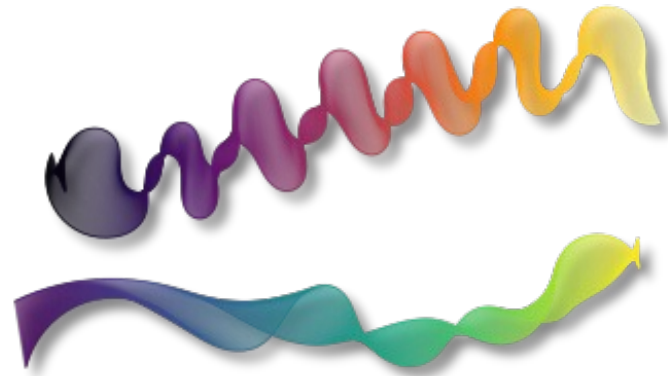


CSS: Frameworks and Tools

Introduction to CSS for beginner
for Personal Educational Purpose.

Jakarta, Indonesia
11 October, 2020



About Epsi



Yet, another underachiever.
But hey, I have my own blog.

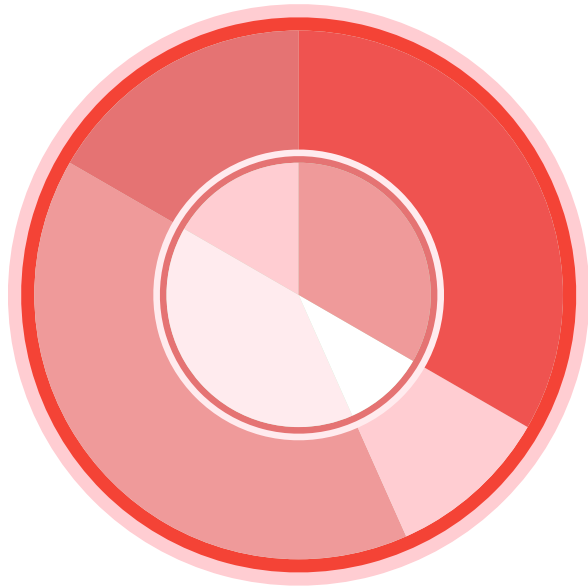
SORRY FOR MY ENGLISH

Frontend: HTML, CSS, JS

What is this rubbish?

A page is a document consist of:

More Reference: [[va_lesson4.pdf](#)]



Structure

HTML Tag: Structure and Formatting
structure = elements in hierarchy fashioned

Presentation

Stylesheet: Rule, Layout, Box Model
rule = selector + declaration
box model, position
layout: flex, grid, float+calc, table

