

Junior Web Programmer

LPP Garden Hotel - Yogyakarta

Prayudi Utomo

1. Kolla Education - Jakarta
- Head of Education

2. Asesor Kompetensi - LSP TD
- Penguji Sertifikasi Programmer

3. Mitra Netra - Jakarta
- Coding Trainer for the Blind

4. Intl. Design School - Jakarta
- UX Trainer

5. Binar Academy - Jakarta
- UX Curriculum Consultant

6. Facebook Masterclass - Jakarta
- JavaScript Trainer

7. Grab Pte Ltd - Singapore
- UX Engineer

8. Day7 Pte Ltd - Singapore
- Front-End Developer

9. Vox Teneo Asia - Bandung
- SEO Specialist

10. Institut Manajemen Telkom
- Web Design & Dev Lecturer

Table of Contents

1. Kompetensi
2. HTML
3. CSS
4. Debugging
5. JavaScript
6. Responsive Web Design
7. Framework
8. Roadmap



Kompetensi



Kompetensi

1. Skema Unit
2. Unit

Unit Kompetensi Junior Web Programmer

No	Kode Unit	Judul Unit
1	J.620100.004.02	Menggunakan struktur data
2	J.620100.005.01	Mengimplementasikan user interface
3	J.620100.011.01	Melakukan instalasi software tools pemrograman
4	J.620100.016.01	Menulis kode dengan prinsip sesuai guidelines dan best practices
5	J.620100.017.02	Mengimplementasikan pemrograman terstruktur
6	J.620100.019.02	Menggunakan library atau komponen pre-existing
7	J.620100.023.02	Membuat dokumen kode program
8	J.620100.025.02	Melakukan debugging

Menggunakan Struktur Data

KODE UNIT : J.620100.004.01

JUDUL UNIT : Menggunakan Struktur Data

DESKRIPSI UNIT: Unit ini menentukan kompetensi, pengetahuan dan sikap kerja yang diperlukan dalam mempelajari dan membuat struktur data yang akan diterapkan pada pemrograman, tanpa tergantung bahasa pemrograman yang akan dipakai.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Mengidentifikasi konsep data dan struktur data	1.1 Konsep data dan struktur data diidentifikasi sesuai dengan konteks permasalahan. 1.2 Alternatif struktur data dibandingkan kelebihan dan kekurangannya untuk konteks permasalahan yang diselesaikan.
2. Menerapkan struktur data dan akses terhadap struktur data tersebut	2.1 Struktur data diimplementasikan sesuai dengan bahasa pemrograman yang akan dipergunakan. 2.2 Akses terhadap data dinyatakan dalam algoritma yang efisiensi sesuai bahasa pemrograman yang akan dipakai.



Mengimplementasikan User Interface

KODE UNIT : J.620100.005.01

JUDUL UNIT : Mengimplementasikan *User Interface*

DESKRIPSI UNIT: Unit ini menentukan kompetensi, pengetahuan dan Sikap kerja yang diperlukan dalam membuat rancangan antar muka program.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Mengidentifikasi rancangan <i>user interface</i>	<ul style="list-style-type: none">1.1 Rancangan <i>user interface</i> diidentifikasi sesuai kebutuhan.1.2 Komponen <i>user interface dialog</i> diidentifikasi sesuai konteks rancangan proses.1.3 Urutan dari akses komponen <i>user interface dialog</i> dijelaskan.1.4 Simulasi (<i>mock-up</i>) dari aplikasi yang akan dikembangkan dibuat.
2. Melakukan implementasi rancangan <i>user interface</i>	<ul style="list-style-type: none">2.1 Menu program sesuai dengan rancangan program diterapkan.2.2 Penempatan <i>user interface dialog</i> diatur secara sekuenzial.2.3 <i>Setting aktif-pasif</i> komponen <i>user interface dialog</i> disesuaikan dengan urutan alur proses.2.4 Bentuk <i>style</i> dari komponen <i>user interface</i> ditentukan.2.5 Penerapan simulasi dijadikan suatu proses yang sesungguhnya.



Melakukan Instalasi Software Tools Pemrograman

Kode Unit : J.620100.011.01

Judul Unit : Melakukan Instalasi *Software Tools Pemrograman*

Deskripsi Unit : Unit ini mengukur kemampuan *programmer* dalam melakukan instalasi lingkungan pemrograman yang akan digunakan dalam pekerjaan membuat program.

Elemen Kompetensi	Kriteria Unjuk Kerja
1. Memilih <i>tools</i> pemrograman yang sesuai dengan kebutuhan	1.1. <i>Platform</i> (lingkungan) yang akan digunakan untuk menjalankan <i>tools</i> pemrograman diidentifikasi sesuai dengan kebutuhan. 1.2. <i>Tools</i> bahasa pemrograman dipilih sesuai dengan kebutuhan dan lingkungan pengembangan.
2. Instalasi tool pemrograman	2.1. <i>Tools</i> pemrograman ter- <i>install</i> sesuai dengan prosedur. 2.2. <i>Tools</i> pemrograman bisa dijalankan di lingkungan pengembangan yang telah ditetapkan
3. Menerapkan hasil pemodelan kedalam eksekusi <i>script</i> sederhana	3.1. <i>Script</i> (<i>source code</i>) sederhana dibuat sesuai <i>tools</i> pemrograman yang di- <i>install</i> 3.2. <i>Script</i> dapat dijalankan dengan benar dan menghasilkan keluaran sesuai scenario yang diharapkan



Menulis Kode dengan Prinsip sesuai Best Practice

Kode Unit : J.620100.016.01

Judul Unit : Menulis Kode Dengan Prinsip Sesuai *Guidelines* dan *Best Practices*

Deskripsi Unit : Unit ini menentukan kompetensi, pengetahuan dan Sikap kerja yang diperlukan dalam menerapkan prinsip penulisan kode yang baik agar kode tersebut dapat dirawat (*Maintainability*).

Elemen Kompetensi	Kriteria Unjuk Kerja
3. Menerapkan <i>coding-guidelines</i> dan <i>best practices</i> dalam penulisan program (kode sumber)	<ul style="list-style-type: none">3.1. Kode sumber dituliskan mengikuti <i>coding-guidelines</i> dan <i>best practices</i>.3.2. Struktur program yang sesuai dengan konsep paradigmnya dibuat.3.3. Galat/error ditangani.
4. Menggunakan ukuran performansi dalam menuliskan kode sumber.	<ul style="list-style-type: none">4.1. Efisiensi penggunaan <i>resources</i> oleh kode dihitung.4.2. Kemudahan interaksi selalu diimplementasikan sesuai standar yang berlaku.

Mengimplementasikan Pemrograman Terstruktur

Kode Unit : J.620100.017.02

Judul Unit : Mengimplementasikan Pemrograman Terstruktur

Deskripsi Unit : Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan untuk membuat program terstruktur atau prosedural.

Elemen Kompetensi	Kriteria Unjuk Kerja
1. Menggunakan tipe data dan <i>control program</i>	1.1. Tipe data yang sesuai standar ditentukan. 1.2. <i>Syntax program</i> yang dikuasai digunakan sesuai standar. 1.3. Struktur kontrol program yang dikuasai digunakan sesuai standar

Menggunakan Library atau Komponen Pre-Existing

KODE UNIT : J.620100.019.002

JUDUL UNIT : Menggunakan *Library* atau *Komponen Pre-Existing*

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang diperlukan untuk menggunakan komponen-komponen *reuse* (yang dapat dipergunakan secara berulang) untuk mendukung pengembangan aplikasi yang efisien.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Melakukan pemilihan unit-unit <i>reuse</i> yang potensial	1.1 <i>Class</i> unit-unit <i>reuse</i> (dari aplikasi lain) yang sesuai dapat diidentifikasi. 1.2 Keuntungan efisiensi dari pemanfaatan komponen <i>reuse</i> dapat dihitung. 1.3 Lisensi, Hak cipta dan hak paten tidak dilanggar dalam pemanfaatan komponen <i>reuse</i> tersebut.
2. Melakukan integrasi <i>library</i> atau komponen <i>pre-existing</i> dengan <i>source code</i> yang ada	2.1 Ketergantungan antar unit diidentifikasi. 2.2 Penggunaan komponen yang sudah <i>obsolete</i> dihindari. 2.3 Program yang dihubungkan dengan <i>library</i> diterapkan.
3. Melakukan pembaharuan <i>library</i> atau komponen <i>pre-existing</i> yang digunakan	3.1 Cara-cara pembaharuan <i>library</i> atau komponen <i>pre-existing</i> diidentifikasi. 3.2 Pembaharuan <i>library</i> atau komponen <i>pre-existing</i> berhasil dilakukan.



