

# Build a Basic Calculator App

This project guides you through building a functional calculator using HTML, CSS, and JavaScript. We'll cover creating the user interface, capturing user input, performing calculations, and displaying the results.

 by Nishtha Yadav

```
testtemeralt.terl.  
ar tterferelt.tultesrthtarEd;  
ftcefttreltnesittibt(nced;  
ftcrtitcereltttucerd)  
ftcefiteltelttitbirc;  
vtorfitceretemaettCd(bemd;  
fitcreterelltittttice(Car :
```

# Introduction to JavaScript Arrays and Loops

## 1 Arrays

Arrays are data structures that store collections of related values. They allow us to organize data efficiently and access individual elements.

## 2 Loops

Loops enable us to repeat a block of code multiple times, allowing us to process each element in an array or perform operations iteratively.

## 3 Combining Arrays and Loops

We will leverage arrays to store the calculator's buttons and loops to dynamically create their HTML representations.

# Creating the HTML Structure

## Calculator Layout

The calculator will have a display area to show user input and results. Below, we'll arrange buttons for numbers, operators, and the equal sign.

## HTML Structure

We'll use HTML elements like `<div>` and `<button>` to construct the calculator's visual elements.

## Button Organization

We'll arrange the buttons in a logical order, using CSS to style their appearance and layout.

# Calculator

## Styling the Calculator with CSS



### Color Palette

Select a consistent color scheme for buttons, text, and the display area.



### Typography

Choose a legible font for the display and buttons, ensuring clear readability.



### Layout and Spacing

Use CSS properties like padding, margin, and flexbox to create a visually appealing and user-friendly layout.



### Button Hover Effects

Add interactive hover effects to buttons, such as changing background color or adding a shadow.

# Capturing User Input in JavaScript

1

## Event Listeners

Add event listeners to each button to capture user clicks.

2

## Input Handling

Store the clicked button's value in a variable or update the display area.

3

## Operator Detection

Identify when an operator button is clicked and prepare to handle the calculation.

```
fatgatens.tor
scefifat
totuter evteht(listem);
conserttinal-battonsfd;
forcabur llst fhtt;
calcaltc alttut;
teecentfisteletceriresfTat;
calecttc stettngs litem)tat(1ldb;

inscersfblattarts);
fcecentitesstall; fext;
call:tistentitcater;
feacase scetthes trceplor),
taltcritatc CaWCatgFelet)..
tcelco.fbte;
taltcattefule*?11#;
callccesteretts:"llet;
teme:

freceerstisterfblsttens#Tatt
fartetscunl tatecstatehcltithilcator;
treefrettatscl:tex:

callentatke
tratffistcilisttert(0.ton() "1lat)
tretashemetritterapt(:5l0;
```

# Performing Arithmetic Operations

1

## Function Definition

Create functions for each arithmetic operation (addition, subtraction, multiplication, division).

2

## Operand Storage

Store the two numbers that the user wants to operate on.

3

## Calculation Execution

Perform the selected operation using the stored operands and return the result.

```
f= deceporte : <7;
caterscfocs: 80; 0;
uaccrarfec(4Postinl;
uscroorioc(ound:= 1-:0; ><0()9;
andetive (= 63.0)
apotticlen: 49); .0:1);

ditecrricc(===3;
crctiile(fiteljædd3);
decroried(ion(= 10; {} ==42)
orcroried(ite(= 448)
uccroricc(==9));
practiled(itct( ==< {, Ance (4:1.); fres (2(7),:) }

oractile {; 08c120 {
  Rescitt(: <)
  ccreitd: well darly.>2); }
  coreption(-wll; ==98)
  corectis wit(==17;
  = {;

};

:*)
;=
```

# Displaying the Result

Method	Description
innerHTML	Update the content of the display area with the calculated result.
textContent	Set the text content of the display area to the calculated result.

