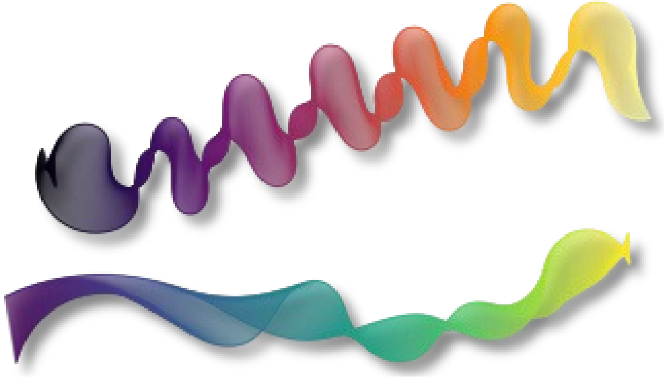
CSS: Frameworks and Tools

Introduction to CSS for beginner

for Personal Educational Purpose.



Jakarta, Indonesia

11 October, 2020

About Epsi

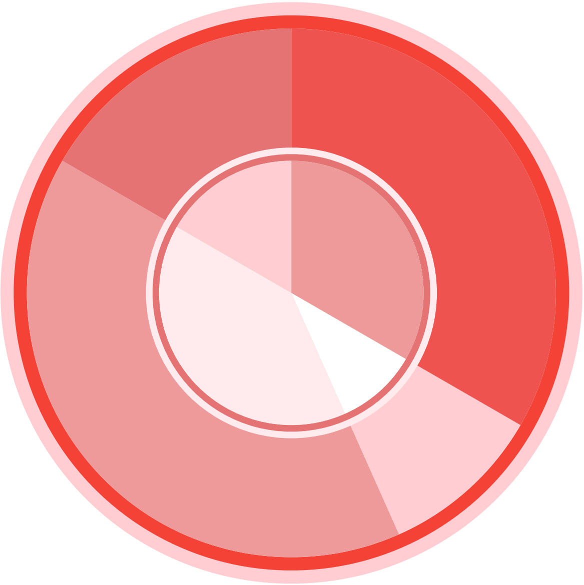
Yet, another underachiever.
But hey, I have my own blog.