Membuat Dokumen Kode Program

Kode Unit : J.620100.023.02

Judul Unit : Membuat Dokumen Kode Program

Deskripsi Unit : Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang diperlukan untuk membuat dokumentasi dari kode program yang telah ditulis secara *hardcopy* termasuk identifikasi penjelasan dari dokumen tersebut.

Elemen Kompetensi	Kriteria Unjuk Kerja
1. Melakukan identifikasi kode program	1.1. Modul program diidentifikasi 1.2. Parameter yang dipergunakan diidentifikasi 1.3. Algoritma dijelaskan cara kerjanya 1.4. Komentar setiap baris kode termasuk data, eksepsi, fungsi, prosedur dan <i>class</i> (bila ada) diberikan
2. Membuat dokumentasi modul program	2.1. Dokumentasi modul dibuat sesuai dengan identitas untuk memudahkan pelacakan 2.2. Identifikasi dokumentasi diterapkan 2.3. Kegunaan modul dijelaskan 2.4. Dokumen direvisi sesuai perubahan kode program
3. Membuat dokumentasi fungsi, prosedur atau method program	3.1. Dokumentasi fungsi, prosedur atau metod dibuat 3.2. Kemungkinan eksepsi dijelaskan 3.3. Dokumen direvisi sesuai perubahan kode program
4. Men-generate dokumentasi	4.1. <i>Tools</i> untuk generate dokumentasi diidentifikasi 4.2. Generate dokumentasi dilakukan



Membuat Debugging

Kode Unit : J.620100.025.02

Judul Unit : Melakukan *Debugging*

Deskripsi Unit : Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan dalam memeriksa kode program dari kesalahan (*bug*).

Elemen Kompetensi	Kriteria Unjuk Kerja
1. Mempersiapkan kode program	1.1. Kode program sesuai spesifikasi disiapkan. 1.2. <i>Debugging tools</i> untuk melihat proses suatu modul dipersiapkan
2. Melakukan <i>debugging</i>	2.1. Kode program dikompilasi sesuai bahasa pemrograman yang digunakan. 2.2. Kriteria lulus <i>build</i> dianalisis. 2.3. Kriteria eksekusi aplikasi dianalisis. 2.4. Kode kesalahan dicatat.
3. Memperbaiki program	3.1. Perbaikan terhadap kesalahan kompilasi maupun <i>build</i> dirumuskan. 3.2. Perbaikan dilakukan.



A blurred background image of a workspace. It features a laptop displaying a landscape photo, some papers, and a small cup holding several pens or pencils.

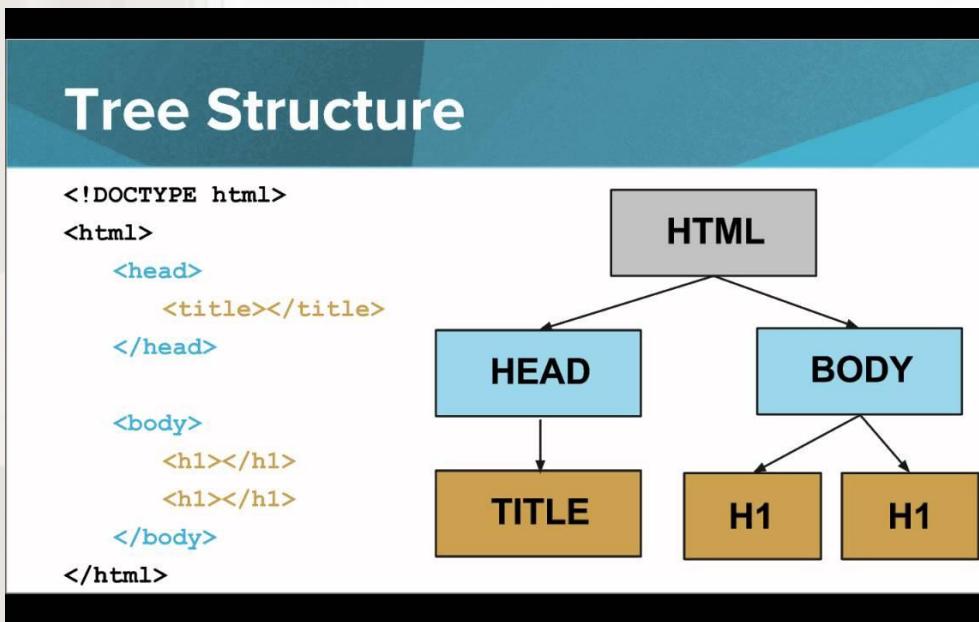
HTML

HTML or Hypertext Markup Language is the standard markup language for creating Web pages. HTML describes the structure of a Web page.

HTML

1. Structure
2. Semantic
3. Typography
4. Hyperlink
5. Multimedia
6. Form
7. Table

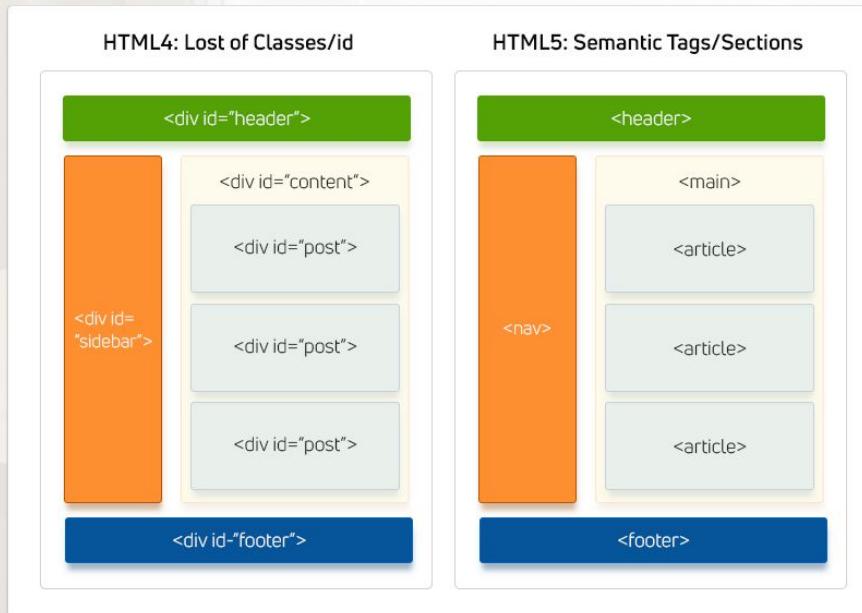
Structure



HTML Anatomy:

1. DOCTYPE
2. <html>
3. <head>
4. <body>
5. Hierarchical Tree = Document Object Model (DOM)

Semantic



Semantic HTML4:

1. `<header>`
2. `<main>`
3. `<nav>`
4. `<article>`
5. `<section>`
6. `<footer>`
7. Etc

Sumber: <https://jasonchen050319.blogspot.com/2020/09/68-semantic-tags-for-seo-how-to-use.html>

Typography

5.38em

The quick browr

3.84em

The quick brown fox jur

2.74em

The quick brown fox jumped ove

1.96em

The quick brown fox jumped over the lazy do

1.4em

The quick brown fox jumped over the lazy dog.