# CaLcuL to

1234.56

1

×

+



# Conclusion and Next Steps

## Congratulations!

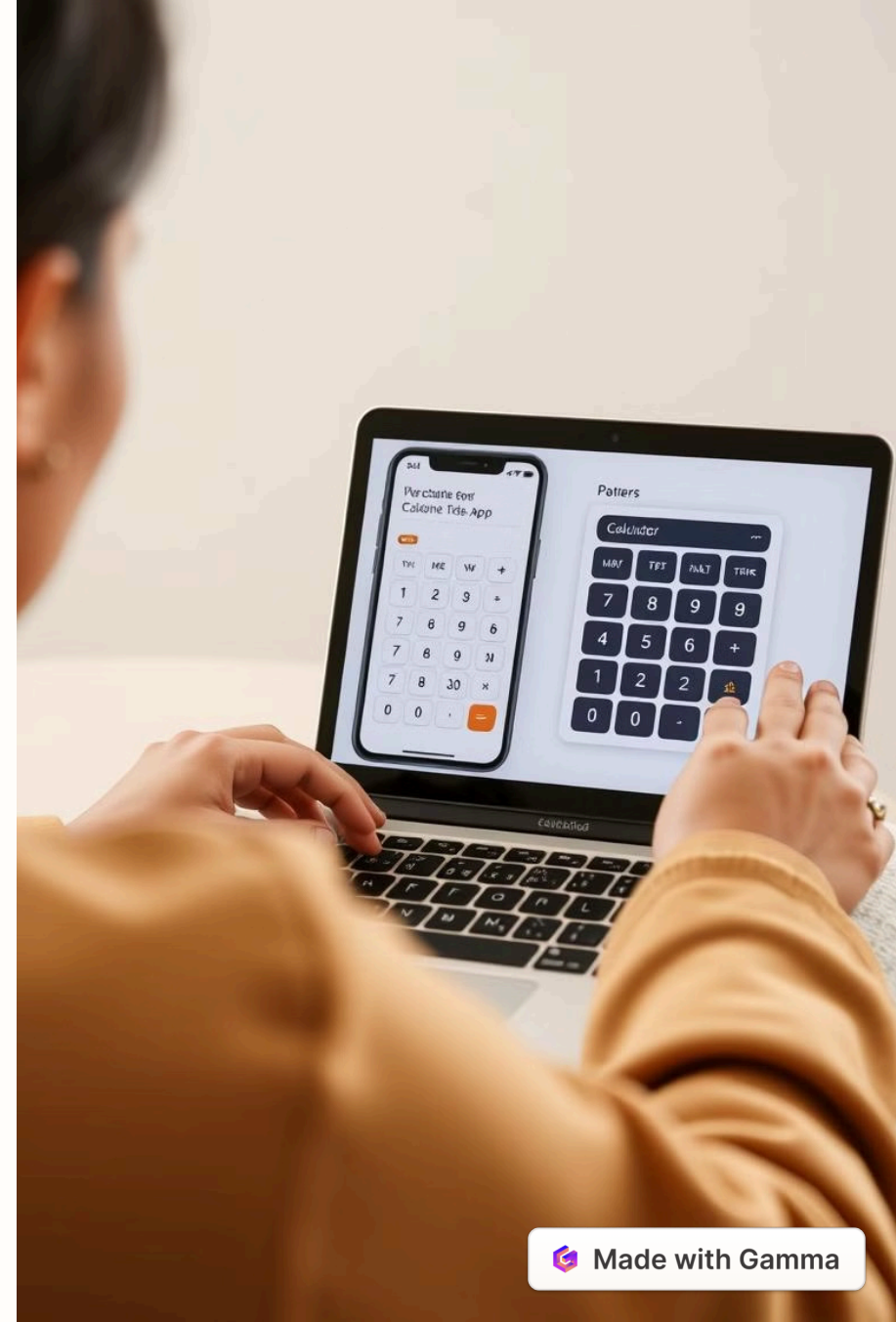
You've successfully built a functional calculator using JavaScript, HTML, and CSS.

## Additional Features

Consider adding features like memory storage, scientific functions, or a history log.

## User Experience

Focus on improving the calculator's usability and responsiveness for a better user experience.





NAME:NISHTHA YADAV

SECTION: C

BRANCH: CS

ROLL NO:2300290120150

REPO LINK:

[https://github.com/Nishthayadav20/MLSAKIET\\_  
FWdevops\\_INTERNSHIP.git](https://github.com/Nishthayadav20/MLSAKIET_FWdevops_INTERNSHIP.git)