

Quote Generator Student Guide

[Beginner-Friendly]

This guide walks you through every part of building your very own quote generator using **HTML**, **CSS**, and **JavaScript**. If you’ve never coded before — don’t worry. We explain everything step by step.

STEP 1: Basic Web Page (HTML)

Create a new file called index.html and copy in the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quote Generator</title>
  <style>
    /* CSS goes here - we'll explain in the next step */
  </style>
</head>
<body>
  <div class="container">
    <h1>Random Quote Generator</h1>
    <div id = "quote"> Click Button to See Quote </div>
    <div id= "author"></div>
    <button id = "new-quote"> New Quote</button>
  </div>

  <script>
    // JavaScript goes here - we'll explain in a later step
  </script>
</body>
</html>
```

What each part means:

- <!DOCTYPE html> tells the browser we’re using modern HTML5.
- <html lang="en"> starts our webpage and says the language is English.
- <head> contains settings like the page title and style info.
- <meta charset="UTF-8"> lets us show special characters correctly.
- <meta name="viewport" ...> helps the layout adjust for phone/tablet screens.
- <title> is the text shown in the browser tab.
- <style> is where we’ll put our design [CSS] later.
- <body> holds everything the user sees.
- div class="container" is a wrapper for our content to style it better.
- <h1> is the big heading at the top.
- <p id="quote"> is where the quote will appear.
- <button> creates a clickable button that will run our code.
- <script> is where our JavaScript [behavior logic] will live.

CSS Styling [Step-by-Step Explanation]

CSS [Cascading Style Sheets] is used to **style** your HTML — it makes your app look clean, centered, and modern. Let's go through it piece by piece.

body — Full Page Styling

```
body {
  font-family: Arial, sans-serif;
  background: #f0f0f0;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  margin: 0;
}
```

Explanation:

- font-family: Arial, sans-serif; → Sets the default text style to be clean and easy to read.
- background: #f0f0f0; → Gives a light grey background to the whole page.
- display: flex; → Turns the body into a **flex container**, allowing us to center items.
- align-items: center; → Vertically centers the content.
- justify-content: center; → Horizontally centers the content.
- height: 100vh; → Makes the page body take up the **full screen height**.
- margin: 0; → Removes default spacing around the edges of the page.

.container — The Main Quote Box

```
.container {
  background: white;
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  max-width: 500px;
  text-align: center;
}
```

Explanation:

- background: white; → Makes the box have a white background.
- padding: 2rem; → Adds space **inside** the box (around the content).
- border-radius: 10px; → Rounds the box corners to make it look smoother.
- box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); → Adds a light shadow underneath for depth.
- max-width: 500px; → Keeps it from getting too wide on big screens.
- text-align: center; → Centers the quote and button inside the box.

.quote — The Quote Text

```
#quote {
  font-size: 1.2rem;
  margin: 1rem 0;
}
```

Explanation:

- font-size: 1.2rem; → Makes the quote a bit larger for better readability. (👁️👁️ “rem” is relative to the default font size, which is usually 16px. So 1.2rem ≈ 19.2px)
- margin: 1rem 0; → Adds vertical spacing (top and bottom only).

.author — The Author Text

```
.author {  
  font-size: 1rem;  
  color: #555;  
  margin 1.5rem;  
}
```

Explanation:

- font-size: 1rem; → *rem*” is relative to the default font size, which is usually 16px. [(👁👁) “rem” is relative to the default font size, which is usually 16px. So 1.2rem ≈ 19.2px]
- margin: 1.5rem 0; → Adds vertical spacing (top and bottom only).

button — The Generator Button

```
button {  
  background-color: #4caf50;  
  color: white;  
  border: none;  
  padding: 0.7rem 1.5rem;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

Explanation:

- background-color: #4caf50; → Sets a nice green color.
- color: white; → Makes the text white for contrast.
- border: none; → Removes the default browser border.
- padding: 0.7rem 1.5rem; → Adds space inside the button (top/bottom and left/right).
- border-radius: 5px; → Rounds the corners a bit for a modern look.
- cursor: pointer; → Changes the cursor to a pointer when you hover over it.

button:hover — Button Hover Effect

```
button:hover {  background-color: #45a049; } w
```

Explanation:

- This changes the button’s background color **when the user hovers** over it.
- It’s a subtle effect that makes the button feel more interactive.

