

# Coming soon! Image gallery from conception to deployment

By Dr. Holger Flick

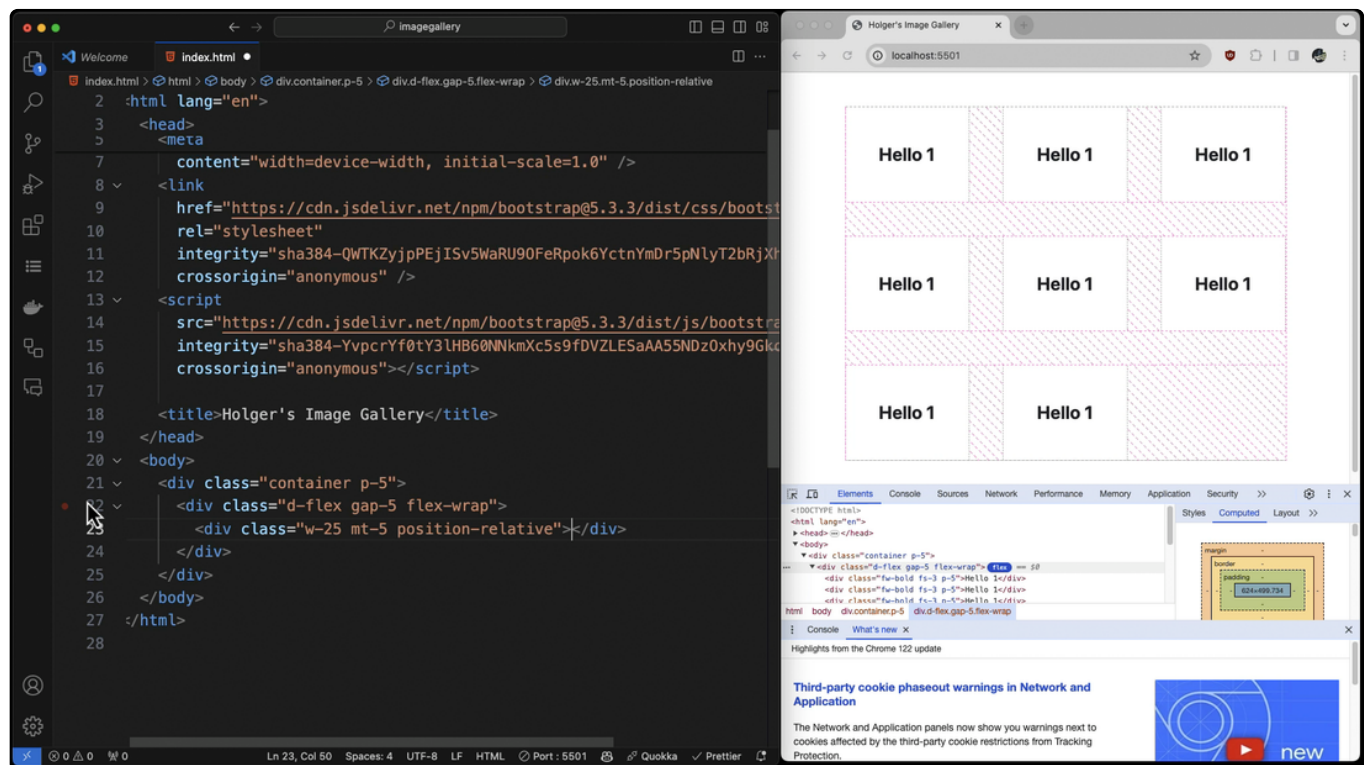
In this upcoming video series, I will show you how to create a modern, responsive image gallery with TMS WEB Core. In addition, you will learn how to integrate the Pexels API. Starting with basic principles of web development using cascading stylesheets (CSS) including a detailed introduction of flex box, the new grid system, and cards. Floats are a thing of the past! We will also use the Bootstrap framework to create a responsive design. With regard to functionality, we will begin just to show a few curated images from one endpoint. At the end, you will create a web application that allows you to search for specific keywords and display the images in a grid with the ability to navigate between pages (pagination).