Behaviour

Javascript

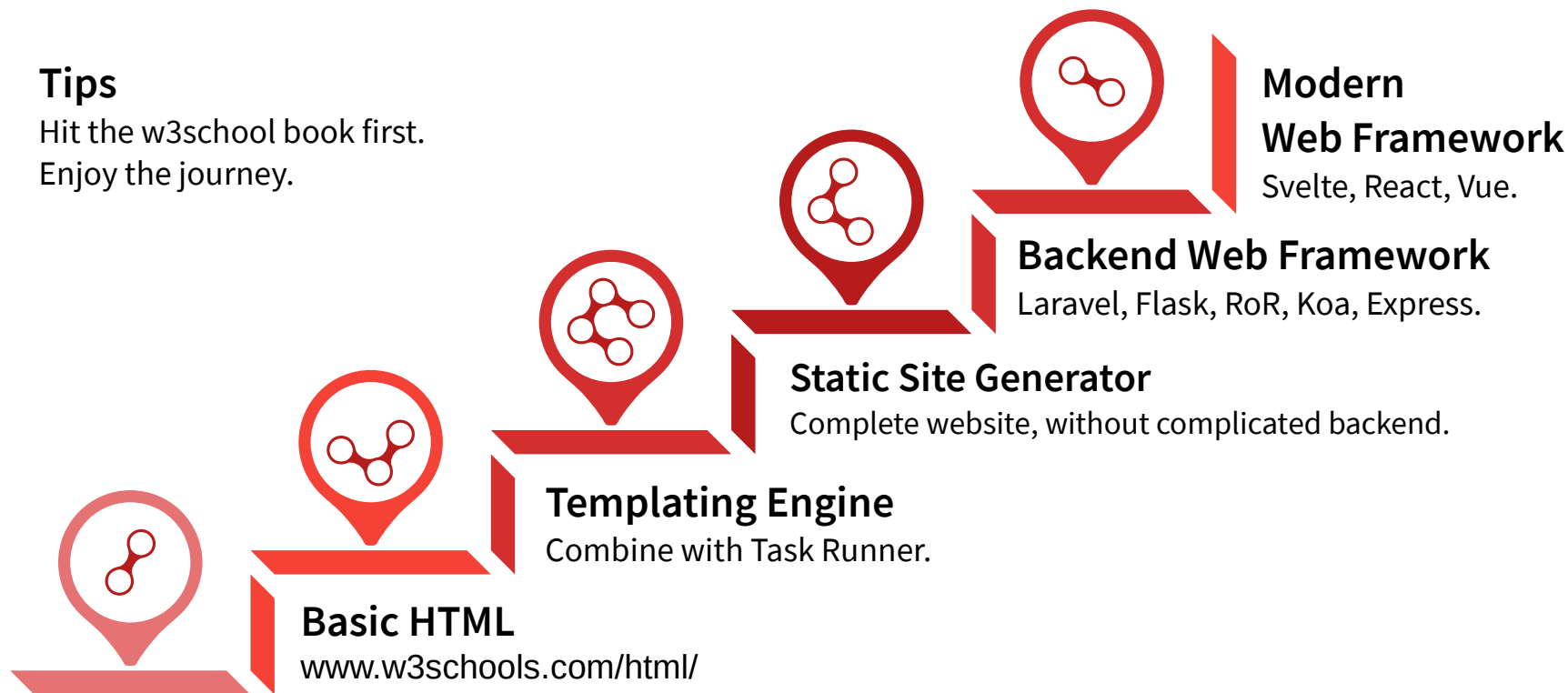
Content

Text and Image (or other media)

Learning **Plain HTML** Step by Step?

Tips

Hit the w3school book first.
Enjoy the journey.

