The web as craft or the web as assembly line?

Doing web development used to be simple. All we needed was a text editor and either good knowledge of HTML, CSS and Javascript or bookmarks on your browser pointing to Eric Meyer's CSS resources on the CWRU server and links to the best tutorials you could find about CSS and Javascript with the differences for each browser and how to branch the code to acommodate each browser's idiosyncracies.

Those days are long gone. Reading David Rupert's The tangled webs we weave reflected some of my own struggles when working with existing projects.

But even when working on our projects, starting a new project forces us to ask many questions before we even get started.

Are we using Typescript? (and do we need to?) Are we using a build system? (if so which one?) If we're not using Typescript what version of the language are we using? Are we transpilling it so that it'll work with ES5?

If you build web content on a regular basis you may have internalized this and have a set of scripts ready to run and generate the skeleton of a website project ready to go with all the tools you know you will use.

But what happens to people new to frontend, backend, or full stack development or to people who have left the field for a while and now return to a drastically changed development landscape?

Two interesting counterpoints on this issue are Jessica Joy Kerr's Back when software was a craft and Justin Searls Twitter thread

Either research all the alternatives and pick the ones you think may work best for the project (if the project is yours) or learn the new tools the project has implemented, whether you agree with them or not.

- Everything Easy is Hard Again — Frank Chimero
- What Screens Want — Frank Chimero
- The Web's Grain — Frank Chimero
- Systems, Mistakes, and the Sea

- The arc of design