**meme\_generator**

Meme generator Project

Part 4 - JavaScript and the DOM, Part I

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

As a user, I can submit a form on the page to generate a new meme on the page.

As a user, I can add multiple memes to the page by submitting the form multiple times.

As a user, I can click on a button to remove a meme that I’ve added to the page.

(Technical) When the meme form is submitted, values in the form inputs should be cleared out.

Be sure to style your meme generator! It should be functional but also look nice.

Make sure you use vanilla JavaScript only - no frameworks or third-party libraries.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" integrity="sha384-rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ" crossorigin="anonymous">

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

<title>Meme Generator</title>

<script defer src="https://use.fontawesome.com/releases/v5.0.4/js/all.js"></script>

</head>

<body>

<div class="background-blue">

<header class="container-fluid">

<div class="container mx-auto text-center mb-5 title">

<h1 class="text-info">A Meme Generator for the Generations</h1>

<h2>Create your own unique memes with this easy to use generator</h2>

</div>

<div class="steps">

<p></p>

</div>

</header>

<section class="container">

<form id="meme-form">

<!-- Image section of form -->

<div class="group-form row mb-5">

<label for="link-to-pic" class="col-md-2 col-form-label">Image URL Here:</label>

<div class="col-md-10">

<input type="url" class="form-control" id="uploaded-img" placeholder="Your URL Goes Here" required>

</div>

</div>

<!-- text-top section of form -->

<div class="group-form row mb-5">

<label for="text-on-top" class="col-md-2 col-form-label">Text on top:</label>

<div class="col-md-10">

<input type="text" class="form-control" id="text-top" placeholder="Your Top Text Goes Here (Optional)">

</div>

</div>

<!-- text-bottom section of form -->

<div class="group-form row mb-5">

<label for="text-on-bottom" class="col-md-2 col-form-label">Text on Bottom:</label>

<div class="col-md-10">

<input type="text" class="form-control" id="text-bottom" placeholder="Your Bottom Text Goes Here (Optional)">

</div>

</div>

<div class="group-form row mb-5">

<label class="col-md-2 col-form-label">&nbsp;</label>

<div class="col-md-2">

<input type="submit" class="btn btn-success" value="Generate it!">

</div>

</div>

</form>

<div class="line"></div>

</section>

</div>

<section class="meme-gallery">

<ul class="gallery">

</ul>

</section>

<footer>

<div class="container-fluid footer">

<p>Keep on meme-ing on. Never give up on the meme times</p>

</div>

</footer>

<!-- JS SCRIPTS -->

<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js" integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed950n" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js" integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js" integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkE+jo0ieGizqPLForn" crossorigin="anonymous"></script>

<script src="script.js"></script>

<!-- <script scr="script.js" type="text/javascript"></script> -->

</body>

</html>

/\* --------------------------------------------------- \*/

/\* Basic Setup \*/

/\* --------------------------------------------------- \*/

\* {

margin: 0;

padding: 0;

}

html {

color: #555;

font-family: 'Raleway', 'Arial', sans-serif;

font-weight: 400;

font-size: 16px;

text-rendering: optimizeLegibility;

}

/\* --------------------------------------------------- \*/

/\* Resuable content \*/

/\* --------------------------------------------------- \*/

h1 {

font-family: Impact, Arial, sans-serif;

-webkit-text-stroke-width: 1.5px;

-webkit-text-stroke-color: black;

font-size: 320%;

text-transform: uppercase;

letter-spacing: 1px;

}

h2 {font-family: Tahoma, Arial, sans-serif;}

.background-blue,

.meme-gallery {

background-color: #d4f0fd;

padding: 1%;

}

/\* --------------------------------------------------- \*/

/\* Header \*/

/\* --------------------------------------------------- \*/

.title {width: 900px;}

/\* --------------------------------------------------- \*/

/\* Meme Gallery \*/

/\* --------------------------------------------------- \*/

.gallery {

list-style: none;

width: 100%;

}

.meme-img{

display: inline-block;

margin: 1%;

width: 500px;

height: auto;

text-align: center;

position: relative;

opacity: 1;

}

img {width: 500px;}

.meme-img:hover {opacity: 0.6;}

.red-cross {

width: 500px;

height: 250px;

content: "X";

position: absolute;

top: 50%;

left:50%;

transform: translate(-50%, -66%);

font-weight: 700;

text-align: center;

font-size: 1500%;

opacity: 0;

}

.red-cross:hover {

opacity: 1.6;

z-index: 20;