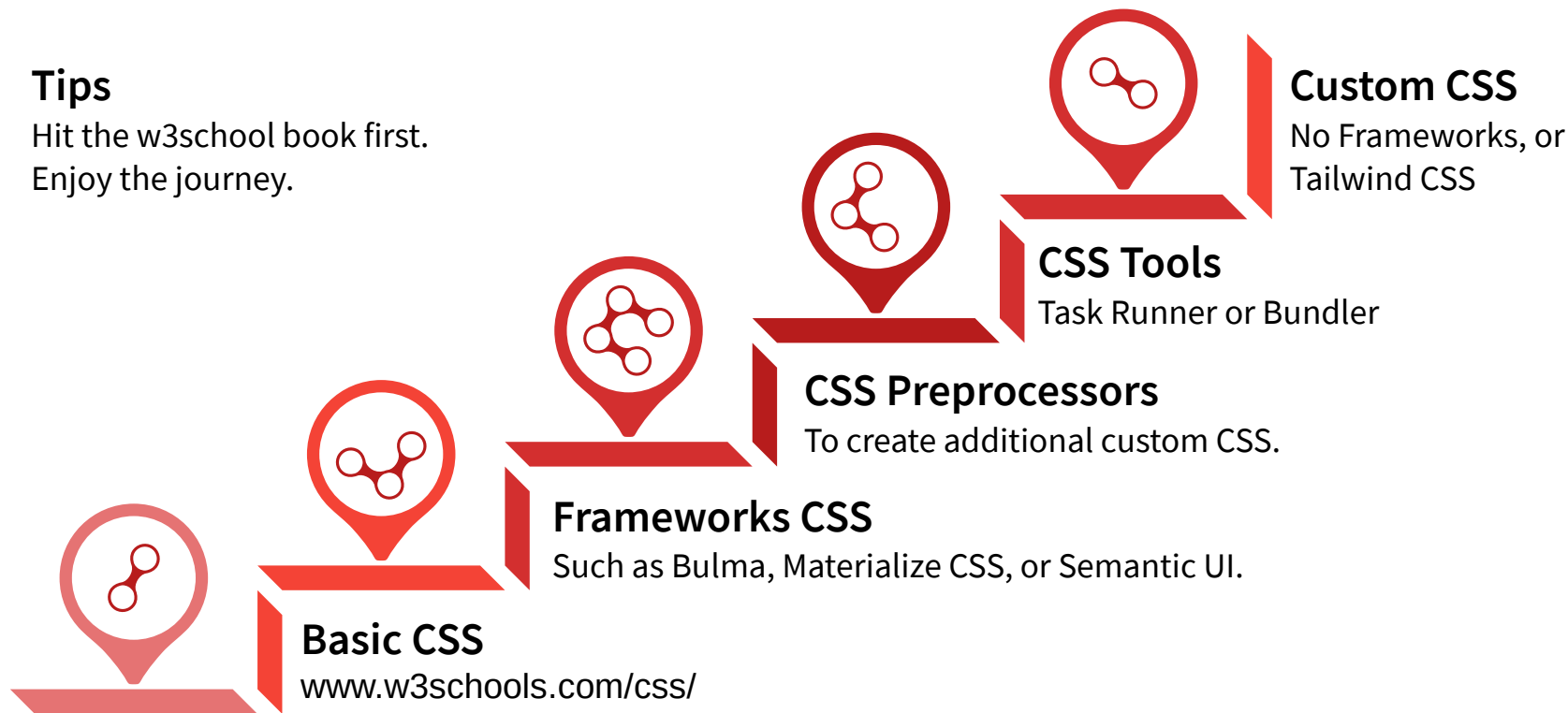


Learn how to google, make a screenshot, read documentation, and english.

Learning **CSS** Step by Step?

Tips

Hit the w3school book first.
Enjoy the journey.

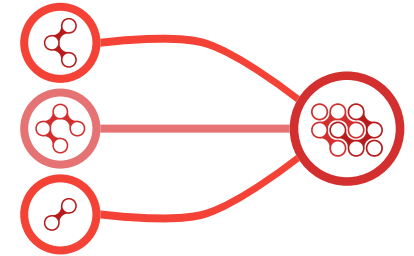


Learn how to google, make a screenshot, read documentation, and english.

Where to Put?

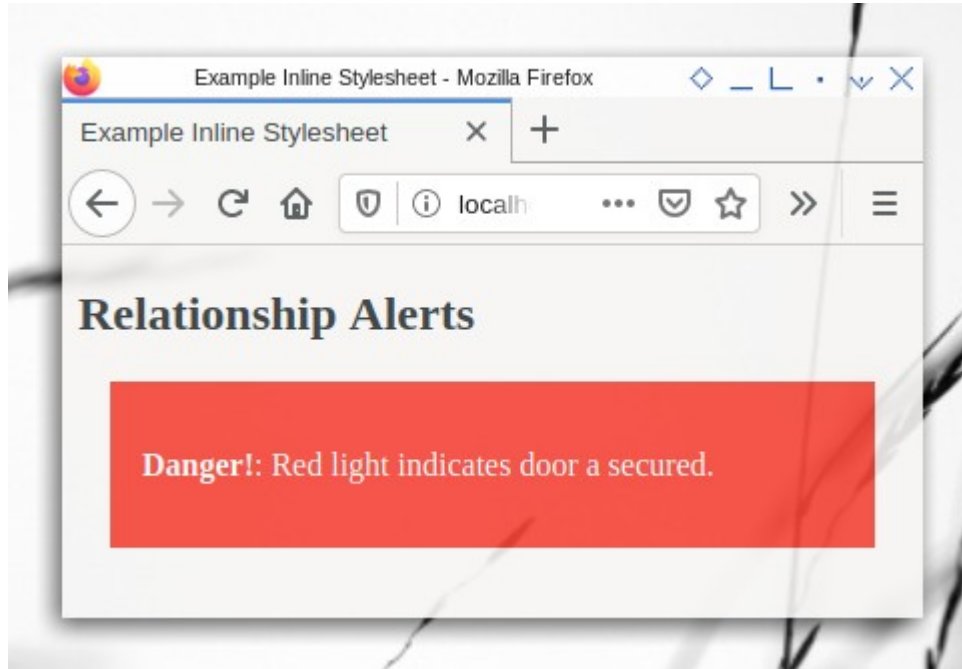
Placement

- Inline, or
- Internal, or
- External.