## Sponsor

If you love these video tutorials, [please consider sponsoring me on GitHub](#). Your support will help me to create more content like this. Thank you very much!

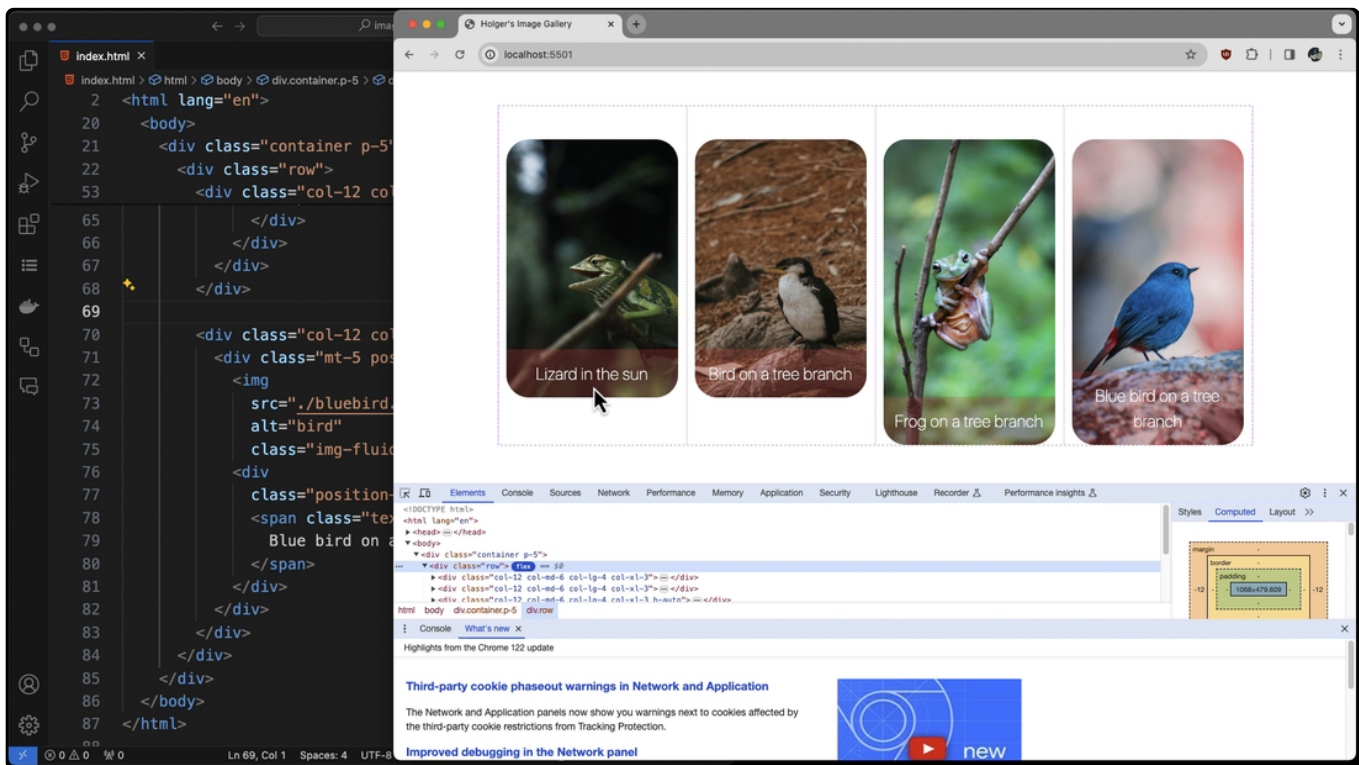
## Summary

- Basic concepts of CSS layout are explained in detail:



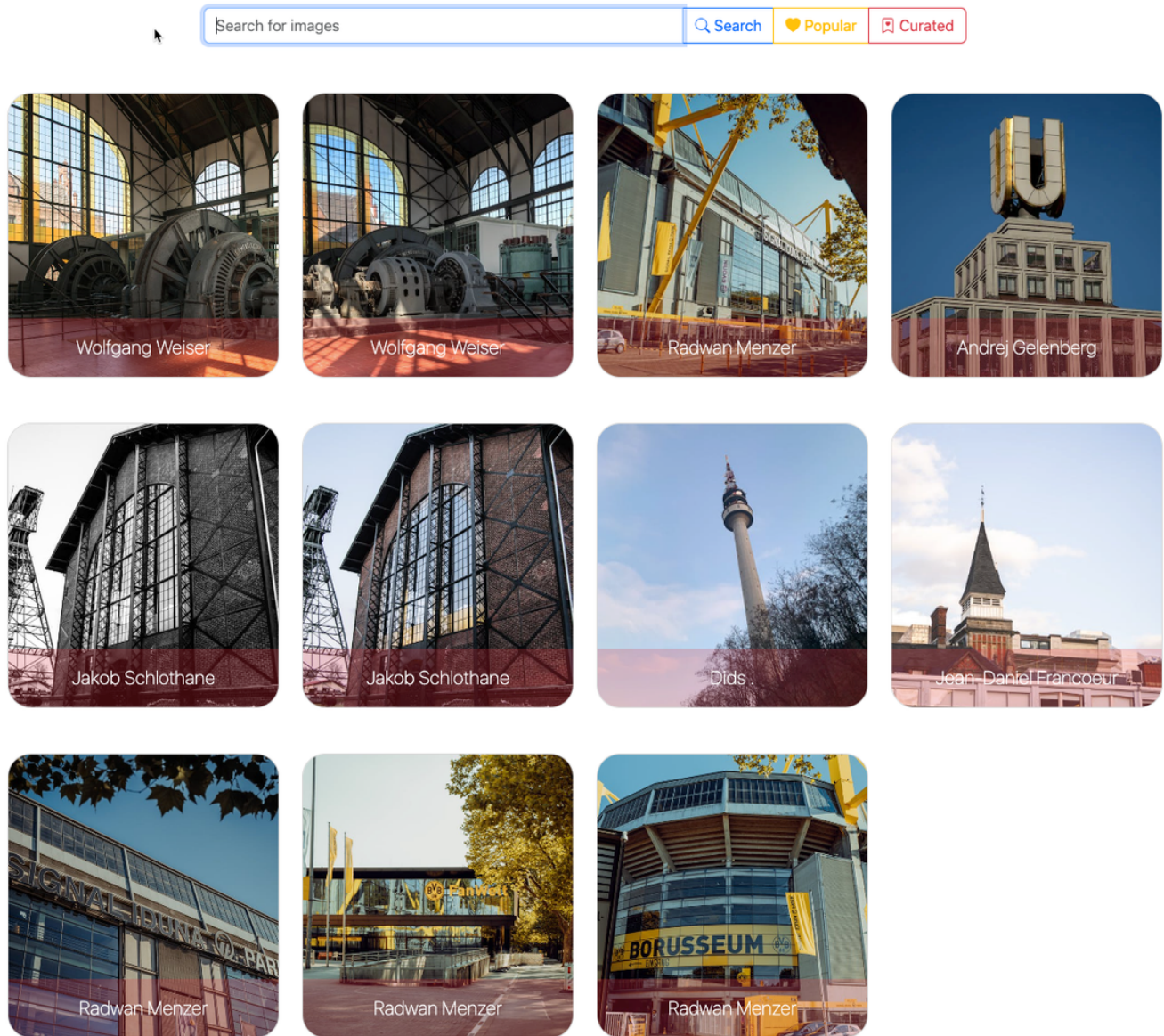
Layout basics

- Looking at ins and outs of layout:



Ins and outs of layout

- The finished "product" showing images from the Pexels API in a gallery with pagination and search functionality:



Finished image gallery in TMS WEB Core

## Thank you to the sponsor!

This video series will be sponsored by [TMS Software](https://www.tmssoftware.com). Their enthusiasm for Delphi made it possible for me to use all their products used in this tutorial and build real-world use cases for the community. They make it possible to offer this multi-part video series for free on YouTube.

<figure markdown> [!TMS Software](https://www.tmssoftware.com)</figurecaption> </figure>

# Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

---

## Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

## Ways to support

KO-FI

### Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

### Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

### Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

---

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

[Support on Ko-fi —](#)