1em

The quick brown fox jumped over the lazy dog.

Typography:

1. Heading
2. Paragraph
3. List
4. Inline Style
5. Etc

Hyperlink



Hyperlink:

1. Tag <a>
2. Attribute: href
3. Attribute: title
4. Etc

Form

First name:

Last name:

E-mail:

Male
 Female

Login To Your Account

Sumber: <https://www.sliderrevolution.com/resources/css-forms/>

Table

No.	Full Name	Position	Salary
1	Bill Gates	Founder Microsoft	\$1000
2	Steve Jobs	Founder Apple	\$1200
3	Larry Page	Founder Google	\$1100
4	Mark Zuckerberg	Founder Facebook	\$1300

Table:

1. Tag <table>
2. Tag <thead>
3. Tag <tfoot>
4. Tag <tr>
5. Tag <th>
6. Tag <td>
7. Etc



A blurred background image of a workspace featuring a laptop, a tablet displaying a horse image, and various desk accessories like a cup with pens and a small plant.

CSS

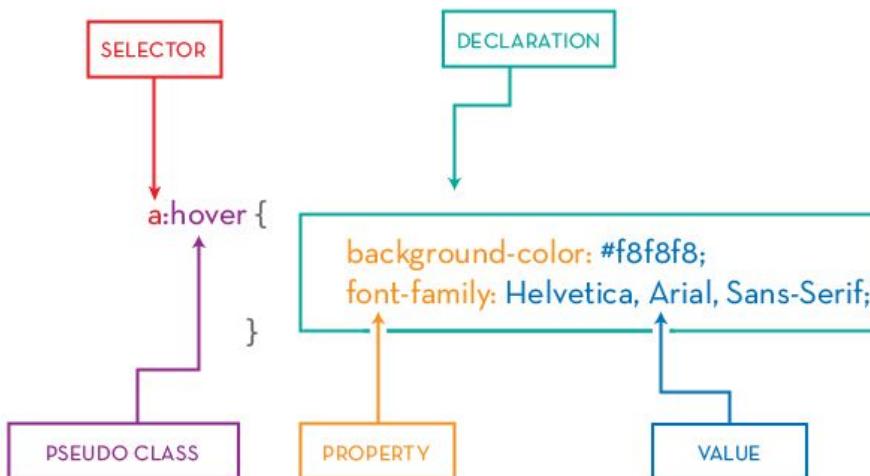
CSS or Cascading Style Sheet is the standard markup language for creating Web pages. HTML describes the structure of a Web page.

CSS

1. Structure
2. Color
3. Typography
4. Box Model
5. Layout

Structure

CLASS SELECTOR WITH PSEUDO CLASS CSS RULE



CSS Anatomy:

1. Selector
2. Pseudo Class
3. Declaration
4. Property
5. Value

Sumber: <https://ironion.com/blog/2015/06/12/anatomy-of-a-css-rule/>



Color

Adobe Color

CREATE EXPLORE TRENDS LIBRARIES

Color Wheel Extract Theme Extract Gradient Accessibility Tools New

Apply Color Harmony Rule

Analogous
 Monochromatic
 Triad
 Complementary
 Split Complementary
 Double Split Complementary
 Square
 Compound
 Shades
 Custom

A B C D E

#A7A110 #34A3F0 #50E6C0 #E0E437 #A20023

Language: English User Forums Community Guidelines Copyright © 2021 Adobe. All rights reserved. Privacy Terms of Use Cookie preferences Do Not Sell My Personal Information

Save

Color wheel (or image in Extract Theme tab) can be used to generate color palette which can be saved into Creative Cloud, after signing in.

You can then use your saved color themes, in Adobe products (Photoshop, Illustrator, Fresco etc.), via Adobe Color theme panel or CC Libraries.

Adobe

Color:

1. Analogous
2. Monochromatic
3. Complementary
4. Etc

Sumber: <https://color.adobe.com/create/color-wheel>

Typography

Learn to Code

HTML & CSS

Lesson 1
[Building Your First Web Page](#)

Lesson 2
[Getting to Know HTML](#)

Lesson 3
[Getting to Know CSS](#)

Lesson 4
[Opening the Box Model](#)

Lesson 5
[Positioning Content](#)

Lesson 6
[Working with Typography](#)

Lesson 7
[Setting Backgrounds & Gradients](#)

Lesson 6

Working with Typography

The field of [web typography](#) has grown substantially over time. There are a couple of different reasons for its rise in popularity; one widely acknowledged reason is the development of a system for embedding our own web fonts on a website.

In the past we were limited to a small number of typefaces that we could use on a website. These typefaces were the most commonly installed fonts on computers, so they were the most likely to render properly on-screen. If a font wasn't installed on a computer, it wouldn't render on the website either. Now, with the ability to embed fonts, we have a much larger palette of typefaces to choose from, including those that we add to a website.

While the ability to embed fonts gives us access to countless new typefaces, it's also important for us to know the basic principles of typography. In this lesson we're going to take a look at some of these basic principles and how to apply them to our web pages using HTML and CSS.

Typeface vs. Font

The terms "typeface" and "font" are often interchanged, causing confusion. Here is a breakdown of exactly what each term means.

In this Lesson

6

HTML

- » [Citations & Quote](#)

CSS

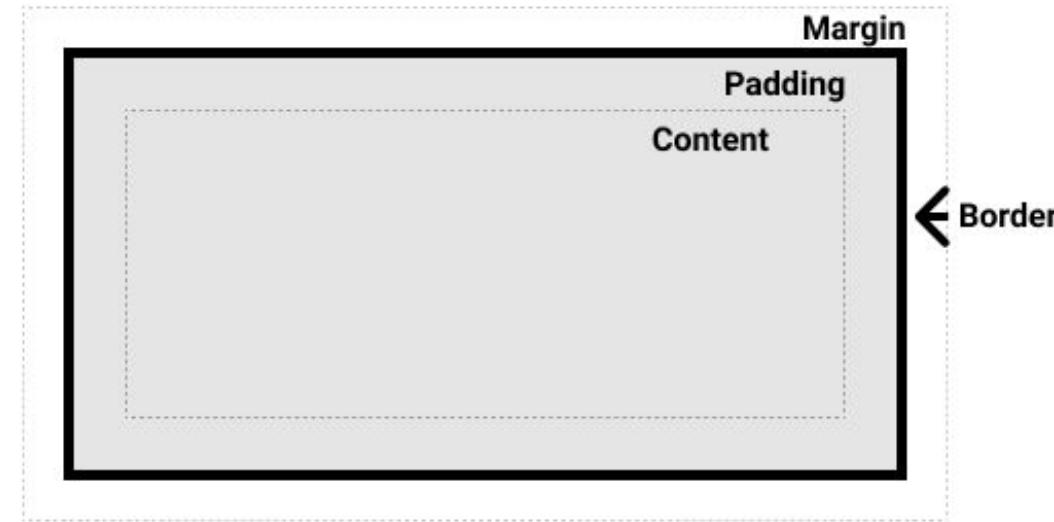
- » [Text Color](#)
- » [Font Properties](#)
- » [Text Properties](#)
- » [Web-Safe Fonts](#)
- » [Embedding Web Fonts](#)

SHARE

Typography:

1. Heading
2. Paragraph
3. List
4. Inline Style
5. Etc

Box Model



Box Model Elements:

1. Content
2. Padding
3. Border
4. Margin

Flexbox

FLEXBOX FROGGY

Level 1 of 24 ▶

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lilypad on the right by using the `justify-content` property, which aligns items horizontally and accepts the following values:

- `flex-start`: Items align to the left side of the container.
- `flex-end`: Items align to the right side of the container.
- `center`: Items align at the center of the container.
- `space-between`: Items display with equal spacing between them.
- `space-around`: Items display with equal spacing around them.