Yet another humble
oldschool case.

[[HTML - Stylesheet](#)]



CSS Frameworks

11 October, 2020

CSS: Frameworks and Tools
epsi-rns.gitlab.io

Issue in oldschool CSS Development

Reinvent the wheel

Different stylesheet for each project.
In need of code reuse.

Debugging

Most of the time, debugging.
The horror while testing in different browser.

No standard for teamwork

Waste of time, thinking of nice name for classes.

Can we have a generic one?

Why CSS Framework?

Save Man Hours

Deliver preview quickly.

Reduce Debugging and Test

Already well tested. Hence reduce bug.

Official Documentation

Ready to read.

Community Friendly

Known solution for recurring case.

Modularity

Built on top of CSS preprocessor: Sass, Less, or Stylus.
Tailwind on top of PostCSS even more modular.

Direct **Advantage** to Developer



Easy to layout

Grid system or such.



Ready to use

Reset, element, component, helper.



Predefined Properties

Color, and such.

Disadvantage

Learning Time

Require More Cups of Caffeine.

Bloated

Unless utilize modular feature,
exclude unneeded artefact.

Disadvantage for Beginner

Basic Design Provided

Every site made, will have similar looks.

Feels Like Witchery

This things work,
and nobody knows why.

When **not Using** CSS Framework?

You want to write your own

This is technically make sense, for custom design, to make something that suitable for your need.

Legacy Project

You step up into a team.

AMP

This require embedded stylesheet.

Step by Step Examples?

Bulma

[[Bulma + SASS](#)]

[[Bulma + Custom SASS](#)]
(Material Design)

Materialize CSS

[[Materialize CSS + SCSS](#)]

Semantic UI

[[Semantic UI + Custom LESS](#)]
(Material Design)

Bootstrap

[[Bootstrap + Custom SASS](#)]
(Open Color)

Tool: Preprocessor!

What is this CSS preprocessor rubbish?

In short: CSS with superpowers!

Just the tool we desire.

The CSS Frustration



CSS code get ugly from time to time

- Nomore clean code.
- Hard to read.
- Hard to maintain.



Always rewrite, for any changes

No variables in old browser.



Modularization affect HTTP perfomance.

Multiple files require more HTTP requests.

In short

Developing with CSS without tools is not scalable for large project.

Direct Advantage to Developer



Modify custom properties

Color, and such.



Ready to alter

Reset, element, component



You can claim **yourself as a coder**

This is a joke. Yes, there is a compilation process.

Why CSS Preprocessor?

Sass, Less or Stylus

Modularity

Let's get organized.
Cleaner code.



Stylesheet

Easier to Maintain Stylesheet

Basic Coding

import partials, variables, interpolation,
mixins, function arguments



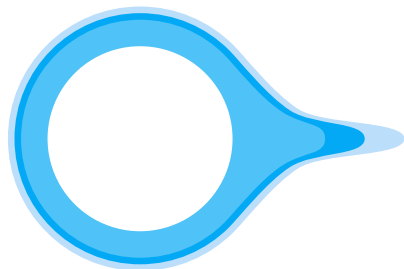
Made for Developer

Simplified declaration.
While compiled css is,
made for machine (web browser).

When **not Using** CSS Preprocessor?

You want to use unmodified version
of CSS Frameworks.

Such as using CDN.



Else

I'm still thinking...

Last 2019 State



Implementation

[SASS](#) has wide implementation in many language

Most common are: dart-sass, node-sass, and deprecated ruby-sass

Supported in both Gulp and Grunt.
And bundler: webpack, rollup, and parcel.



LESS is less used



Tailwind CSS

PostCSS with [Tailwind CSS](#) is a rising star.

[PostCSS](#) itself is more than just Preprocessor.



Else

I do not know about this one: [Stylus](#)

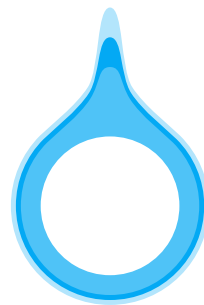
Which One Should I Choose?



**Choose what
works for you!**

**I don't know.
I'm just a blogger.**

Just like you. I'm still learning.