}

.text {

font-family: Impact, Arial, sans-serif;

width: 500px;

font-size: 250%;

line-height: 1.1em;

letter-spacing: 1.5px;

color: black;

-webkit-text-fill-color: white;

-webkit-text-stroke-width: 1.7px;

-webkit-text-stroke-color: black;

text-transform: uppercase;

overflow-wrap: break-word;

word-wrap: break-word;

hyphens: auto;

opacity: 1;

}

.text:hover {opacity: 0.6;}

.top {

position: absolute;

top: 0;

z-index: 9;

max-height: 50%;

}

.bottom {

position: absolute;

bottom: 0;

z-index: 9;

max-height: 50%;

}

/\* --------------------------------------------------- \*/

/\* Footer \*/

/\* --------------------------------------------------- \*/

footer {

background-color: #b9ebee;

padding: 20px;

font-size: 80%;

text-align: center;

border-top: 1px solid #c7eff1;

}

.footer p {font-family: Impact, Arial, sans-serif;

text-transform: uppercase;}

/\* --------------------------------------------------- \*/

/\* Media Queries \*/

/\* --------------------------------------------------- \*/

/\*large ipad to desktop/laptop \*/

@media only screen and (max-width: 1200px) {

}

/\*small tablet Large Ipad \*/

@media only screen and (max-width: 1023px) {

}

/\*Large Phones to small tablets\*/

@media only screen and (max-width: 767px) {

.title {width: 85%;}

h1{font-size: 280%;}

h2 {font-size: 190%;}

}

/\*Small Phones \*/

@media only screen and (max-width: 480px) {

h1{font-size: 250%;}

h2 {font-size: 110%;}

html{font-size: 90%;}

img,

.meme-img,

.red-cross,

.text {width: 350px;}

.text {font-size: 160%;}

document.addEventListener("DOMContentLoaded", function(){

var memeForm = document.getElementById("meme-form");

var listGallery = document.querySelector(".gallery");

memeForm.addEventListener("submit", function(event){

event.preventDefault();

// creating the li element

var memeLi = document.createElement('li');

memeLi.classList.add("meme-img");

// creating the canvas element to put our img background

var topText = document.getElementById("text-top");

var urlInput = document.getElementById('uploaded-img').value,

src = urlInput,

img = document.createElement('img');

img.src = src;

// img.width = "500";

var topTextDiv = document.createElement('div');

topTextDiv.classList.add("text", "top");

topTextDiv.innerText = document.getElementById("text-top").value;

var bottomTextDiv = document.createElement('div');

bottomTextDiv.classList.add("text", "bottom");

bottomTextDiv.innerText = document.getElementById("text-bottom").value;

var removeDiv = document.createElement('div');

removeDiv.classList.add("red-cross");

removeDiv.innerText = "X";

removeDiv.style.color = "red";

listGallery.appendChild(memeLi);

memeLi.appendChild(img);

memeLi.appendChild(topTextDiv);

memeLi.appendChild(bottomTextDiv);

memeLi.appendChild(removeDiv);

//memeLi.appendChild(removeButton);

memeForm.reset();

});

function remove(event){

event.target.parentNode.remove();

}

listGallery.addEventListener('click', remove, false);

});

HTML Code

Starter Template

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<!-- Font Awesome Icons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css"

integrity="sha512-+4zCK9k+qNFUR5X+cKL9EIR+ZOhtIloNl9GIKS57V1MyNsYpYcUrUeQc9vNfzsWfV28IaLL3i96P9sdNyeRssA=="

crossorigin="anonymous" />

<!-- CSS -->

<link rel="stylesheet" href="style.css">

<title>Meme Generator using JavaScript - Coding Torque</title>

</head>

<body>

<!-- Further code here -->

<script src="script.js"></script>

</body>

</html>

Paste the below code in your <body> tag.