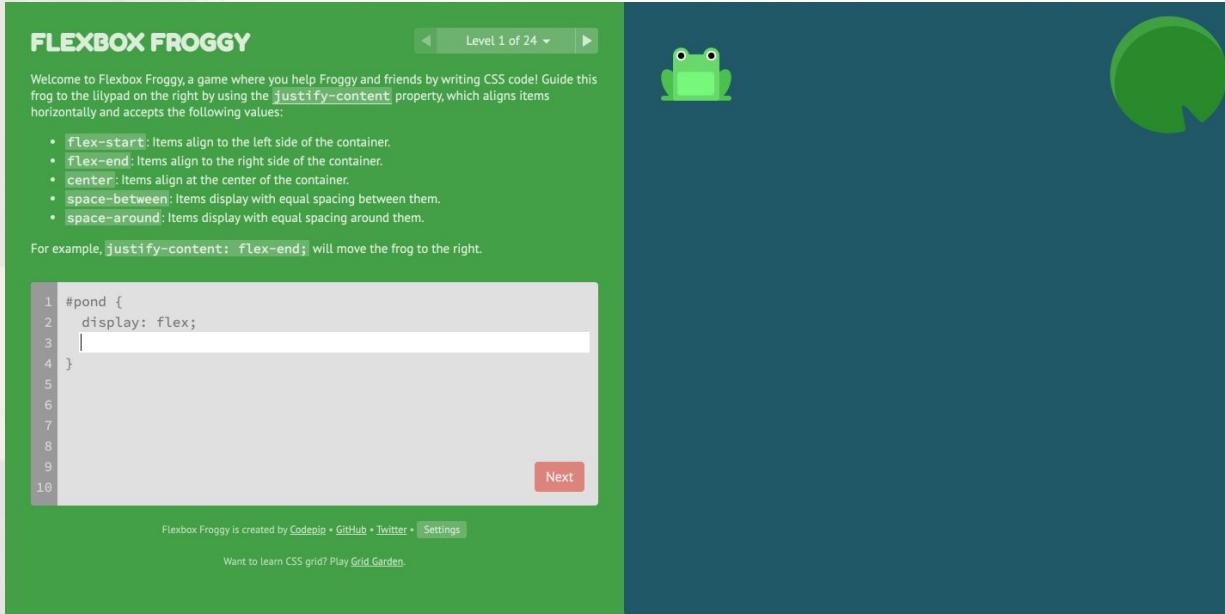
For example, `justify-content: flex-end;` will move the frog to the right.

```
1 #pond {  
2   display: flex;  
3   |  
4 }  
5  
6  
7  
8  
9  
10
```

[Next](#)

Flexbox Froggy is created by [CodePen](#) • [GitHub](#) • [Twitter](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).



Sumber: <https://flexboxfroggy.com/>

CSS - Layout

Grid

GRID GARDEN

Level 1 of 28 ▶

Welcome to Grid Garden, where you write CSS code to grow your carrot garden! Water only the areas that have carrots by using the `grid-column-start` property.

For example, `grid-column-start: 3;` will water the area starting at the 3rd vertical grid line, which is another way of saying the 3rd vertical border from the left in the grid.

```
1 #garden {  
2   display: grid;  
3   grid-template-columns: 20% 20% 20% 20% 20%;  
4   grid-template-rows: 20% 20% 20% 20% 20%;  
5 }  
6  
7 #water {  
8     
9 }  
10  
11  
12  
13  
14
```

[Next](#)

Grid Garden is created by [Codepig](#) • [GitHub](#) • [Twitter](#) • [English](#)

Want to learn CSS flexbox? Play [Flexbox Froggy](#).



Sumber: <https://cssgridgarden.com/>



Debugging



Table of Contents

1. Comment
2. Inspect Element



A blurred background image of a desk setup. On the desk is a white laptop displaying a nature photograph, a white mouse, a smartphone, and a small cup holding pens. The desk is light-colored and appears to be made of wood. In the background, there are windows with blinds and a lamp.

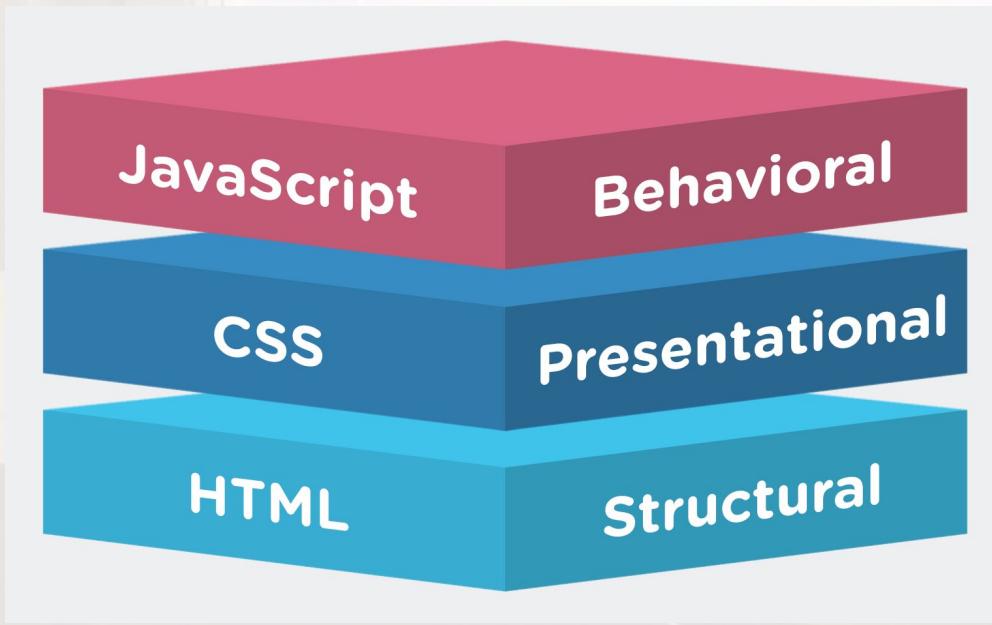
JavaScript



Table of Contents

1. Ecosystem
2. Fundamental
3. DOM
4. Standard

Romance of Three Kingdoms



HTML:

- Structural
- Content

CSS:

- Presentational
- Stylize

JavaScript:

- Behavioral
- Interactivity

JavaScript Fundamental

JavaScript Fundamentals

2.1 Hello, world!	2.7 Type Conversions	2.13 Loops: while and for
2.2 Code structure	2.8 Basic operators, maths	2.14 The "switch" statement
2.3 The modern mode, "use strict"	2.9 Comparisons	2.15 Functions
2.4 Variables	2.10 Conditional branching: if, ?:	2.16 Function expressions
2.5 Data types	2.11 Logical operators	2.17 Arrow functions, the basics
2.6 Interaction: alert, prompt, confirm	2.12 Nullish coalescing operator ??	2.18 JavaScript specials

Code quality

3.1 Debugging in the browser	3.3 Comments	3.5 Automated testing with Mocha
3.2 Coding Style	3.4 Ninja code	3.6 Polyfills and transpilers

Objects: the basics

4.1 Objects	4.4 Object methods, "this"	4.7 Symbol type
4.2 Object references and copying	4.5 Constructor, operator "new"	4.8 Object to primitive conversion
4.3 Garbage collection	4.6 Optional chaining ?.	