Preprocessor **Coding** Examples?

Custom SASS

[SASS: Loop - Spacing Class]



Custom PostCSS

[PostCSS: Loop - Spacing Class]



Custom LESS

[LESS: Loop - Spacing Class]
[LESS: Conditional - Color Class]



Respect!



Special **thank you** for
the smart **guys/girls**,
behind these
CSS preprocessor project.



CSS supporting Technologies

Supporting Technologies

Bundler

[[Webpack](#)]



[[Rollup](#)]



[[Parcel](#)]



Task Runner

[[Grunt](#)]



Gulp,
Brunch,
Broccoli.



Not Covered Yet?

More about CSS supporting Technologies



PostCSS

Along with their plugins.



Custom CSS

(Without Framework)

Along with repository example.



Tailwind

Along with repository example.

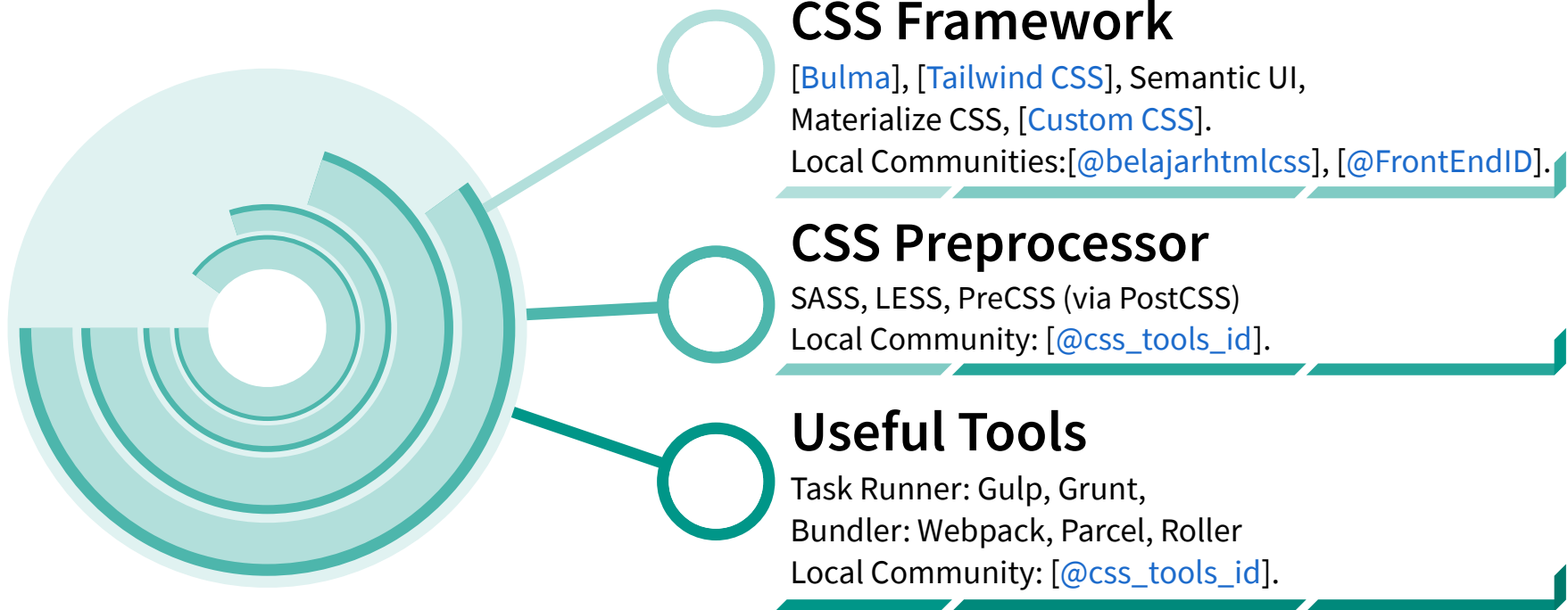
Notes:

This presentation will likely to change.
Depend on the growth of my knowledge.

Summary

Summary

Along with Local Communities





What is Next?

Leverage to SSG!

[[Introduction to SSG for beginner](#)]

Learn to make a fully mature website
without the burden of complicated backend.

Questions?

Don't be shy!

The End

Thank You for Your Time.