<div class="header">

<h4>Meme Generator</h4>

<div>

<button id="export">CREATE</button>

</div>

</div>

<div class="container">

<div>

<div id="canvasWrapper">

</div>

</div>

<div class="memeForm">

<h5>Source Image</h5>

<div class="box">

<div>

<p>From URL</p>

<div class="text\_input">

<input id="imgURL" class="input" type="text" placeholder="Link to image" />

</div>

</div>

<div>

<p>Upload from your device</p>

<input id="imgFile" type="file" accept="image/\*" />

</div>

</div>

<h4 style="margin-top: 50px;">Meme Text</h4>

<div class="box">

<div>

<p>Top Text</p>

<div class="text\_input">

<input id="textTop" type="text" class="input" placeholder="Top text" />

</div>

</div>

<div>

<p>Bottom Text</p>

<div class="text\_input">

<input id="textBottom" type="text" class="input" placeholder="Bottom text" />

</div>

</div>

</div>

<h4>Text Size</h4>

<div class="box">

<div>

<p>Top Text: <span id="textSizeTopOut">10</span></p>

<div class="sliderContainer">

<input id="textSizeTop" type="range" min="2" max="50" step="2" />

</div>

</div>

<div>

<p>Bottom Text: <span id="textSizeBottomOut">10</span></p>

<div class="sliderContainer">

<input id="textSizeBottom" type="range" min="2" max="50" step="2" />

</div>

</div>

</div>

<div class="box">

<div>

<h4>Preview Size</h4>

<input id="trueSize" type="checkbox" />

<label for="trueSize"><span>Show true size</span></label>

</div>

</div>

</div>

</div>

CSS Code

Create a file style.css and paste the code below.

body {

background: #1e293b;

color: white;

}

.fullwidth {

width: 100%;

min-width: 400px;

max-height: 800px;

}

.header {

display: flex;

align-items: center;

justify-content: space-between;

padding: 1rem 5rem;

background-color: #334155;

}

#export {

background: #3b82f6;

color: white;

border: none;

border-radius: 4px;

padding: 5px 20px;

}

.container {

display: flex;

padding: 0.5rem 5rem;

}

.memeForm {

display: flex;

flex-direction: column;

justify-content: flex-start;

padding-left: 2rem;

}

.box {

display: flex;

}

.box div {

margin-right: 30px;

}

.text\_input {

margin-bottom: auto;

margin-top: auto;

height: 50px;

background-color: #353b48;

border-radius: 30px;

padding: 10px;

}

.input {

color: white;

border: 0;

outline: 0;

background: none;

width: 250px;

caret-color: transparent;

line-height: 30px;

transition: width 0.4s linear;

padding: 0 10px;

}

.text\_input:hover > .input {

padding-right: 0 15px;

caret-color: deepskyblue;

transition: width 0.4s linear;

}

/\* custom sliders for text sizes \*/

div.sliderContainer {

width: 200px;

text-align: center;

}

#textSizeTop,

#textSizeBottom {

-webkit-appearance: none;

appearance: none;

height: 18px !important;

width: 100%;

border-radius: 10em;

background-color: deepskyblue;

outline: none;

margin-bottom: 14px;

}

#textSizeBottom::-webkit-slider-thumb,

#textSizeTop::-webkit-slider-thumb {

-webkit-appearance: none;

appearance: none;

width: 25px;

height: 25px;

border-radius: 50%;

background: #3b82f6;

cursor: pointer;

border: 3px solid #f4f4f4;

}

#textSizeBottom::-moz-range-thumb,

#textSizeTop::-moz-range-thumb {

width: 25px;

height: 25px;

border-radius: 50%;

background-color: #dbc500;

cursor: pointer;

border: 3px solid #f4f4f4;

}

input[type="checkbox"] {

display: none;

}

input[type="checkbox"] + label {

display: block;

position: relative;

padding-left: 35px;

margin-bottom: 20px;

font: 14px/20px "Open Sans", Arial, sans-serif;

color: #ddd;

cursor: pointer;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

}

input[type="checkbox"] + label:last-child {

margin-bottom: 0;

}

input[type="checkbox"] + label:before {

content: "";

display: block;

width: 20px;

height: 20px;

border: 2px solid #6cc0e5;

position: absolute;

left: 0;

top: 0;

opacity: 0.6;

-webkit-transition: all 0.12s, border-color 0.08s;

transition: all 0.12s, border-color 0.08s;

}

input[type="checkbox"]:checked + label:before {

width: 10px;

top: -5px;

left: 5px;

border-radius: 0;

opacity: 1;

border-top-color: transparent;

border-left-color: transparent;

-webkit-transform: rotate(45deg);

transform: rotate(45deg);

}

JavaScript Code

Create a file script.js and paste the code below.

