

Meme Generator

For this assignment, you'll be building a meme generator in the browser using HTML, CSS, and JavaScript.

Your generator should consist of a form that accepts a link to an image, text for the top of the meme, and text for the bottom of the meme. When the user submits the form, use JavaScript to append to the DOM a div which contains the meme, including the image and its text.

Requirements

- The HTML page shows a form with three input options - Top Text, Bottom Text, Image - and a submit button
- Users should be able to submit a form on the page to generate a new meme on the page
- They should be able to add **multiple memes** to the page by submitting the form multiple times
- Users should be able to click on a button to delete/remove a meme from the page
- The form boxes should clear out automatically after the submit is clicked
- The form fields need to have validation so they will not submit if a field is missing
- Be sure to style your meme generator! It should be functional but also look nice
- **Only use vanilla JavaScript: no frameworks/third-party libraries are allowed**

To create a meme generator with the specified requirements, we'll need to implement HTML, CSS, and JavaScript. Here's a step-by-step guide to build the meme generator:

HTML Structure

First, let's create the HTML structure for the meme generator form and the area where memes will be displayed:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Meme Generator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Meme Generator</h1>
```

```

    <form id="memeForm">
      <input type="text" id="topText" placeholder="Top Text" required>
      <input type="text" id="bottomText" placeholder="Bottom Text" required>
      <input type="url" id="imageUrl" placeholder="Image URL" required>
      <button type="submit">Generate Meme</button>
    </form>
    <div id="memeContainer"></div>
  </div>
  <script src="script.js"></script>
</body>
</html>

```

CSS Styling

Next, let's add some basic styling to make the meme generator look nice:

```

body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  margin: 0;
  padding: 20px;
}

.container {
  max-width: 600px;
  margin: 0 auto;
  background-color: white;
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

h1 {
  text-align: center;
  color: #333;
}

form {
  display: flex;
  flex-direction: column;
  gap: 10px;
  margin-bottom: 20px;
}

input, button {
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ddd;
  border-radius: 4px;
}

button {
  background-color: #4CAF50;
  color: white;
  border: none;
}

```

```
        cursor: pointer;
        transition: background-color 0.3s;
    }

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

    .meme {
        position: relative;
        margin-bottom: 20px;
    }

    .meme img {
        width: 100%;
        display: block;
    }

    .meme-text {
        position: absolute;
        left: 0;
        right: 0;
        padding: 10px;
        font-size: 24px;
        font-weight: bold;
        text-align: center;
        text-transform: uppercase;
        color: white;
        text-shadow: 2px 2px 0 #000;
    }

    .top-text {
        top: 10px;
    }

    .bottom-text {
        bottom: 10px;
    }

    .delete-btn {
        position: absolute;
        top: 5px;
        right: 5px;
        background-color: #ff4444;
        color: white;
        border: none;
        padding: 5px 10px;
        cursor: pointer;
        font-size: 14px;
        border-radius: 3px;
    }
```

JavaScript Functionality

Finally, let's implement the JavaScript functionality to generate memes, add them to the page, and allow for deletion:

```
document.addEventListener('DOMContentLoaded', () => {
  const memeForm = document.getElementById('memeForm');
  const memeContainer = document.getElementById('memeContainer');

  memeForm.addEventListener('submit', (e) => {
    e.preventDefault();

    const topText = document.getElementById('topText').value;
    const bottomText = document.getElementById('bottomText').value;
    const imageUrl = document.getElementById('imageUrl').value;

    if (topText && bottomText && imageUrl) {
      createMeme(topText, bottomText, imageUrl);
      memeForm.reset();
    }
  });

  function createMeme(topText, bottomText, imageUrl) {
    const memeDiv = document.createElement('div');
    memeDiv.classList.add('meme');

    const img = document.createElement('img');
    img.src = imageUrl;
    img.alt = 'Meme Image';
    img.onerror = () => {
      alert('Error loading image. Please check the URL and try again.');
```

```
      memeDiv.remove();
    };

    const topTextDiv = document.createElement('div');
    topTextDiv.classList.add('meme-text', 'top-text');
    topTextDiv.textContent = topText;

    const bottomTextDiv = document.createElement('div');
    bottomTextDiv.classList.add('meme-text', 'bottom-text');
    bottomTextDiv.textContent = bottomText;

    const deleteBtn = document.createElement('button');
    deleteBtn.classList.add('delete-btn');
    deleteBtn.textContent = 'Delete';
    deleteBtn.addEventListener('click', () => memeDiv.remove());

    memeDiv.appendChild(img);
    memeDiv.appendChild(topTextDiv);
    memeDiv.appendChild(bottomTextDiv);
    memeDiv.appendChild(deleteBtn);

    memeContainer.prepend(memeDiv);
  }
});
```

This implementation fulfills all the requirements:

1. The HTML form includes input fields for top text, bottom text, and image URL, along with a submit button^[1] ^[2].
2. Users can submit the form to generate a new meme on the page^[3].
3. Multiple memes can be added by submitting the form multiple times^[3].
4. Each meme has a delete button to remove it from the page^[4].
5. The form fields clear automatically after submission^[5].
6. Form validation is implemented using HTML5 `required` attributes and JavaScript checks^[6] ^[7].
7. The meme generator is styled to look visually appealing^[8].
8. Only vanilla JavaScript is used, without any external libraries or frameworks^[4] ^[9].

This meme generator allows users to create custom memes by entering text and an image URL. The memes are dynamically added to the page, and users can delete individual memes as needed. The design is responsive and user-friendly, providing a smooth experience for creating and managing memes.

✱✱

1. <https://www.youtube.com/watch?v=io5FcMAdLyQ>
2. https://www.w3schools.com/js/js_validation.asp
3. <https://stackoverflow.com/questions/73198508/stuck-trying-to-append-a-meme-for-my-meme-generator>
4. https://github.com/Mhowitt/meme_generator
5. https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Forms/Form_validation
6. <https://github.com/seanoliver/meme-gen>
7. <https://www.freecodecamp.org/news/form-validation-with-html5-and-javascript/>
8. <https://followedapp.com/meme-generator/>
9. <https://codingtorque.com/meme-generator-using-javascript/>