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 hierarcy fashioned

Presentation

Stylesheet: Rule, Layout, Box Model

rule = selector + declaration

box model, position

layout: flex, grid, float+calc, table

Behaviour

Javascript

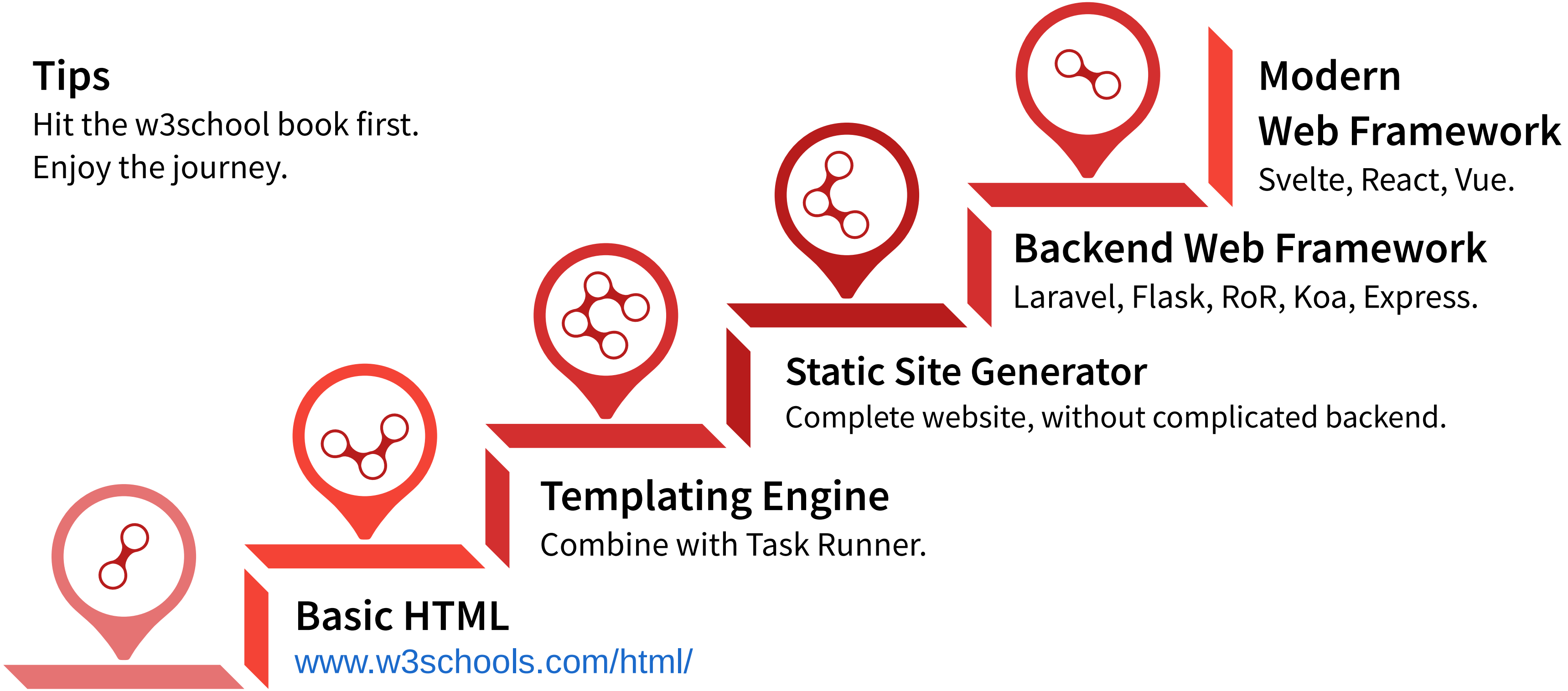
Content

Text and Image (or other media)

Learning Plain HTML Step by Step?

Tips

Hit the w3school book first.
Enjoy the journey.



Modern Web Framework
Svelte, React, Vue.

Backend Web Framework
Laravel, Flask, RoR, Koa, Express.

Static Site Generator
Complete website, without complicated backend.

Templating Engine
Combine with Task Runner.