// CAN\NVAS.js plugin

// ninivert, december 2016

(function (window, document) {

/\*\*

\* CAN\VAS Plugin - Adding line breaks to canvas

\* @arg {string} [str=Hello World] - text to be drawn

\* @arg {number} [x=0] - top left x coordinate of the text

\* @arg {number} [y=textSize] - top left y coordinate of the text

\* @arg {number} [w=canvasWidth] - maximum width of drawn text

\* @arg {number} [lh=1] - line height

\* @arg {number} [method=fill] - text drawing method, if 'none', text will not be rendered

\*/

CanvasRenderingContext2D.prototype.drawBreakingText = function (str, x, y, w, lh, method) {

// local variables and defaults

var textSize = parseInt(this.font.replace(/\D/gi, ''));

var textParts = [];

var textPartsNo = 0;

var words = [];

var currLine = '';

var testLine = '';

str = str || '';

x = x || 0;

y = y || 0;

w = w || this.canvas.width;

lh = lh || 1;

method = method || 'fill';

// manual linebreaks

textParts = str.split('\n');

textPartsNo = textParts.length;

// split the words of the parts

for (var i = 0; i < textParts.length; i++) {

words[i] = textParts[i].split(' ');

}

// now that we have extracted the words

// we reset the textParts

textParts = [];

// calculate recommended line breaks

// split between the words

for (var i = 0; i < textPartsNo; i++) {

// clear the testline for the next manually broken line

currLine = '';

for (var j = 0; j < words[i].length; j++) {

testLine = currLine + words[i][j] + ' ';

// check if the testLine is of good width

if (this.measureText(testLine).width > w && j > 0) {

textParts.push(currLine);

currLine = words[i][j] + ' ';

} else {

currLine = testLine;

}

}

// replace is to remove trailing whitespace

textParts.push(currLine);

}

// render the text on the canvas

for (var i = 0; i < textParts.length; i++) {

if (method === 'fill') {

this.fillText(textParts[i].replace(/((\s\*\S+)\*)\s\*/, '$1'), x, y + (textSize \* lh \* i));

} else if (method === 'stroke') {

this.strokeText(textParts[i].replace(/((\s\*\S+)\*)\s\*/, '$1'), x, y + (textSize \* lh \* i));

} else if (method === 'none') {

return { 'textParts': textParts, 'textHeight': textSize \* lh \* textParts.length };

} else {

console.warn('drawBreakingText: ' + method + 'Text() does not exist');

return false;

}

}

return { 'textParts': textParts, 'textHeight': textSize \* lh \* textParts.length };

};

})(window, document);

var canvas = document.createElement('canvas');

var canvasWrapper = document.getElementById('canvasWrapper');

canvasWrapper.appendChild(canvas);

canvas.width = 500;

canvas.height = 500;

var ctx = canvas.getContext('2d');

var padding = 15;

var textTop = 'I use coding torque to learn';

var textBottom = 'web development by creating projects';

var textSizeTop = 10;

var textSizeBottom = 10;

var image = document.createElement('img');

image.onload = function (ev) {

// delete and recreate canvas do untaint it

canvas.outerHTML = '';

canvas = document.createElement('canvas');

canvasWrapper.appendChild(canvas);

ctx = canvas.getContext('2d');

document.getElementById('trueSize').click();

document.getElementById('trueSize').click();

draw();

};

document.getElementById('imgURL').oninput = function (ev) {

image.src = this.value;

};

document.getElementById('imgFile').onchange = function (ev) {

var reader = new FileReader();

reader.onload = function (ev) {

image.src = reader.result;

};

reader.readAsDataURL(this.files[0]);

};

document.getElementById('textTop').oninput = function (ev) {

textTop = this.value;

draw();

};

document.getElementById('textBottom').oninput = function (ev) {

textBottom = this.value;

draw();

};

document.getElementById('textSizeTop').oninput = function (ev) {

textSizeTop = parseInt(this.value);

draw();

document.getElementById('textSizeTopOut').innerHTML = this.value;

};

document.getElementById('textSizeBottom').oninput = function (ev) {

textSizeBottom = parseInt(this.value);

draw();

document.getElementById('textSizeBottomOut').innerHTML = this.value;

};

document.getElementById('trueSize').onchange = function (ev) {

if (document.getElementById('trueSize').checked) {

canvas.classList.remove('fullwidth');

} else {

canvas.classList.add('fullwidth');

}

};

document.getElementById('export').onclick = function () {

var img = canvas.toDataURL('image/png');

var link = document.createElement("a");

link.download = 'My Meme';

link.href = img;

link.click();

var win = window.open('', '\_blank');

win.document.write('<img style="box-shadow: 0 0 1em 0 dimgrey;" src="' + img + '"/>');

win.document.write('<h1 style="font-family: Helvetica; font-weight: 300">Right Click > Save As<h1>');

win.document.body.style.padding = '1em';

};

function style(font, size, align, base) {

ctx.font = size + 'px ' + font;

ctx.textAlign = align;

ctx.textBaseline = base;

