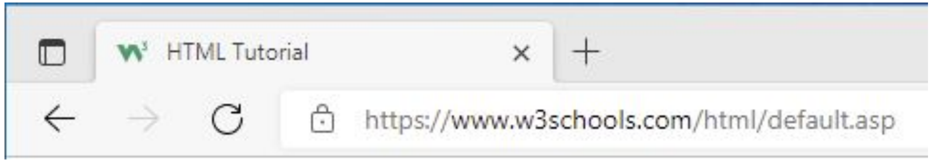
Favicon



```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Common HTML Tags

Page Titlte



```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

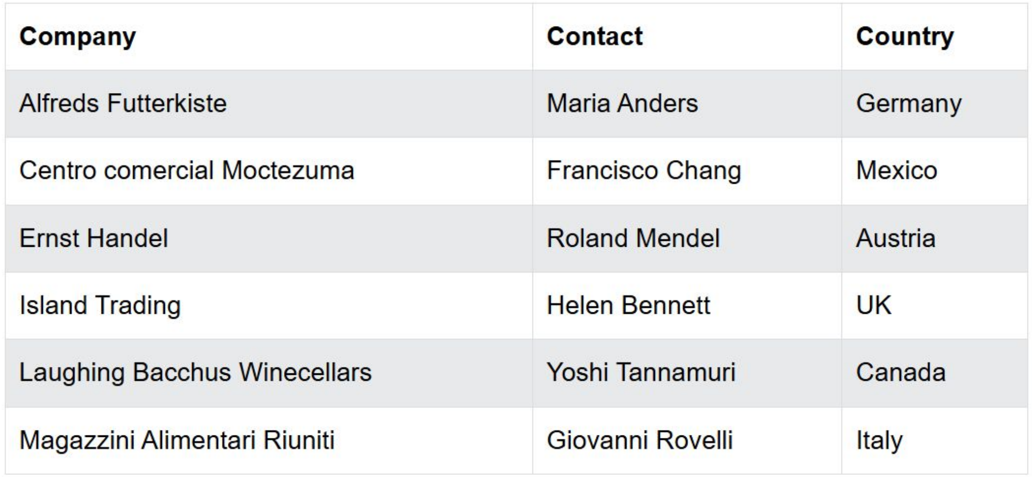
Title should describe the content and the meaning of the page.

The page title is very important for search engine optimization (SEO)

Common HTML Tags

HTML Tables

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```



Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Common HTML Tags

HTML class Attribute

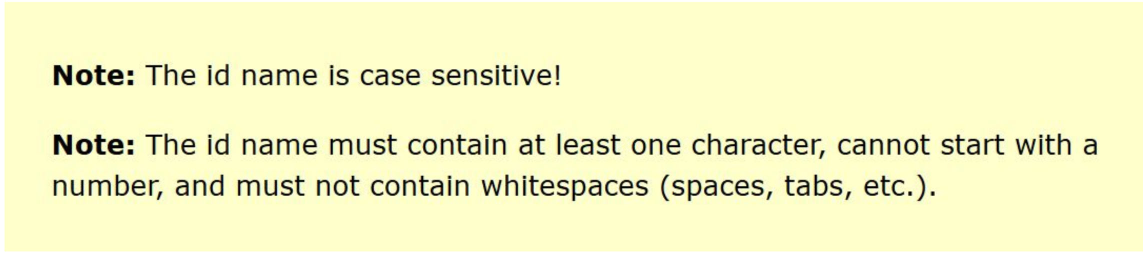
```html
<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>
```

Common HTML Tags

HTML id Attribute

The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document.