What this CSS does:

- Makes the page centered and clean-looking.
- Styles the container box to look nice.
- Changes the font and background.
- Styles the button with colors, padding, and hover effect.

```
body {
  font-family: Arial, sans-serif;
  background: #f0f0f0;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  margin: 0;
}

.container {
  background: white;
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  max-width: 500px;
  text-align: center;
}

#quote {
  font-size: 1.2rem;
  margin: 1rem 0;
}

button {
  background-color: #4caf50;
  color: white;
  border: none;
  padding: 0.7rem 1.5rem;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}
```

JavaScript Logic (Step-by-Step Breakdown)

JavaScript controls the **logic** and **interactivity** of your app. Here, it helps randomly choose a quote when a user clicks the button.

Selecting Elements from the Page (DOM)

```
const quoteText = document.getElementById('quote');
const authorText = document.getElementById('author');
const newQuote = document.getElementById('new-quote');
```

Explanation:

- We're grabbing 3 HTML elements so we can change their content later using JavaScript.
- `document.getElementById()` is how we select elements by their id.
 - `quoteText` is where the quote will appear.

- authorText shows the author name.
- newQuote is the button the user clicks.

Creating the Quotes List

```
const quotes = [  
  { text: "Believe it you are the strongest", author: "Theodore Roosevelt" },  
  { text: "Help yourself out", author: "Davinci" },  
  { text: "Don't stress yourself out", author: "Newton" } ];
```

Explanation:

- This is an **array** of **objects**.
- Each object contains:
 - A text value [the quote],
 - An author value [the name].
- This lets us randomly pick and show one of these when the button is clicked.

Writing the Quote Generator Function

```
function GetQuote() {  
  const randomIndex = Math.floor(Math.random() * quotes.length);  
  const quote = quotes[randomIndex];  
  quoteText.textContent = quote.text;  
  authorText.textContent = quote.author;  
  
}
```

Explanation:

- Math.random() gives a random number **between 0 and 1**.
- Math.random() * quotes.length scales that to the size of the array (in this case, 3).
- Math.floor() rounds it down to the nearest whole number (e.g., 1.8 → 1), so we can use it as an index.
- We then select a **random quote** from the array and update the page with:
 - quoteText.textContent → the actual quote.
 - authorText.textContent → the author.

Making the Button Work

```
newQuote.addEventListener('click', GetQuote);
```

Explanation:

- This line **listens** for a click on the "New Quote" button.
- When clicked, it **runs the** GetQuote() **function**, changing the quote and author.

Final Logic (JavaScript)

Inside the <script> tag at the bottom of your HTML, add this code:

```
const quoteText = document.getElementById('quote');
const authorText = document.getElementById('author');
const newQuote = document.getElementById('new-quote');

const quotes = [
  { text: "Believe it you are the strongest", author: "Theodore Roosevelt" },
  { text: "Help yourself out ", author: "Davinci"},
  { text: "Don't stress yourself out ", author: "Newton"},
];

function GetQuote() {
  const randomIndex = Math.floor(Math.random () * quotes.length);
  const quote = quotes[randomIndex];
  quoteText.textContent = quote.text;
  authorText.textContent = quote.author;
}

newQuote.addEventListener('click', GetQuote);
```

Breakdown:

- quotes[]: This is a list (array) of phrases we want to show.
- Math.random() gives a random decimal between 0 and 1.
- Math.floor() rounds it down to a whole number (like 2.9 → 2).
- quotes.length tells us how many items are in the list.
- document.getElementById("quote"): This grabs the element in the HTML with the id="quote".
- .innerText = selectedQuote; changes what's shown in that box.

Test It!

1. Save your file.
2. Open it in a browser (double-click it or use Live Server in VS Code).
3. Click the button — it should show a random quote!

You've just built your first working JavaScript app! 🎉

Let me know if you'd like to continue with the **AI version** or a guide for adding **bonus features (enhancements)**!