}

function draw() {

// uppercase the text

var top = textTop.toUpperCase();

var bottom = textBottom.toUpperCase();

// set appropriate canvas size

canvas.width = image.width;

canvas.height = image.height;

// draw the image

ctx.drawImage(image, 0, 0, canvas.width, canvas.height);

// styles

ctx.fillStyle = '#fff';

ctx.strokeStyle = '#000';

ctx.lineWidth = canvas.width \* 0.004;

var \_textSizeTop = textSizeTop / 100 \* canvas.width;

var \_textSizeBottom = textSizeBottom / 100 \* canvas.width;

// draw top text

style('Impact', \_textSizeTop, 'center', 'bottom');

ctx.drawBreakingText(top, canvas.width / 2, \_textSizeTop + padding, null, 1, 'fill');

ctx.drawBreakingText(top, canvas.width / 2, \_textSizeTop + padding, null, 1, 'stroke');

// draw bottom text

style('Impact', \_textSizeBottom, 'center', 'top');

var height = ctx.drawBreakingText(bottom, 0, 0, null, 1, 'none').textHeight;

console.log(ctx.drawBreakingText(bottom, 0, 0, null, 1, 'none'));

ctx.drawBreakingText(bottom, canvas.width / 2, canvas.height - padding - height, null, 1, 'fill');

ctx.drawBreakingText(bottom, canvas.width / 2, canvas.height - padding - height, null, 1, 'stroke');

}

image.src = 'https://imgflip.com/s/meme/The-Most-Interesting-Man-In-The-World.jpg';

document.getElementById('textSizeTop').value = textSizeTop;

document.getElementById('textSizeBottom').value = textSizeBottom;

document.getElementById('textSizeTopOut').innerHTML = textSizeTop;

document.getElementById('textSizeBottomOut').innerHTML = textSizeBottom;

Final Output

[**Part 1: Prerequisites**](https://workshops.hackclub.com/meme_generator/#part-1-prerequisites)

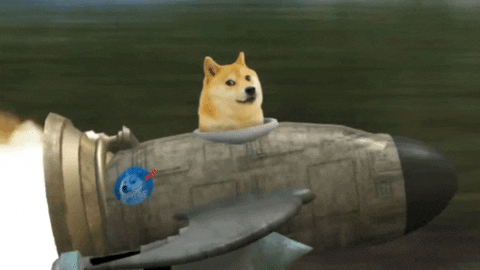
Basic knowledge of:

1. HTML
2. CSS
3. JavaScript

[**Part 2: Setting Up The Environment**](https://workshops.hackclub.com/meme_generator/#part-2-setting-up-the-environment)

We'll be using an online code editor called [repl.it](https://repl.it/) for this workshop.

To get started, go to <https://repl.it/languages/html>. Your coding environment will spin up in just a few seconds!



[**Part 3: Building The Project**](https://workshops.hackclub.com/meme_generator/#part-3-building-the-project)

[**1) HTML**](https://workshops.hackclub.com/meme_generator/#1-html)

Let's start by making a structure for our Meme Generator website.

**We'll be writing all our below code inside our <body> tag.**

* First, add a heading to the website using the <h1> tag:
* <h1>Welcome to Meme-Genie 🧞 </h1>
* Next, we want our memes to have two text fields: ***Top Text*** and ***Bottom Text***, something like this:



To achieve this we'll use two <textarea> tags after our <h1> tag:

<textarea id="top-text"></textarea>

<textarea id="bottom-text"></textarea>

Now we want "Text Size" sliders for both our top and bottom text. For this, we'll use the [<input type="range">](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range) HTML element. Add the following inputs under each textarea, like so:

<textarea id="top-text"></textarea>

Text size: <input type="range" id="top-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

<textarea id="bottom-text"></textarea>

Text size: <input type="range" id="top-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

Here, the min is the minimum text size, max is the maximum text size, value is the default value and the step argument is the stepping interval or the precision of the slider.

* Next, we need to take a meme template as our file input. For this we'll use <input> tags, specifying that we only want images as our file input:
* <input type="file" id="image-input" accept="image/\*">

This will add a button on our website that'll let the user input a meme template as an image.

* Next, we need a Generate button:
* <button id="generate-btn">Generate!</button>
* But we need something to display our meme, right?