Data types

5.1 Methods of primitives	5.5 Array methods	5.9 Object.keys, values, entries
5.2 Numbers	5.6 Iterables	5.10 Destructuring assignment
5.3 Strings	5.7 Map and Set	5.11 Date and time
5.4 Arrays	5.8 WeakMap and WeakSet	5.12 JSON methods, toJSON

Fundamental:

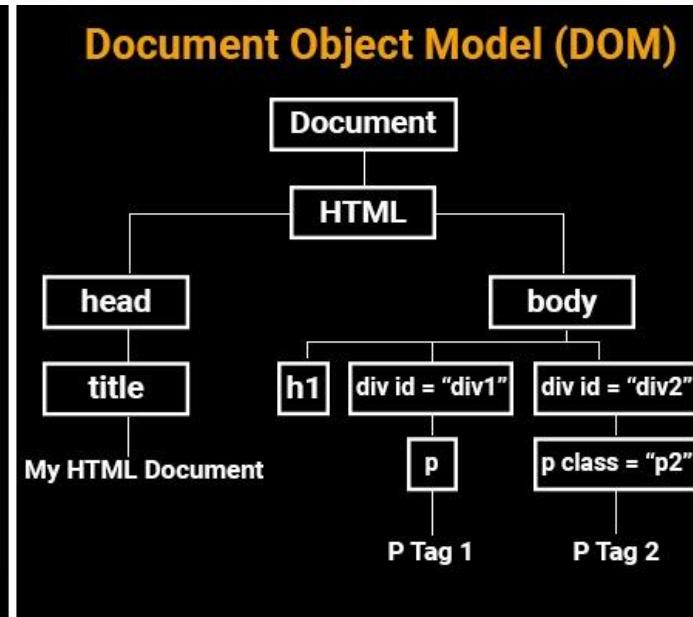
1. Syntax & Grammar
2. Variable & Constant
3. Data Type
4. Operator
5. Conditional
6. Loop
7. Functions
8. Etc

Document Object Model

HTML Document

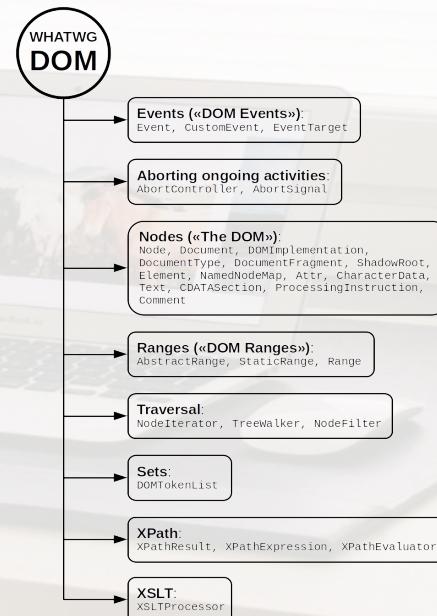
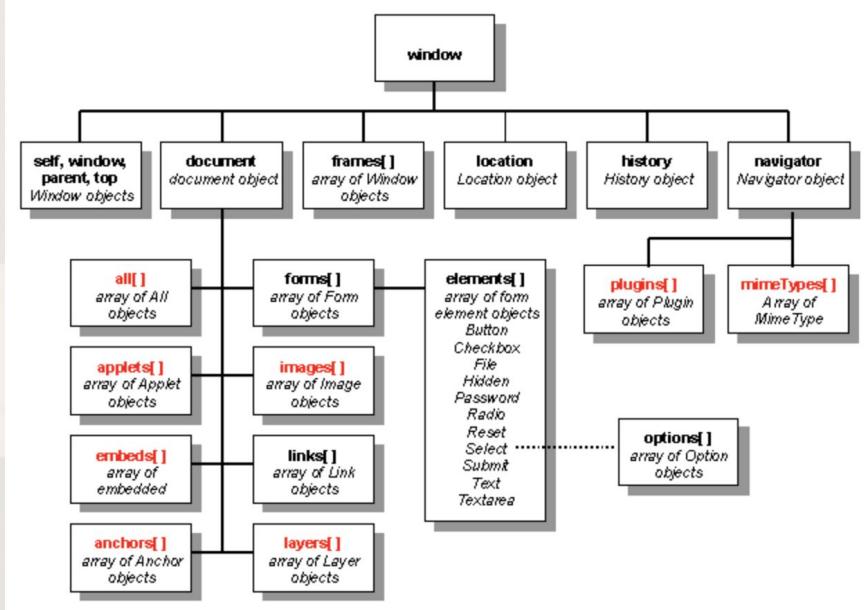
```
<html>
  <head>
    <title>My HTML Document</title>
  </head>

  <body>
    <h1>Heading</h1>
    <div id="div1">
      <p>P Tag 1</p>
    </div>
    <div id="div2">
      <p class="p2">P Tag 2</p>
    </div>
  </body>
</html>
```



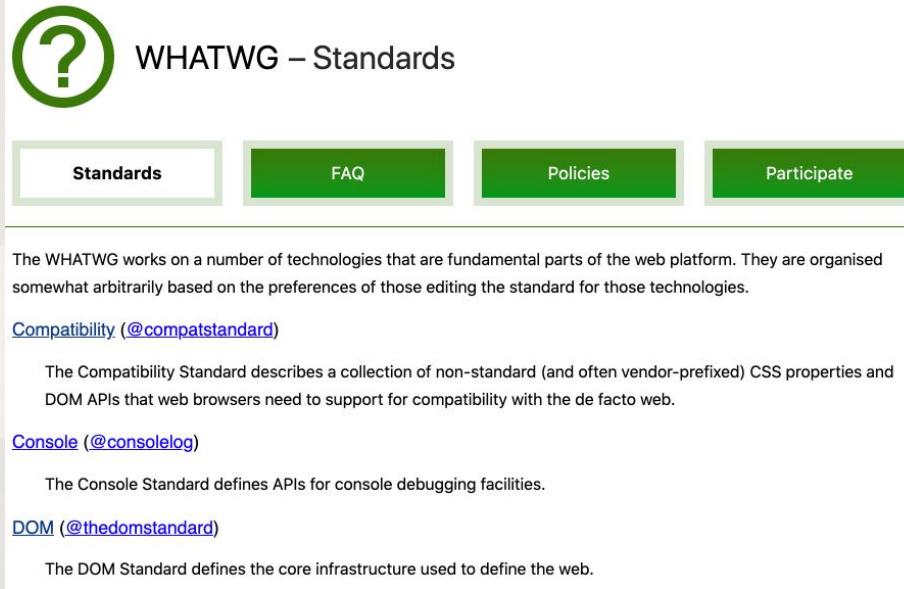
Document Object Model (DOM) defines the logical structure of documents and the way a document is accessed and manipulated.

Document Object Model



Document Object Model (DOM)
defines the logical structure of documents and the way a document is accessed and manipulated.

Standard



The screenshot shows the homepage of the WHATWG Standards website. It features a large green question mark icon and the text "WHATWG – Standards". Below this are four buttons: "Standards" (highlighted in light green), "FAQ", "Policies", and "Participate". A section titled "Compatibility (@compatstandard)" discusses the standard for non-standard CSS properties and DOM APIs. Another section, "Console (@consolelog)", covers console debugging APIs. The "DOM (@thedomstandard)" section is mentioned but not expanded. A footer at the bottom links to the source URL: <https://spec.whatwg.org/>.

WHATWG – Standards

Standards FAQ Policies Participate

The WHATWG works on a number of technologies that are fundamental parts of the web platform. They are organised somewhat arbitrarily based on the preferences of those editing the standard for those technologies.

[Compatibility \(@compatstandard\)](#)

The Compatibility Standard describes a collection of non-standard (and often vendor-prefixed) CSS properties and DOM APIs that web browsers need to support for compatibility with the de facto web.

[Console \(@consolelog\)](#)

The Console Standard defines APIs for console debugging facilities.

[DOM \(@thedomstandard\)](#)

The DOM Standard defines the core infrastructure used to define the web.

WHATWG Standards:

- Compatibility
- Console
- DOM
- Encoding
- Fetch
- MIME
- XMLHttpRequest
- Etc

Sumber: <https://spec.whatwg.org/>

Playground

JavaScript DOM Lab

This page shows some different ways to use the DOM to change the page. It does not cover *every possibility* but tries to cover enough to make a good start.

