Poyo's guide to retro web development!

Poyo!

v0.2

Contents

Prologue (By WoepdieCat)

I trust Poyo! to take this topic seriously and show you what websites are really all about and the magic behind them

PD: He won't make it seriously, will he? Hahaha

Brief introduction to the book

Howdy, internaut! If you're reading this book, I assume you're a nostalgic peep who also wants to build a website reminiscent of the 90s. However, you may have tried to build one, and have discovered that it's challenging to create such beautiful pieces of art. That's why I'm here to help you. :D

We will also dive on the trends of the 1990s-2000s, they are really important to understand how can we replicate them. There were many trends back in the 90s, such as having your visitors sign a guestbook, visitor counters, and blinkies/buttons(More on that later), and believe it or not, Comic Sans was all the rage worldwide. Yep O-O, Comic Sans. What the fu-

Many of these trends originated because search engines were bad. Really bad. So people linked their websites by entering on Webrings and by sharing Buttons with their friends. Buttons are like souvenirs from websites. These can be put on other's websites to show they liked the website, or that they're friends of their corresponding webmaster. We'll also talk about how to create a button yourself in this book.

In this book, I will teach you how to build your own website from scratch, what were the trends in the 1990s-2000s and how can we replicate the looks of such websites. Without further ado, let's get started!

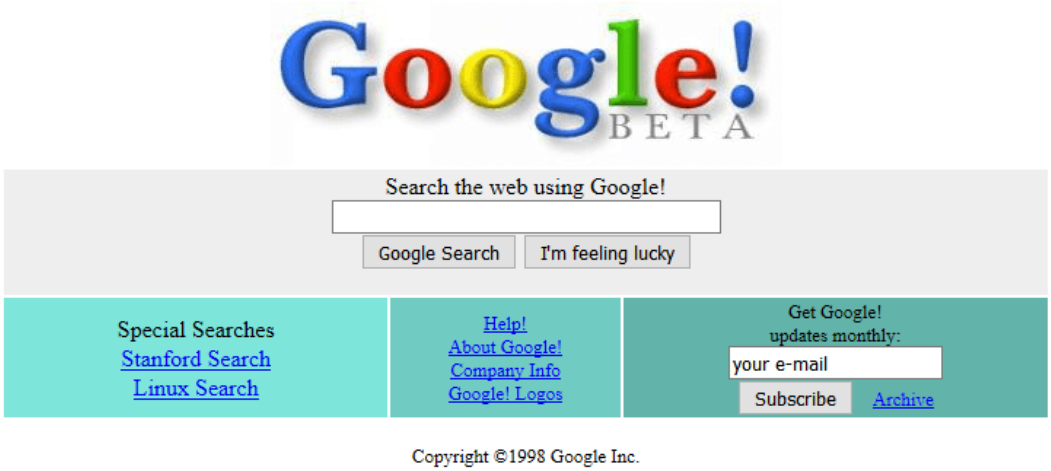

Figure 0.1: This is how google looked in 1998. Image taken from https://www.webdesignmuseum.org/exhibitions/web-design-in-the-90s

Chapter 1: VScode and Web Development Tools

In this chapter, we will explore the powerful combination of Visual Studio Code (VScode) and various web development tools that will be very useful! ## 1.1.1 Getting Started with VScode

VScode is a popular code editor developed by Microsoft. It provides a wide range of features and extensions that make it a favorite among many web developers. To get started with VScode[1], it's as simple as visiting their official website[2] and downloading VScode. Afterwards, make sure to run the installation wizard and you're now good to go!

1.1.2 Essential Extensions for Web Development

VScode offers a huge collection of extensions that enhance the web development experience. Here are some essential extensions to consider:

1. **HTML CSS Support**: Provides autocompletion and syntax highlighting for HTML and CSS.
2. **Live Server**: Launches a local development server and automatically refreshes the browser whenever you make changes to your HTML, CSS, or JavaScript files. I personally love this extension because it makes web development so easy, that I can finish websites in a matter of minutes.
3. **Prettier**: Prettier automatically formats your code to ensure consistent styling and readability.
4. **Rainbow Indent**: It helps a lot with indenting and if you indent correctly, it will display a rainbow! :) Cool, isn't it?
5. **Live Share**: This extension allows you to share your local machine with your peers, so you can work collaboratively in a single dev environment! You can be editing one file while your peer edits another one. It's a must-have extension in my opinion.

1.1.3 Collaboration and Remote Development

VScode enables seamless collaboration and remote development. Here are some features to consider:

1. **Live Share**: With Live Share, you can share your development environment with others, allowing them to edit and debug code in real-time.
2. **GitHub Codespaces**: GitHub offers a service called Codespaces, which provides a cloud-based development environment with VSCode pre-installed. With Codespaces, you can access a Linux machine from anywhere in the world, making it convenient for remote development.

---

[1] Official video-guide: https://www.youtube.com/watch?v=B-s71n0dHUk

[2] You can get VScode from https://code.visualstudio.com/.

1.1.4 How to use VSCode?

1. **Opening a Project**: Once you have installed VSCode, open it and you will see the welcome screen. From there, you can either open an existing project or create a new one. To open an existing project, click on 'Open Folder' and select the folder containing your project files.
2. **Editor Layout**: The VSCode interface consists of several components. The *editor groups* are the editor, where you write your code. On the left side, you have the *activity bar*, which provides access to different views like the file explorer, source control, extensions, and more. At the bottom, you have the *status bar*, which displays information about the current file, a handy terminal and it also provides quick access to various settings.