For this, we'll use an [HTML5 Canvas](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API):

<canvas id="meme-canvas" title="Right click to save this meme"></canvas>

Our HTML code so far:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Meme Generator</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Welcome to Meme-Genie 🧞 </h1>

<textarea id="top-text"></textarea>

Text size: <input type="range" id="top-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

<textarea id="bottom-text"></textarea>

Text size: <input type="range" id="bottom-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

<input type="file" id="image-input" accept="image/\*">

<button id="generate-btn">Generate!</button>

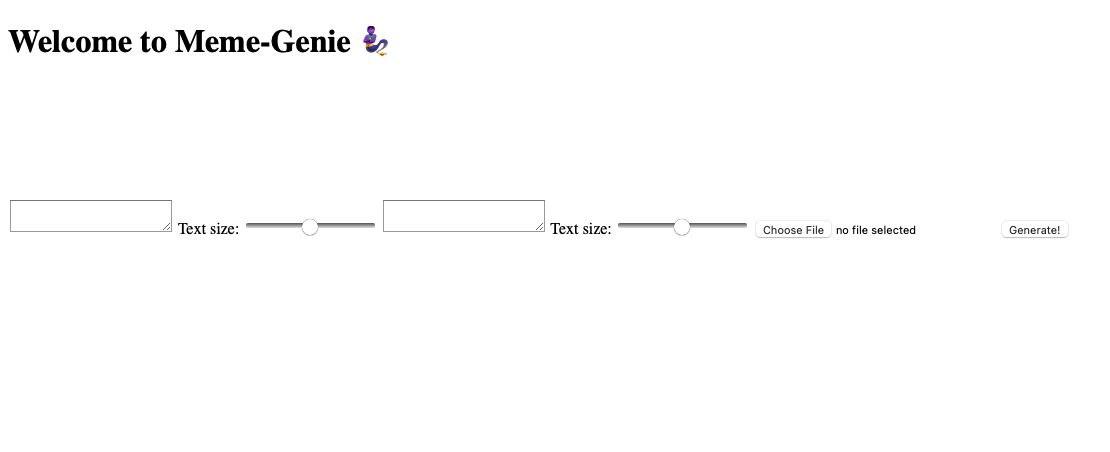
<canvas id="meme-canvas" title="Right click to save this meme"></canvas>

<script src="script.js"></script>

</body>

</html>

If you run the website now by clicking the green "Run" button at the top, you'll notice that the elements look kinda weird, something like this:



To fix this problem, we have the mighty <div> tag which we'll be using to seperate out the different elements!

So, wrap all the different elements inside the <div> tags, like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Meme Generator</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Welcome to Meme-Genie 🧞 </h1>

<div>

<textarea id="top-text"></textarea>

Text size: <input type="range" id="top-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

</div>

<div>

<textarea id="bottom-text"></textarea>

Text size: <input type="range" id="bottom-text-size-input" min="0.05" max="0.25" value="0.15" step="0.01">

</div>

<div>

<input type="file" id="image-input" accept="image/\*">

</div>

<div>

<button id="generate-btn">Generate!</button>

</div>

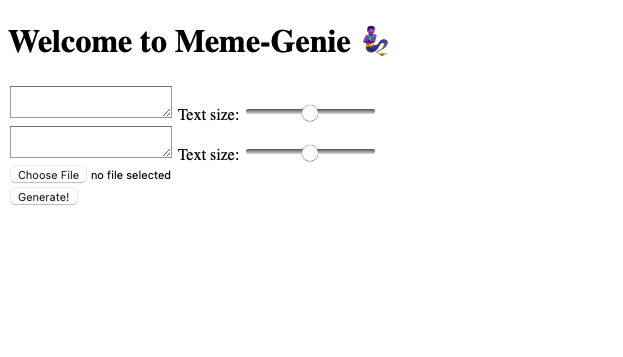
<canvas id="meme-canvas" title="Right click to save this meme"></canvas>

<script src="script.js"></script>

</body>

</html>

Now run the website again. It should look something like this:



And with this we've completed the HTML part!

[**2) CSS**](https://workshops.hackclub.com/meme_generator/#2-css)

Next, let's add some basic styling to our website. Navigate to the style.css file and add the following code:

h1 {

font-family: Impact, 'Arial Narrow Bold', sans-serif;

font-size: 30px;

}

body {

margin: 10px;

background-color: lightblue;

}

#meme-canvas {

width: 300px;

}

In this CSS file:

* We first change the heading's font to Impact (the "meme font"), with a fallback to Arial if the user doesn't have Impact for some reason. Then, we make the text of the heading a little bigger.
* Then, we apply a background color to the body (AKA to the whole page) and add a bit of space around the edges with margin.
* Then, we select our canvas (which, if you recall, we gave an id meme-canvas) using # and set its width to 300px.
  + In CSS, when you select an element, like h1, it will apply the styles you specify to all h1 tags on the page. So if you were to add another heading, it would make it 30px and Impact font as well. However, in the case of meme-canvas, we're telling the CSS file to only select that specific canvas, so if we were to add another canvas at some point, it wouldn't apply that style to that canvas.
  + Why are we selecting that ID if we're not making another canvas? So that you can learn about selecting IDs! :)

That's it! This is all the CSS we'll be doing in the workshop.

