**Jeremy Keith – Weekly Nerd**

Music Design, Musical Composition, contrast the approaches.

One approach is **CLASSICAL** music. (One has an entire sheet).

Other approach is **JAZZ** music. (One has one interval, a key if you will).

**JAZZ** and **MUSICAL** approach

**Programming** languages can also be approached in different way.

Imperative programming = You tell the computer **exactly** what to do, instructive.

Declarative Programming = You yourself specify the language yourself and the computer responds to it.

**SQL** is Imperative.

We have programming languages on the world wide web like HTML. It is **not** imperative.

Imperatives are more universal. HTML is for structuring therefore declarative. CSS is also **not** imperative. Every line of CSS you write in CSS is a **suggestion**. Therefore declarative.

JavaScript is our first **imperative** language. With JavaScript you’re not bound to a specific vocabulary. XML is a declarative language. If you make a mistake in JS, it all freezes and you can’t do nothing. While HTML and CSS keep going. CSS and HTML are more domains bound and easy forgiving, easier to learn.

**Difference between imperative and declarative. Difference in mindset. You’re possible to have both.**

**BUTTON-component,** by simply writing the tag in HTMl. You can obviously create a div and add JS code to it with handlers, functions, etc.

*“JavaScript should only do what only JavaScript can do.”*

JavaScript in terms of the client-side, visible in the browser.

There are actual people out there who actually program a button despite the fact that there’s a functional button, available through HTML. The reason why people *do* this, is to have feeling of control. Handing that control to the browser can come with bugs, unlike when you’re responsible for your own code. Having control is a priority to some programmers. Creating it simply through the declarative format can come with a few cons regarding to the limits.

Regarding the CSS, if you were to change the font-size to 1rem, it would still be 16 pixels. The thing with CSS is how you specify the matter. Padding-left, would that add padding from the text, or the actual start of the button?

Instead of dictating with one font-size, you could calculate it regarding its user. With Clamp you can have the first value being the minimum value, while the last one is the maximum value. Still living by your own giving values. It gives the idea that human and machine can work together. A few examples are:   
  
calc

Clamp

Min and max

Fit-content

Min-content and max-content

Every-layout.dev

Talks about the boundaries of the user and understanding the capabilities of the machine.

https://buildexcellentwebsit.es/

Be the browser’s mentor, not it’s micromanager.

***CSS is the Bicycle of design.***

*Is a declarative design better than imperative?*

***IT DEPENDS…***

***On the culture of the environment of the designer’s company.***

* *On the management, on both the declarative and imperative.*

***On design systems, is HOW those components get made.***

* *The design system is the way someone does around their workspace.*
* *How do* ***we*** *design the system?*

***On how we think***

* *Do we think too analytic? Is that the way how to approach a problem?*
* *System thinking, how does the system think.*

**Design System**, writing down the color codes for instance. If a developer has an imperative mindset, that this works like a charm.

*“The border should be 10% lighter than the background color.”*

*Are declarative Design Systems better than imperative?*

**Print**

**Native Apps**

**Os-specific**

On the world wide web, there’s always been a battle of control. When were first designing the browsers and all, it was even a battle on who got to decide the width. Control, trying to exercise **control**. The more you do it, the less it work.

“The more you tighten your grip, the more the World Wide Web slips through your fingers.”