

BASICS ON INSTALLATION OF BRACKETS, CREATING FILES AND CODING IN THEM

To download and install Brackets, a popular open-source code editor, you can follow these step-by-step instructions:

Step 1: Visit the Brackets Website

Go to the official Brackets website at <https://brackets.io/>.

Step 2: Download Brackets

On the Brackets website's homepage, you'll see a "Download Brackets" button. Click on it.

Step 3: Choose Your Operating System

Brackets is available for Windows, macOS, and Linux. Select the appropriate version for your operating system by clicking on the corresponding icon or text. For example, if you're using Windows, click on the Windows icon.

Step 4: Start the Download

After clicking the icon for your operating system, the download should start automatically. You'll see a progress bar indicating the download's status.

Step 5: Install Brackets

Once the download is complete, locate the downloaded installer file. It's usually in your computer's "Downloads" folder unless you specified a different location.

- For Windows: Double-click the downloaded .exe file to start the installation process. Follow the on-screen instructions. You can choose installation options during this process.
- For macOS: Open the downloaded .dmg file. Drag the Brackets icon to the Applications folder to install it.
- For Linux: The installation process on Linux varies depending on your distribution. You may need to use terminal commands or package managers to install Brackets. Refer to Brackets' documentation or your distribution's package manager for specific instructions.

Step 6: Launch Brackets

Once the installation is complete, you can launch Brackets. On Windows and macOS, you can typically find Brackets in your applications or start menu. On Linux, you might need to run it from the terminal or find it in your applications menu.

Step 7: Start Coding

Brackets should now be installed and ready to use. You can start coding by opening or creating HTML, CSS, and JavaScript files within the editor.

That's it! You've successfully downloaded and installed Brackets, and you're ready to begin coding in this lightweight and user-friendly code editor.

HTML BASICS

Introduction to HTML:

HTML (Hypertext Markup Language) is the standard markup language used to create and structure content on the web. It provides a set of elements or tags that define the structure and presentation of web documents, including text, links, images, forms, and multimedia. HTML is the foundation of web development and is essential for creating web pages and web applications.

HTML documents are text files with an ".html" or ".htm" extension and are composed of elements enclosed in angle brackets ("`<`" and "`>`"). These elements are used to define the structure of a web page and its various components. HTML documents are rendered by web browsers to display the content to users.

Basic Structure of an HTML Document:

An HTML document consists of two main sections: the `<head>` and the `<body>`. Here's an overview of each section and how to code using them:

The `<head>` Section:

The `<head>` section of an HTML document contains meta-information about the web page, such as its title, character encoding, linked stylesheets, and metadata for search engines. **It doesn't display any content directly to the user but plays a crucial role in providing information about the page.**

To create the `<head>` section, you use the `<head>` element, and within it, you

can include various elements and meta-tags. Here's an example of a simple

`<head>` section:

Html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
  <title>My Web Page</title>
```

```
  <link rel="stylesheet" type="text/css" href="styles.css">
```

```
</head>
```

```
<body>
```

```
  <!-- Content goes here -->
```

```
</body>
```

```
</html>
```

In this example:

- `<meta charset="UTF-8">` specifies the character encoding for the document (UTF-8 is commonly used for handling various character sets).
- `<title>` sets the title of the web page, which appears in the browser's title bar or tab.
- `<link>` links an external stylesheet (styles.css) for styling the page.

The `<body>` Section:

The `<body>` section contains the main content of the web page, such as text, images, links, and other elements that are visible to the user. This is where you structure the actual content that users interact with.

To create the `<body>` section, you use the `<body>` element, and within it, you place your page's content. Here's an example with some basic content:

html

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>My Web Page</title>

  <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

  <h1>Welcome to My Web Page</h1>

  <p>This is a paragraph of text.</p>

  

  <a href="https://www.example.com">Visit Example.com</a>

</body>

</html>
```

In this example:

- `<h1>` is a header element for the main heading.
- `<p>` represents a paragraph of text.
- `` displays an image.
- `<a>` creates a hyperlink to another web page.

By combining elements within the `<head>` and `<body>` sections, you can create rich and structured web pages. HTML provides a wide range of elements to format text,

create lists, embed media, and add interactive features, making it a powerful tool for web development.