[**3) JavaScript**](https://workshops.hackclub.com/meme_generator/#3-javascript)

You're doing great! Now, let's actually make our meme generator work with JavaScript.

Navigate to the script.js file and:

* Make a generateMeme() function which takes in parameters img, topText, bottomText, topTextSize, and bottomTextSize:
* function generateMeme(img, topText, bottomText, topTextSize, bottomTextSize) {
* *// Code here*
* }

Now, inside the generateMeme() function:

* 1. First, initialize a [Canvas](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_usage), which will be pointing to our meme-canvas element from our HTML code:

const canvas = document.getElementById('meme-canvas');

Here, we're doing something pretty cool: we're selecting the specific canvas that we wrote in the HTML file with ID meme-canvas, and we're going to modify it right in our JavaScript code!

* 1. Now, to get canvas' [2D rendering context](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D), we'll call getContext() by selecting out <canvas> element from HTML by its ID, and supplying '2d' as the argument. Our 2D rendering context will provide us with a lot of methods to stylize and draw on our canvas!

const ctx = canvas.getContext('2d');

* 1. Next, we set the canvas' dimensions same as our meme image's dimensions:

canvas.width = img.width;

canvas.height = img.height;

* 1. Now, using our 2D rendering context, we clear out a rectangle, by erasing the pixels in a rectangular area by setting them to transparent black:

ctx.clearRect(0, 0, canvas.width, canvas.height);

* 1. After that, we draw the image on the canvas by using the drawImage() function provided by context and supplying our image, and the X and Y coordinates.

ctx.drawImage(img, 0, 0); *// 0, 0 are our X and Y coordinates*

* 1. Next, we set the style of our text using three functions ctx.fillStyle, ctx.strokeStyle and ctx.textAlign:

*// Text style: white with black borders*

ctx.fillStyle = 'white';

ctx.strokeStyle = 'black';

ctx.textAlign = 'center';

* 1. Now, we set our fontSize:

let fontSize = canvas.width \* topTextSize; *//Font Size will change based on our input sliders*

ctx.font = `${fontSize}px Impact`; *// We'll be using Impact font, which is used by most memes*

ctx.lineWidth = fontSize / 20; *// lineWidth will be the outline of our text, and we're setting it to be 20th of our fontSize here.*

* 1. Next, to draw top text on our meme image, we use ctx.fillText to fill text and ctx.strokeText for outlines:

*// Draw top text*

ctx.textBaseline = 'top'; *// textBaseline property specifies the current text baseline used when drawing text.*

topText.split('\n').forEach((t, i) => {

ctx.fillText(t, canvas.width / 2, i \* fontSize, canvas.width); *// fillText takes 3 arguments: first is our text, second and third arguments are our X and Y coordinates of the point at which to begin drawing the text.*

ctx.strokeText(t, canvas.width / 2, i \* fontSize, canvas.width); *// Arguments are same as fillText but strokeText draws outlines on our text.*

});

* 1. Now, we repeat the same steps for bottom text:

*// Bottom text font size*

fontSize = canvas.width \* bottomTextSize;

ctx.font = `${fontSize}px Impact`;

ctx.lineWidth = fontSize / 20;

*// Draw bottom text*

ctx.textBaseline = 'bottom';

bottomText.split('\n').reverse().forEach((t, i) => { *// .reverse() because it's drawing the bottom text from the bottom up*

ctx.fillText(t, canvas.width / 2, canvas.height - i \* fontSize, canvas.width);

ctx.strokeText(t, canvas.width / 2, canvas.height - i \* fontSize, canvas.width);

});

} *// End of our generateMeme() function*

* Now, after the generateMeme() function, add a [Window: DOMContentLoaded](https://developer.mozilla.org/en-US/docs/Web/API/Window/DOMContentLoaded_event) event listener which will listen for DOMContentLoaded event.
* window.addEventListener('DOMContentLoaded', () => {
* *//Code here*
* });

The code inside here will only be run after all of the elements on the page are loaded. We need this because we're about to select a *bunch* of elements on the HTML page, and if we didn't have it, it's possible that the code could try to select elements before they exist.