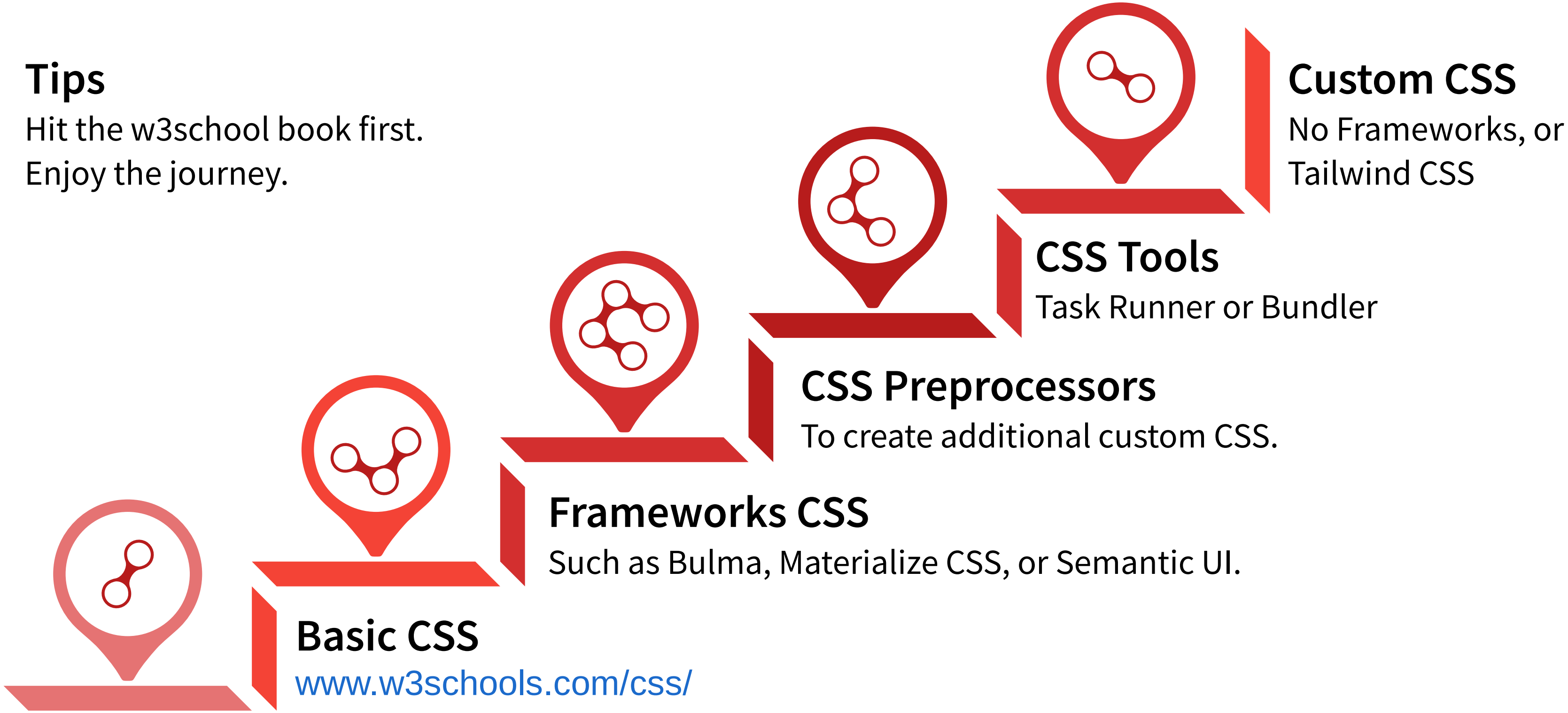
Basic HTML
www.w3schools.com/html/

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.



Basic CSS
www.w3schools.com/css/

Frameworks CSS
Such as Bulma, Materialize CSS, or Semantic UI.

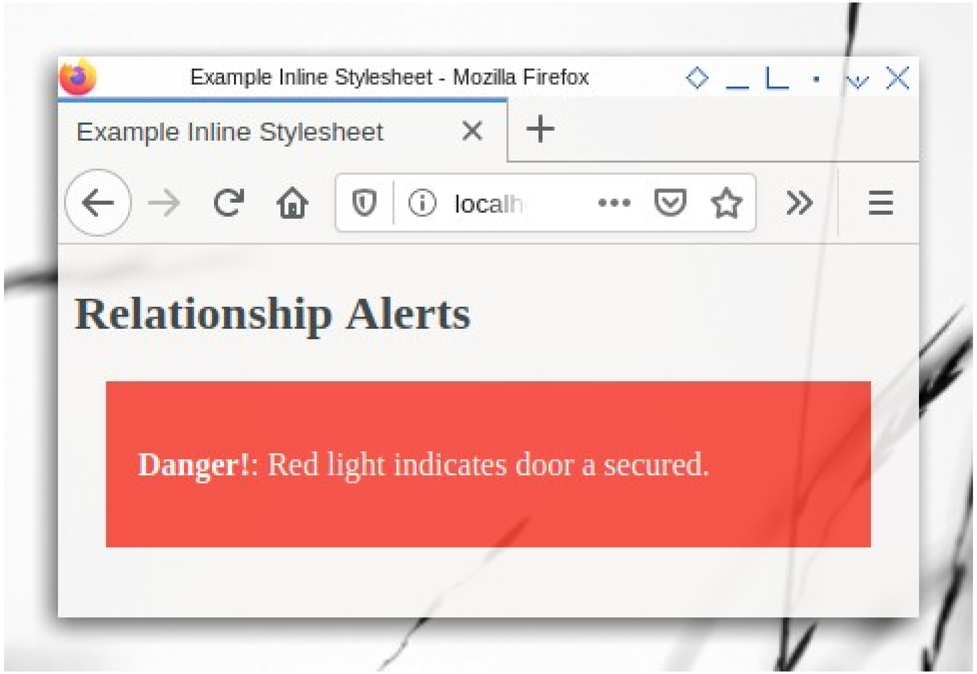
CSS Preprocessors
To create additional custom CSS.