Try expanding some of the tasks on the right and filling in some of the values and applying the changes to see how it affects the Demo Output (preview) and the HTML code.

Demo Output Reset Demo HTML

Demo Document

Absolutely anything could happen here

Demo HTML

```
<html>
  <head>
    <style>
      .red {
        color: red;
        background-color: #20202A;
      }
    </style>
  </head>
  <body id="demo">
    <h1 id="demo1">Demo Document</h1>
    <p id="demo2">Absolutely anything could happen here</p>
  </body>
</html>
```

used to dynamically grow a list of options in a select tag for a dynamic drop-down menu.

setAttribute() to Hide Expand

Hides an element by setting the 'hidden' attribute to *true*

getAttribute() to Get Info Collapse

```
var element = document.body.querySelector('#demo1');
var value = element.getAttribute("id");
console.log(value)
```

_Element

Attribute Name

Gets any attribute on an element. You can store info in a "data-[X]" attribute. This will alert if the requested attribute isn't present.

Apply

demo1

removeAttribute() to remove an Attribute Collapse

```
document.body.removeAttribute("")
```

_Element

Remove an attribute from an element. Perfect for when you

Sumber: <https://wazzamo.github.io/dom-demo/demo/>



A blurred background image of a laptop and a smartphone resting on a desk, suggesting a workspace or technology theme.

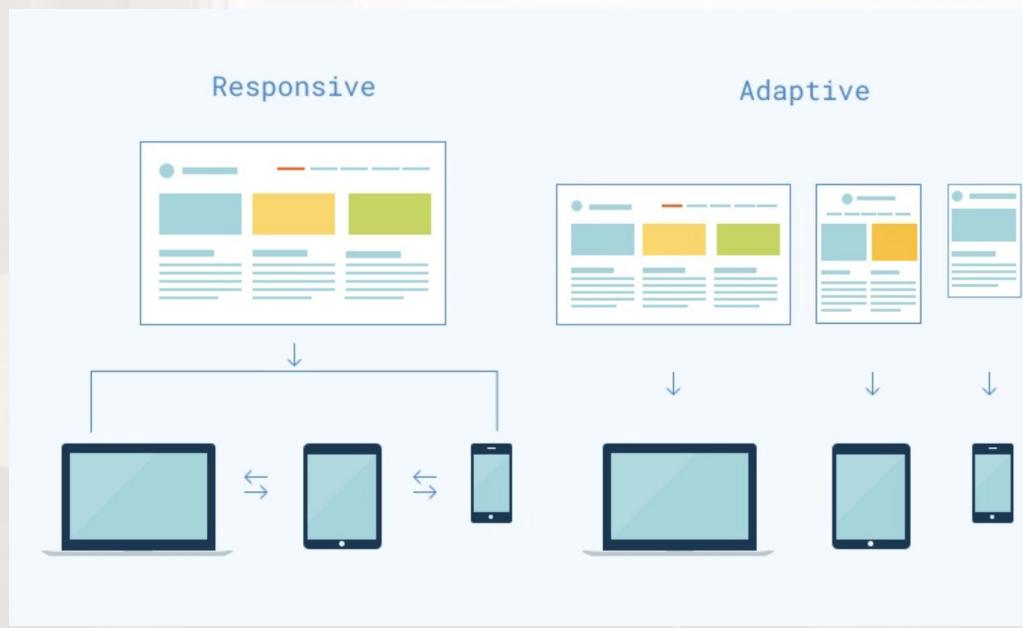
Responsive Web Design

Framework is HTML/CSS/JavaScript frameworks and packages having pre-written, standardized code in folders and files

Responsive Web Design

1. Approaches
2. Viewport
3. Breakpoint
4. Media Query
5. Responsive Image

Approaches



Responsive Web Design:

- Write once
- Fit All

Adaptive Web Design:

- Write once
- Fit once

Approaches



Responsive Web Design



Mobile First Web Design



Degradation Gracefully:
Larger screen first

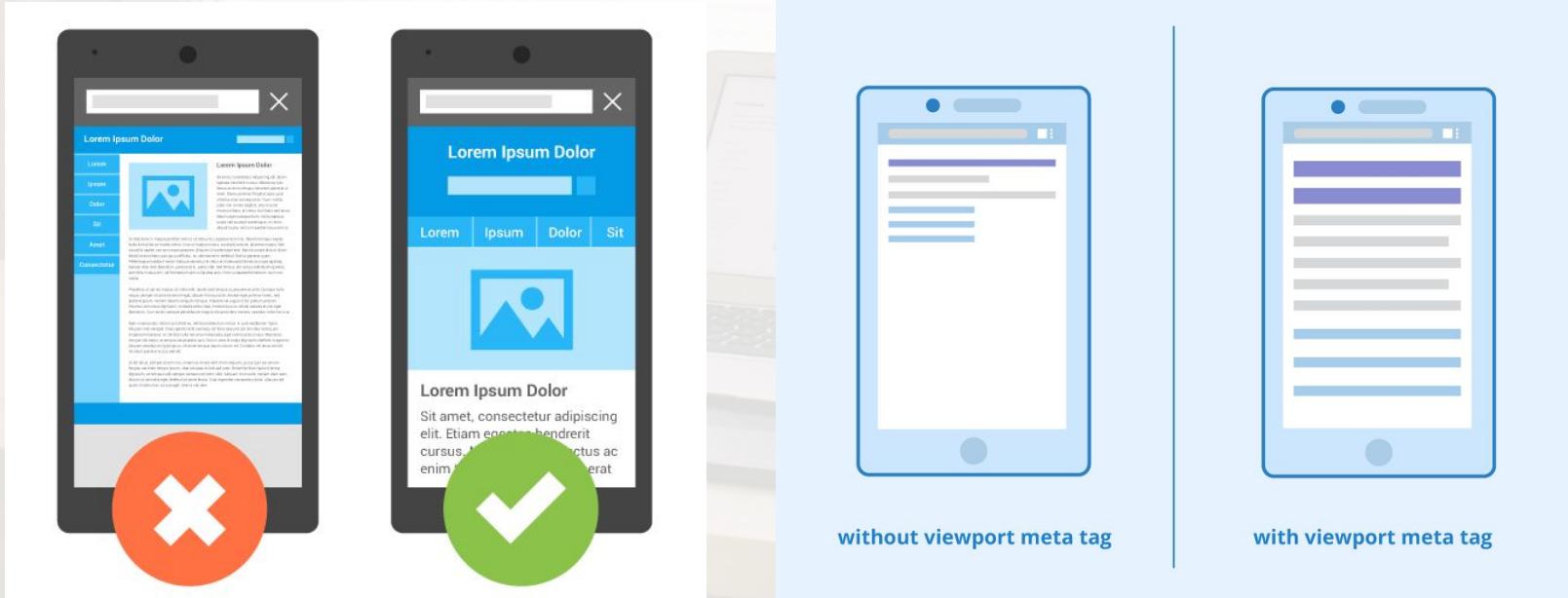
Progressive Enhancement:
Mobile screen first

Viewport

```
● ● ●  
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    ...  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
  </head>  
  ...
```