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
```



Note: The id name is case sensitive!

Note: The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.).

Difference Between Class and ID

Common HTML Tags

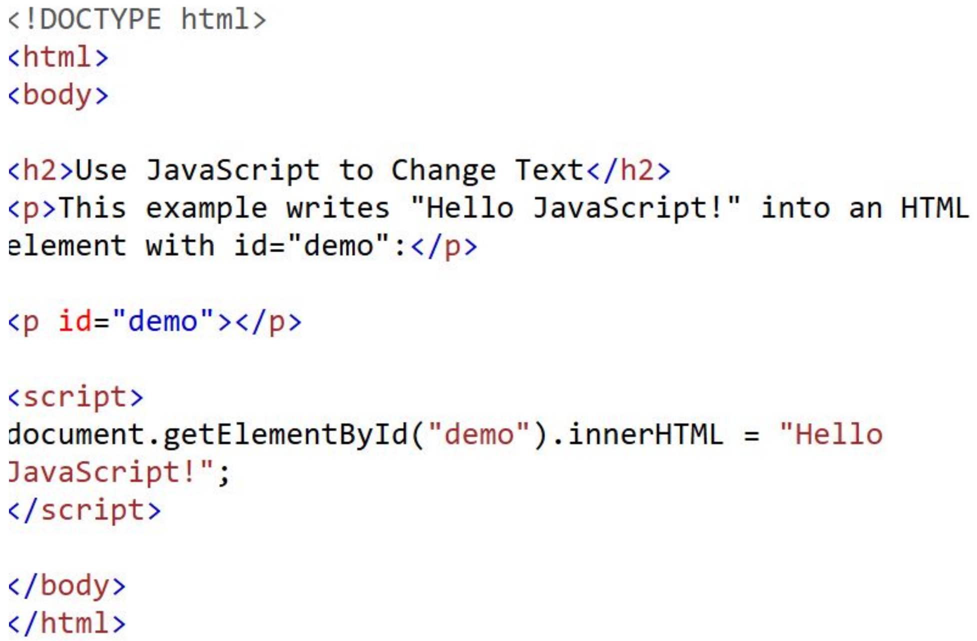
HTML Iframe

```
<iframe src="url" title="description"></iframe>
```

- Height
- Width
- Style
- Remove the border
- `<iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe>`

Common HTML Tags

HTML Javascript



```
<!DOCTYPE html>
<html>
<body>

<h2>Use JavaScript to Change Text</h2>
<p>This example writes "Hello JavaScript!" into an HTML
element with id="demo":</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello
JavaScript!";
</script>

</body>
</html>
```

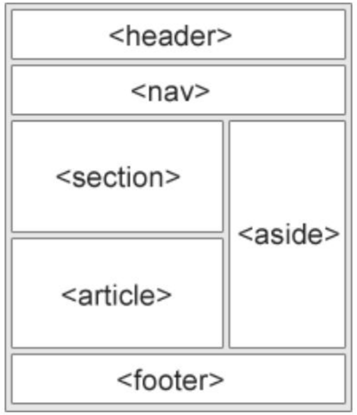
Common HTML Tags

HTML <head> Element

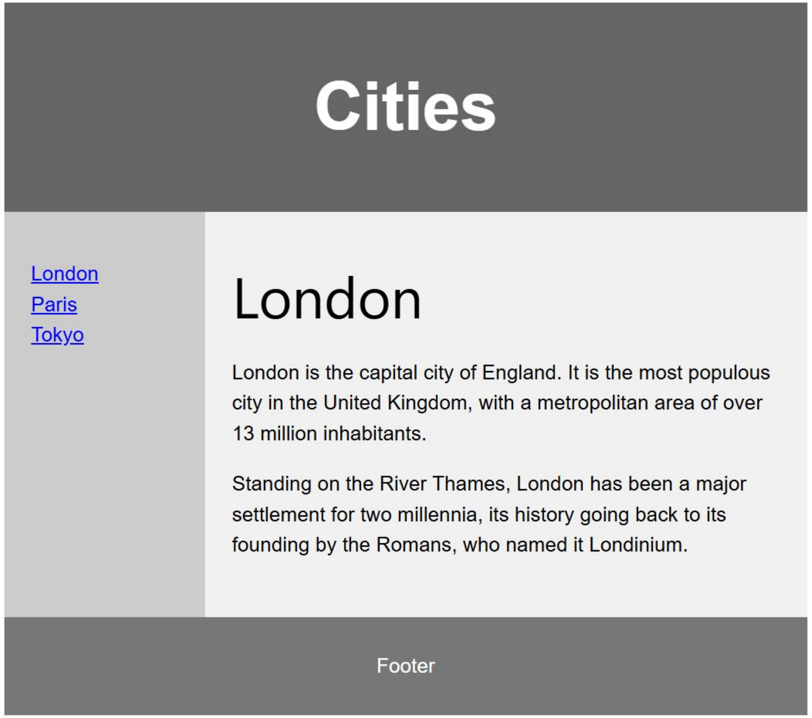
```
<!DOCTYPE html>