CSS Tools
Task Runner or Bundler

Custom CSS
No Frameworks, or
Tailwind CSS

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

Where to Put?


Example Inline Stylesheet - Mozilla Firefox
Example Inline Stylesheet
localh

Relationship Alerts

Danger!: Red light indicates door a secured.

Placement

- Inline, or
- Internal, or
- External.

Yet another humble oldschool case.
[HTML - Stylesheet]

CSS Frameworks

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.

Feels Like Witchery

This things work,
and nobody knows why.

Disadvantage for Beginner

Basic Design Provided

Every site made, will have similar looks.

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 unmodifed 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.

Tailwind CSS

PostCSS with Tailwind CSS is a rising star.

PostCSS itself is more than just Preprocessor.

LESS is less used

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 LESS

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

Custom PostCSS

[PostCSS: Loop - Spacing Class]

Respect!

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

CSS supporting Technologies

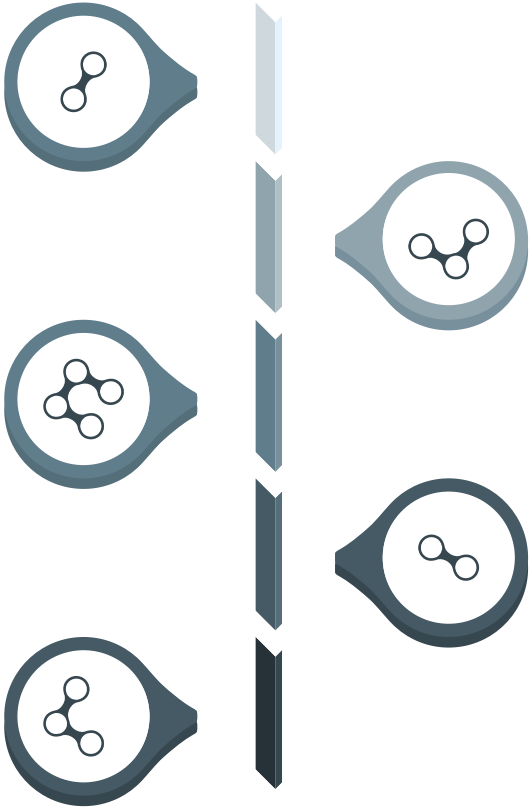
Supporting Technologies

Bundler

[Webpack]

[Rollup]

[Parcel]



Task Runner

[Grunt]

Gulp,
Brunch,
Brocolli.

Not Covered Yet?

More about CSS supporting Technologies

1st

PostCSS

Along with their plugins.

2nd

Custom CSS

(Without Framework)

Along with repository example.

3rd

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

CSS Framework

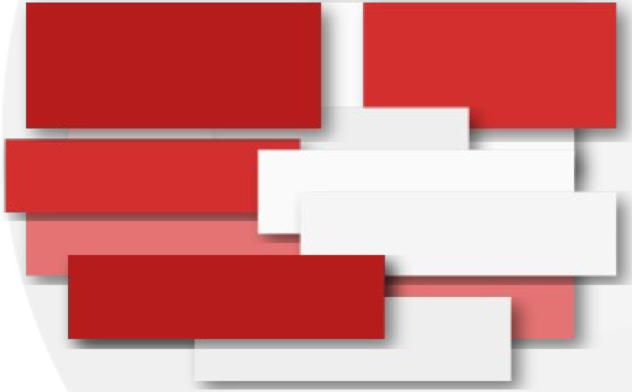
[Bulma], [Tailwind CSS], Semantic UI,
Materialize CSS, [Custom CSS].
Local Communities:[@belajarhtmlcss], [@FrontEndID].

CSS Preprocessor

SASS, LESS, PreCSS (via PostCSS)
Local Community: [@css_tools_id].

Useful Tools

Task Runner: Gulp, Grunt,
Bundler: Webpack, Parcel, Roller
Local Community: [@css_tools_id].



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.