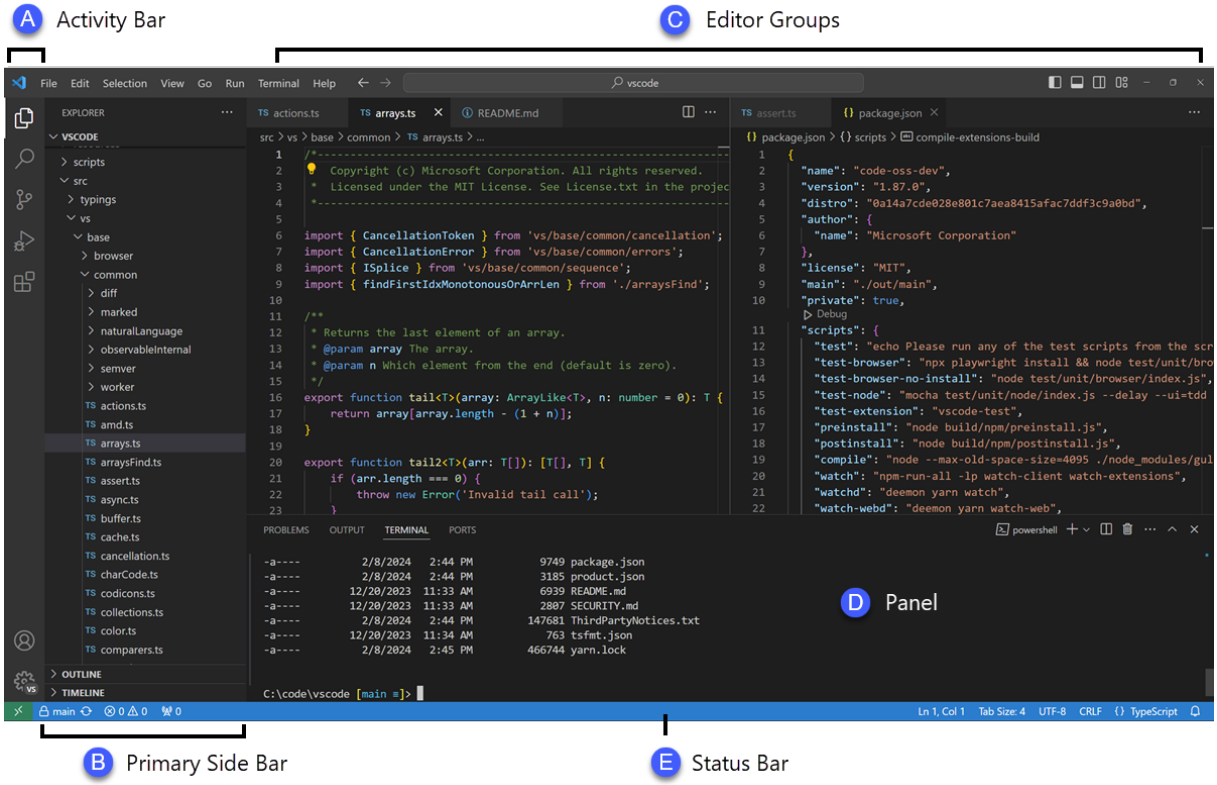

Figure 0.2: VScode components

3. **Terminal**: VScode has an integrated terminal to make coding easier. You can access it by opening the bottom bar. But be careful! It will be a live terminal running on your computer. I suggest doing all of your changes in a single folder and then, once you're finished, move all of the files inside to the location where you'll be running them.

1.2 Git

1.2.1 What is Git?

Figure 0.3: Git's logo

Git is a widely used version control system. It's useful for having and mantaining different versions of your code, and to store them in GitHub for free. Git works by branches. Textually from Git's docs:

> Branching means you diverge from the main line of development and continue to do work without messing with that main line.

What does that mean? In Git there is a *main* branch, and then you can create more branches to independently write code/features for your project without messing up the *main* branch, the current *working* code. Once you finish coding, you can pull the branch to *main* Thankfully, VScode offers excellent integration with Git, which means we won't have to ever use any Git commands while using VScode. Let's see how can we use it!

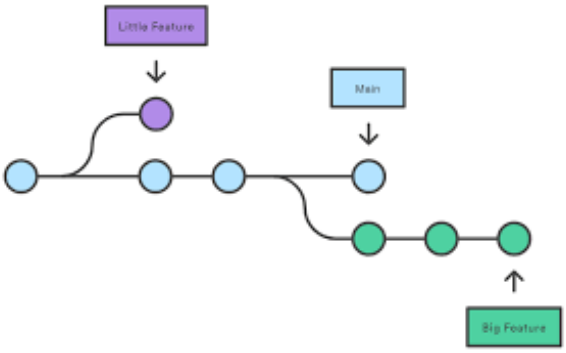

Figure 0.4: An example of a Git repository with its *main*, *big feature* and *little feature* branches.

How to use git?

Git is a command line tool. That means it can be accessed and used via the terminal. The main Git commands[3] are:

1. To authenticate yourself:

```
git config --global user.name "<username>"
git config --global user.email <email>
```

2. To start developing:
    - `git init` - Create a new repository
    - `git clone <repository url>` - Download a copy of a repository to commit changes or just to use the code.
3. To save your work
    - `git add .` - Add all the files changed to the commit.
    - `git commit -m "<Commit message>"` - Commit changes to the head branch of your git repo(repository)
    - `git push` - Send the changes and store them in the github repo

The github code to authenticate and retriev, we explored the powered how to get started with VScode, essential extensions for web development, debugging and testing capabilities, version control with Git, and collaboration features. In the next chapter, we will learn about the basics of building websites. See you there!

[3] Full list of commands available at https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html