Inside the EventListener:

* 1. Initialize our topTextInput, bottomTextInput, topTextSizeInput, bottomTextSizeInput, imageInput variables using document.getElementById() method:

const topTextInput = document.getElementById('top-text');

const bottomTextInput = document.getElementById('bottom-text');

const topTextSizeInput = document.getElementById('top-text-size-input');

const bottomTextSizeInput = document.getElementById('bottom-text-size-input');

const imageInput = document.getElementById('image-input');

const generateBtn = document.getElementById('generate-btn');

* 1. Next, we give a default text (or value) for the topTextInput and bottomTextInput:

topTextInput.value = 'Top\nValue';

bottomTextInput.value = 'Bottom\nValue';

* 1. After that, add a click event listener to the generateBtn. The code inside here will run every time that button is clicked.

generateBtn.addEventListener('click', () => {

*// Code here*

});

* 1. Inside the EventListener, we'll use the [FileReader API](https://developer.mozilla.org/en-US/docs/Web/API/FileReader) to read the meme template from the filesystem as a [DataURL](http://developer.mozilla.org/en-US/docs/Web/API/FileReader/readAsDataURL):

const reader = new FileReader();

reader.readAsDataURL(imageInput.files[0])

* 1. Now, after the FileReader finishes reading the meme template we want to generate a new [Image](https://developer.mozilla.org/en-US/docs/Web/API/HTMLImageElement/Image), and that image's source will be our reader's result. After our new image is loaded, we'll pass our new image, text and text sizes to the generateMeme() function where it'll be displayed on our canvas!

reader.onload = () => {

const img = new Image();

img.src = reader.result;

img.onload = () => {

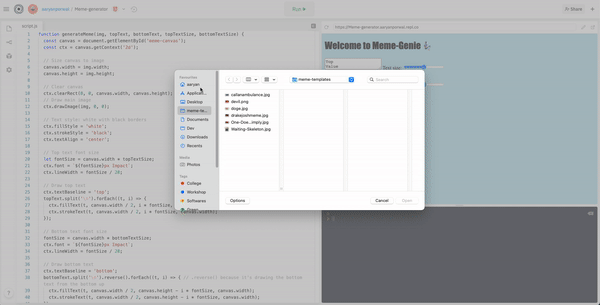
generateMeme(img, topTextInput.value, bottomTextInput.value, topTextSizeInput.value, bottomTextSizeInput.value);

};

};

Final JavaScript code:

***And with this, we've finished coding our Meme-Genie! Check out what you've just built by clicking the "Run" Button at the top.***



Note: To save the generated meme, you can right click the meme and click "save as..."

[**Part 4: Hacking**](https://workshops.hackclub.com/meme_generator/#part-4-hacking)

Now that we have finished building a very basic website, we can add a lot of cool things to it!

Here are some things you can try to add:

1. Try to add [colour picker](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/color), so that user can change topText/bottomText colour.
2. Add a drag and drop feature, using the [HTML5 Drag and Drop API](https://web.dev/drag-and-drop/).
3. Make a [Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox) to create a nice layout.
4. Use an API to fetch meme-templates, like the one provided by [ImgFlip](https://imgflip.com/api).
5. Add Share and Download buttons, to share your super awesome meme!

[**Part 5: The End**](https://workshops.hackclub.com/meme_generator/#part-5-the-end)

If you haven't created an account on [repl.it](https://repl.it/), make sure you do so to save this wonderful creation!

If you're having trouble signing up, ask your club leader or someone on the [Hack Club Slack](https://hackclub.com/slack)!

[**Some More Examples**](https://workshops.hackclub.com/meme_generator/#some-more-examples)

* [Ninivert](https://codepen.io/ninivert/pen/BpLKRx)
* [Vox Media's Meme Generator](https://github.com/voxmedia/meme)
* [Salt Bay Meme Generator](https://codepen.io/yelly/pen/demrxp)

[**What Other Hack Club Hackers Made**](https://workshops.hackclub.com/meme_generator/#what-other-hack-club-hackers-made)

* [Sam Poder](https://beautiful-languid-passive.glitch.me/) used the Unsplash API to get an image based on the Bottom Text!
* [Khushraj Rathod](https://repl.it/@aaryanporwal/MemeGenie#indexhtml) built a meme generator that uses the ImgFlip API to fetch random meme templates!

[**Part 6: Sharing with the Community**](https://workshops.hackclub.com/meme_generator/#part-6-sharing-with-the-community)