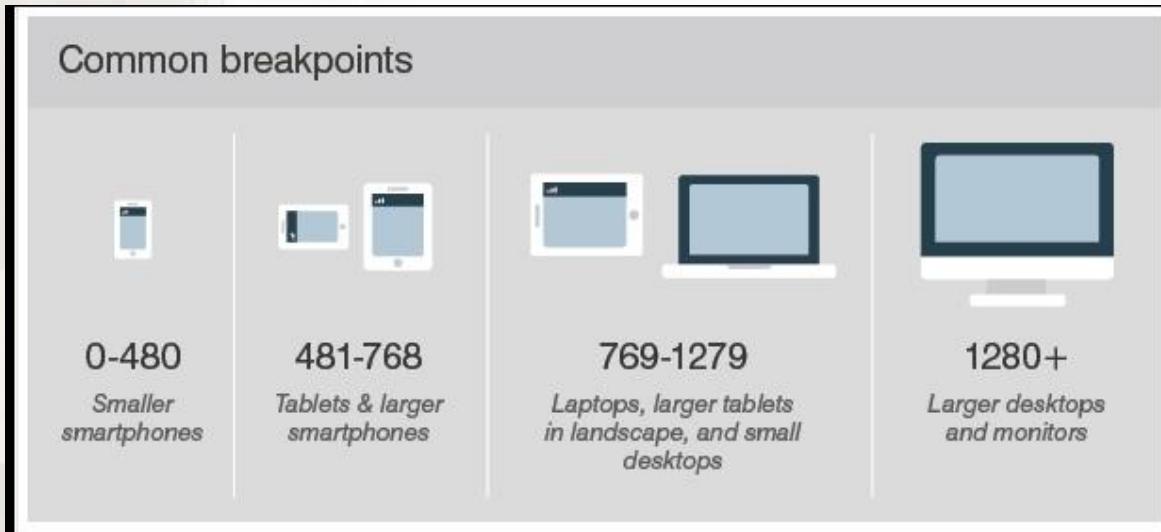
<meta name="viewport" content="width=device-width, initial-scale=1">

Viewport



Sumber: <https://www.seobility.net/en/wiki/Viewport>

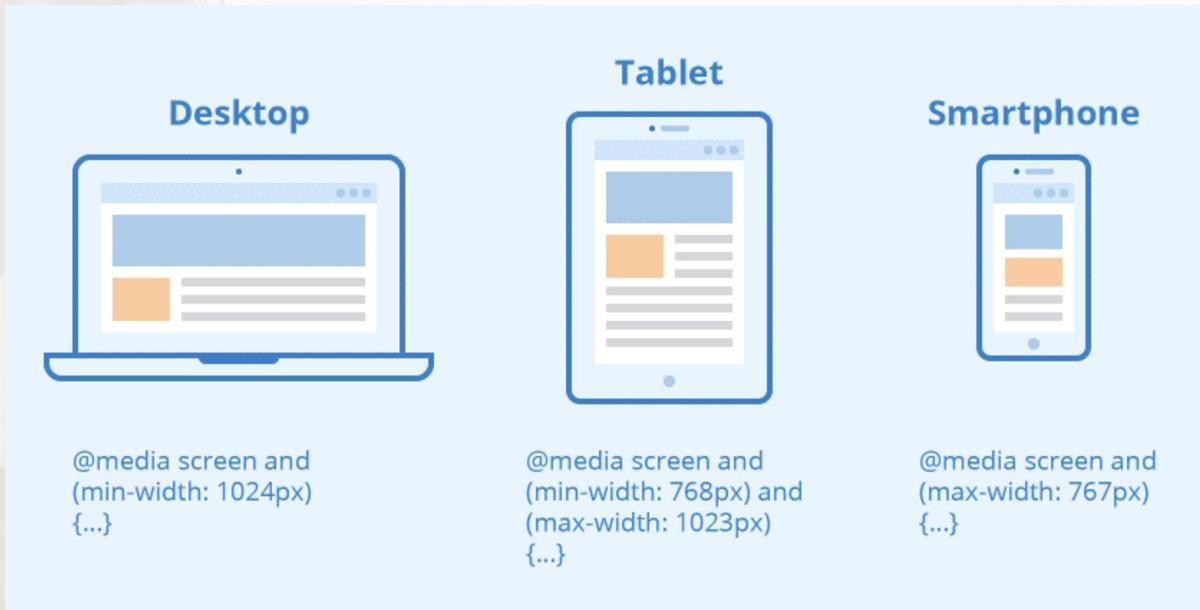
Breakpoint



Breakpoint:

1. Smartphone
<480
2. Tablet
481 - 768
3. Laptop/Desktop
769 - 1279
4. Large Screen
1280+

Breakpoint



Breakpoint:

1. **Smartphone**
<480
2. **Tablet**
768 - 1023
3. **Laptop/Desktop**
1024+

Media Query



```
@media screen and (min-width: 40em) {  
    // your code here  
}  
  
@media screen and (min-width: 70em) {  
    // your code here  
}
```

```
@media screen and (min-width: 40em) {  
    // your code here  
}  
}
```

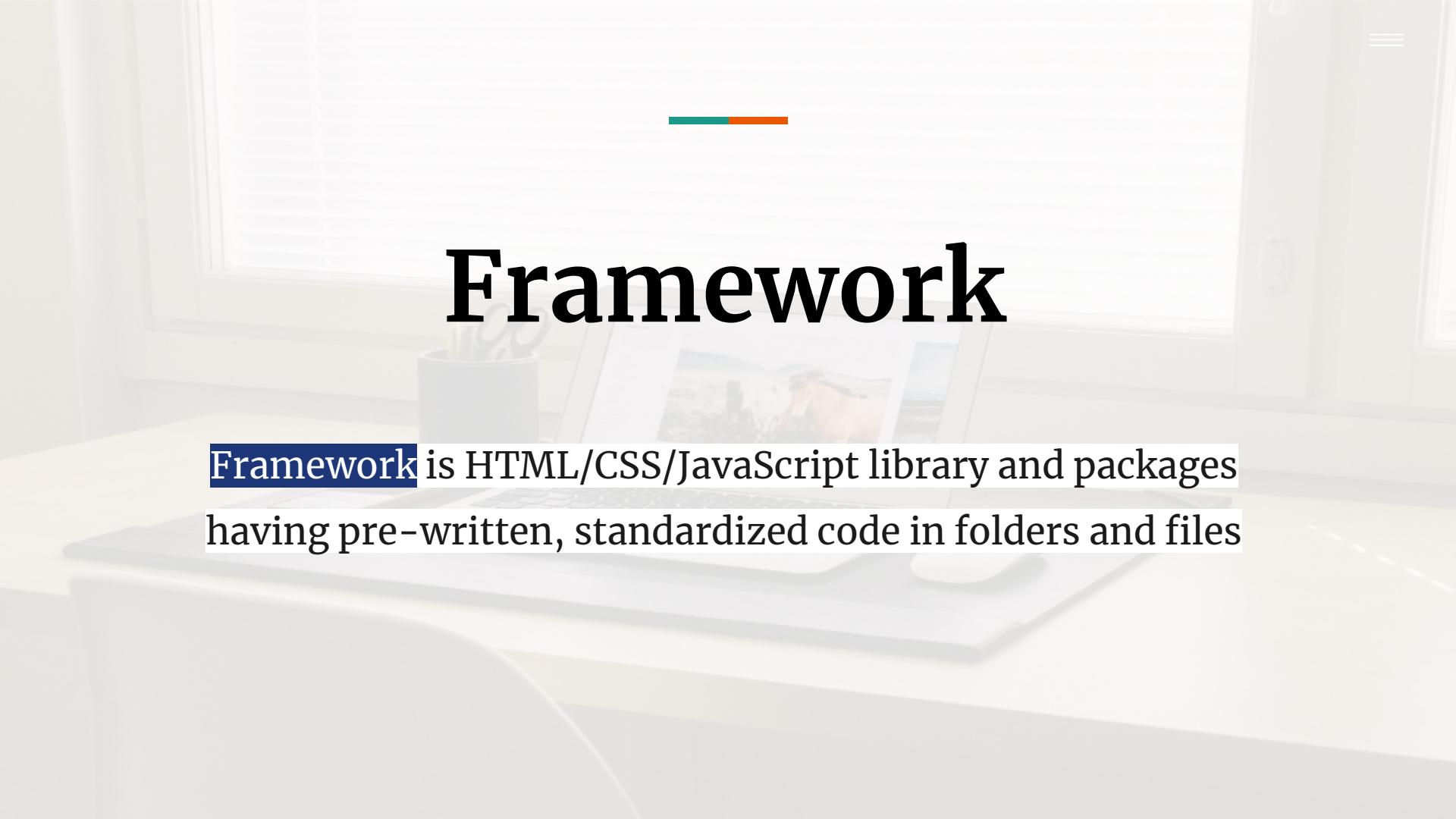
```
@media screen and (min-width: 70em) {  
    // your code here  
}
```

Responsive Image



```
img {  
  max-width: 100%;  
  display: block;  
}
```

```
img {  
  max-width: 100%;  
  display: block;  
}
```



A blurred background image of a workspace featuring a laptop displaying a landscape photo, several books, and a white mug containing a dark liquid.