<html>

<head>

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

  <meta charset="UTF-8">

  <meta name="description" content="Free Web tutorials">

  <meta name="keywords" content="HTML, CSS, JavaScript">

  <meta name="author" content="John Doe">

</head>

<body>

<p>All meta information goes inside the head section.</p>

</body>
```

Common HTML Tags

HTML Layout



- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a set of navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent, self-contained content
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details that the user can open and close on demand
- `<summary>` - Defines a heading for the `<details>` element



London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

- CSS framework(bootstrap)
- CSS float property
- CSS flexbox

Common HTML Tags

HTML Responsive

```html
<img src="img_girl.jpg" style="max-width:100%;height:auto;">
```

Responsive web design is about creating web pages that look good on all devices!

Responsive Images(max-width)

Media queries

```html
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_smallflower.jpg" alt="Flowers">
</picture>
```

```css
/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or
smaller */
  }
```

Forms in HTML

Used for user input, The user input is most often sent to a server for processing..

Key tags:

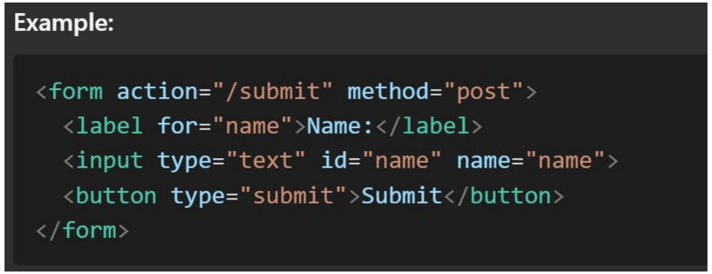
- <form>: Defines a form.
- Method: post, get

- <input>: Input fields (e.g., text, password, radio, checkbox, submit, button).

- <button>: Submit button.

- <textarea>: Multi-line text input.

- Label Element


Example:

```
<form action="/submit" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">
  <button type="submit">Submit</button>
</form>
```

* Input field must have a `name` attribute to be submitted
If the `name` attribute is omitted, the value of the input field will not be sent at all.

Forms in HTML

Form Attributes

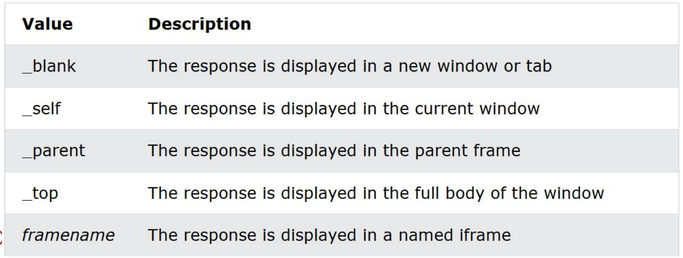
- Action. If the action attribute is omitted, the action is set to the current page.
- The Target Attribute. The target attribute specifies where to display the response that is received after submitting the form.
-
- The Method Attribute (GET, POST)

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visib
- The length of a URL is limited (2048 characters)

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

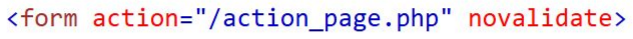


Value	**Description**
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

Forms in HTML

Form Attributes

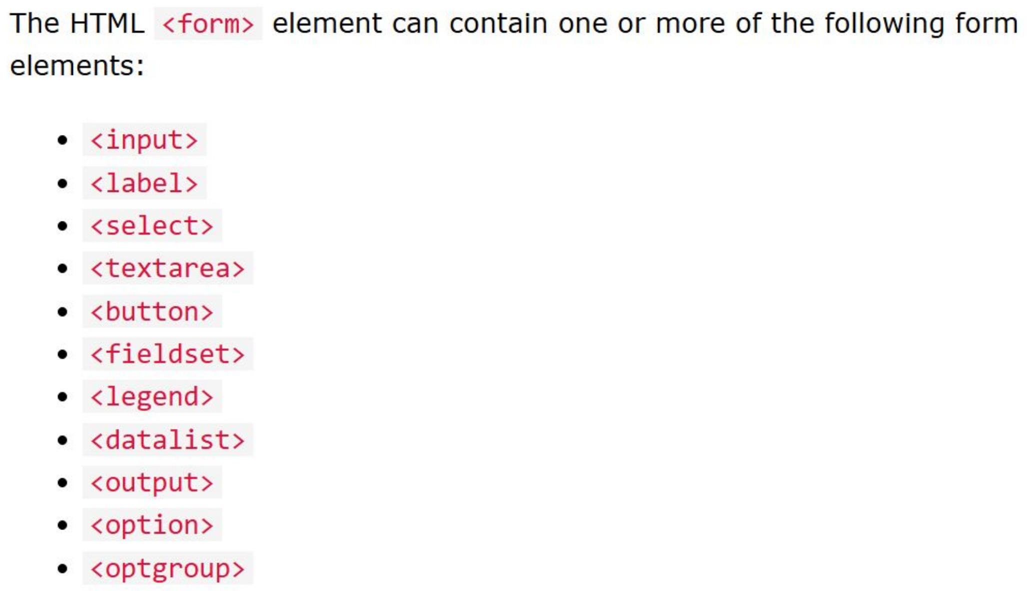
- Autocomplete. If the action attribute is omitted, the action is set to the current page.
- Novalidate:



```
<form action="/action_page.php" novalidate>
```

Forms in HTML

Form Elements



The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

Forms in HTML

Form Elements Input Attribute

- value: The input value attribute specifies an initial value for an input field
- readonly: The input readonly attribute specifies that an input field is read-only.A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it). The value of a read-only input field will be sent when submitting the form!
- disable: The input disabled attribute specifies that an input field should be disabled. A disabled input field is unusable and un-clickable. The value of a disabled input field will not be sent when submitting the form!
- size: The input size attribute specifies the visible width, in characters, of an input field (default:20). Apply for text, search, tel, url, email, and password.
- maxlength: The input maxlength attribute specifies the maximum number of characters allowed in an input field.
- min and max: The input min and max attributes specify the minimum and maximum values for an input field.

Forms in HTML

Form Elements Input Attribute

- multiple: the input multiple attribute specifies that the user is allowed to enter more than one value in an input field.
- pattern: The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted. Apply for text, date, search, url, tel, email, and password.
- placeholder: The input placeholder attribute specifies a short hint
- required: The input required attribute specifies that an input field must be filled out before submitting the form
- autofocus: should automatically get focus when the page loads
- height and width

Multimedia in HTML

Images: <img>

Audio: <audio controls>Example: <audio src="audio.mp3" controls></audio>

Video: <video controls>Example: <video src="video.mp4" controls></video>

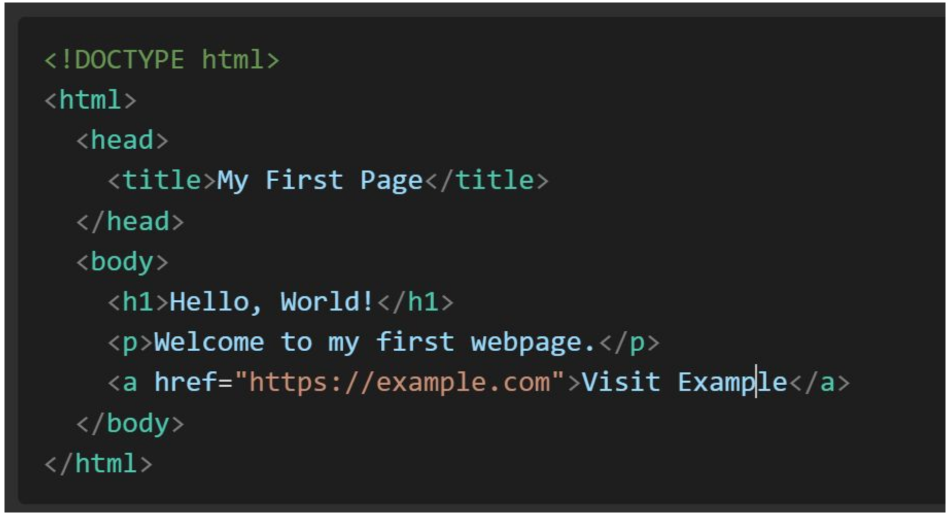
HTML5 Features

- Semantic Elements: <header>, <footer>, <article>, <section>.
- Multimedia Support: <audio>, <video>.
- Forms Enhancements: New input types (e.g., email, date).
- Canvas for Graphics: <canvas> tag for drawing.
- Improved Accessibility: ARIA roles and attributes.

Best Practices in HTML

- Use semantic tags for clarity and accessibility.
- Keep code clean and organized.
- Use alt attributes for images.
- Validate your HTML using online tools (e.g., W3C Validator).

Example Project



```html
<!DOCTYPE html>
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Welcome to my first webpage.</p>
    <a href="https://example.com">Visit Example</a>
  </body>
</html>
```

Home work

Create basic HTML for your project