Framework

Framework is HTML/CSS/JavaScript library and packages having pre-written, standardized code in folders and files

Framework

1. HTML/CSS Template
2. HTML/CSS Framework
3. Front-end Framework

HTML/CSS Template

The screenshot shows the homepage of nicepage.com. At the top, there's a navigation bar with links for Download, 200+ Features, 8500+ Templates, Website Builders, Premium, Forums, Blog, and Help. On the right side of the nav bar are links for EN, Sign In, and Register. The main banner features a large text "8500+ Best HTML5 Templates" and a search bar with the placeholder "Enter Keywords". Below the banner, a subtext reads "Free Download HTML5 Templates. HTML5 Website Templates. Basic and Simple Responsive HTML5 Template". To the right of the banner is a large graphic of the HTML5 logo. The bottom section displays "New HTML5 Template Thematic Categories" with seven categories: Art & Design, Technology, Food & Restaurant, Architecture & Build..., Business & Law, Fashion & Beauty, Education, and Cars & Transport. Each category has a thumbnail image and a brief description.

Sumber: <https://nicepage.com/html5-template>

HTML/CSS Template:

1. html5up.net
2. nicepage.com
3. etc

HTML/CSS Framework

PARAMETERS	Bootstrap	Foundation	Materialize CSS	Semantic UI	Bulma
Release Date	Aug 19,2011	Sep, 2011	4 Nov,2014	26 Sep,2013	24 Jan, 2016
License	MIT	MIT	MIT	MIT	MIT
Current version	4.4.1	6.6.3	1.0.0	2.4.0	0.8.2
Size	Minified CSS:- 61.7KB	Minified CSS:- 139.2KB	Minified CSS:- 175.2KB	-	Minified CSS:- 209.5KB
Language & Pre-Processor					
Grid	12 column flexible grid system	XY12-column grid Floatedsystem	Standered 12 column fluid responsive grid system	-	12 column flexible grid system
Core concept	RWD and mobile first	RWD and mobile first,Semantic	RWD and based on google material design	RWD and mobile first	RWD and mobile first

HTML/CSS Framework

2020:

1. **Bootstrap**
2. **Foundation**
3. **Materialize**
4. **Semantic UI**
5. **Bulma**
6. **Tailwind**

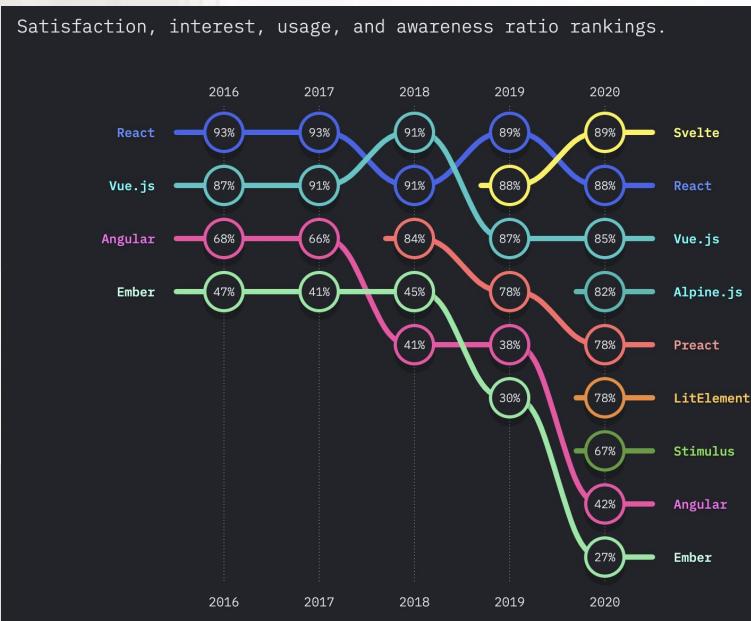
Ecosystem



Frontend Framework
popular:

1. Angular
2. React
3. Vue.js
4. Etc

Front-end Framework



Front-end Framework 2020:

1. Svelte
2. React
3. Vue.js
4. Alpine.js
5. Preact
6. LitElement
7. Stimulus
8. Angular
9. Ember

Sumber: <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>

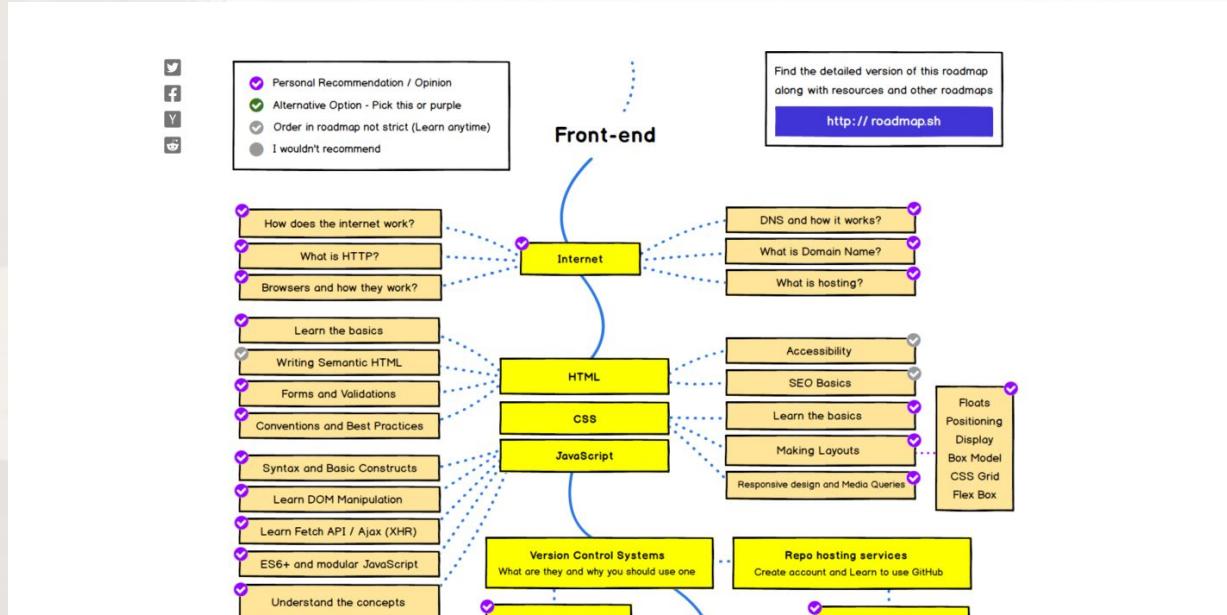


A blurred background image of a desk setup. On the desk is a white laptop displaying a nature photograph, a white mouse on a matching mousepad, a small black cup holding pens and pencils, and a white smartphone. The desk is light-colored wood, and there are windows with blinds in the background.

Roadmap



Roadmap Front-end Developer



Sumber: <https://roadmap.sh/frontend>

Front-end Developer Roadmap:

1. Practitioner
2. Academic
3. Hybrid

Thank you

