Mastering the Web:

A Hands-On Web Development Workshop

Building Blocks of Web Development

Contents of the Workshop Part 1

Part 1 Development Environment Setup

Part 2 HTML, CSS and JavaScript Essentials

Part 3 Frontend vs Backend

Tools We Will Need

- Google Chrome Browser (Chrome Developer Tools)
- Visual Studio Code
- Git / Github Account

Development Environment Setup

VSCODE EXTENSIONS

:HTML Boilerplate

:Auto Close Tag

:Live Server

:Prettier

:CSS Peek

Github Repo Materials

micah3252625/
Workshop-Web-Dev

2 Contributors | 0 Issues | 0 Stars | 0 Forks

micah3252625/Workshop-Web-Dev

Contribute to micah3252625/Workshop-Web-Dev development by creating an account on GitHub.

GitHub

Speaker's Notes:

Hello, I'm Mhezel Mohammad

- Computer Engineering Graduate - ADZU 2018
- Studies MIT in ADZU 2023
- 4 years Experience as Full Stack Developer

System Application Developer

https://github.com/mhezel

instagram.com/mhezyqt

https://g.dev/mhezel-mohammad

facebook.com/mhezelkhan

WEB DEVELOPMENT

How the Web Works?

Understand the Basics

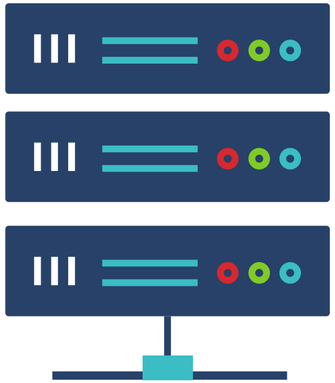
Client

A user visiting a website via a browser

Request

Response

A document with the website content

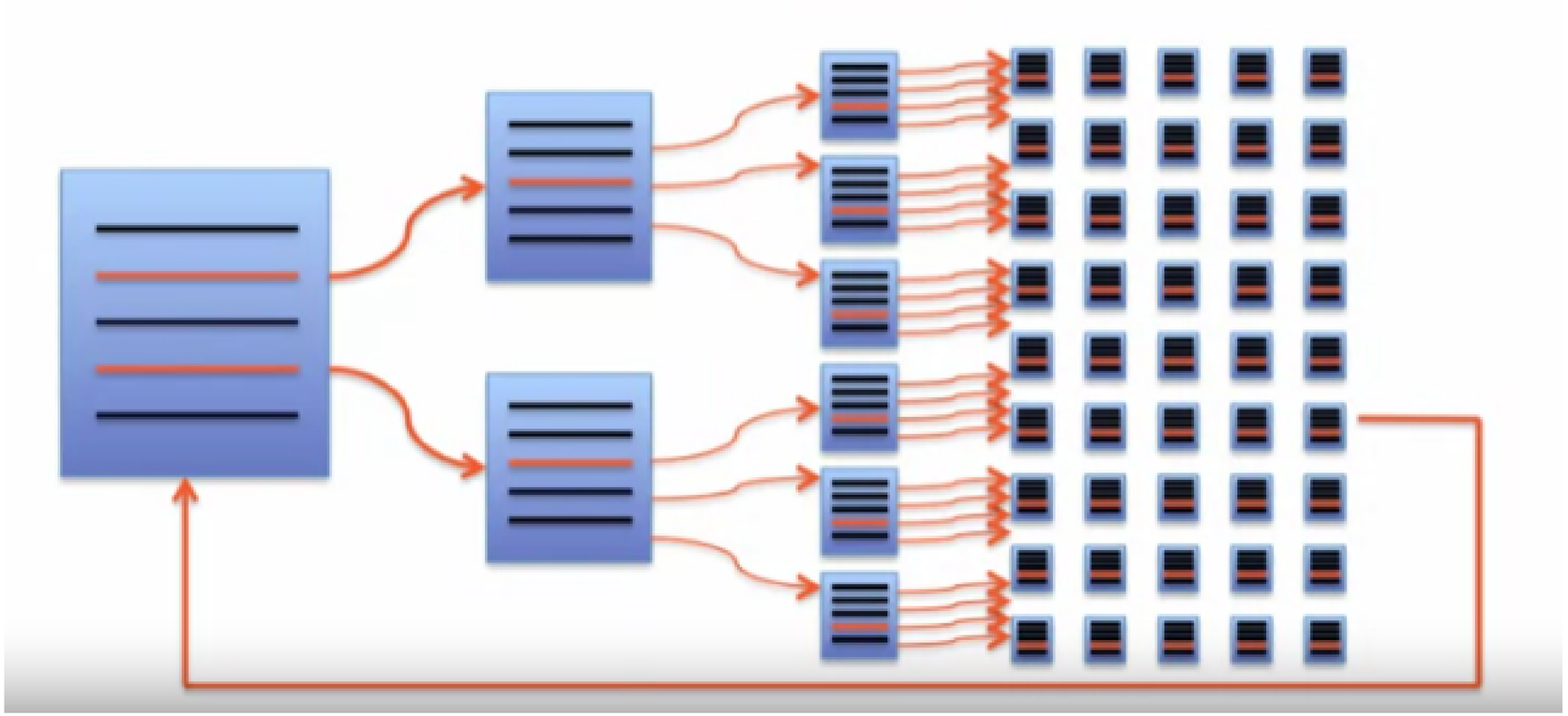


Server

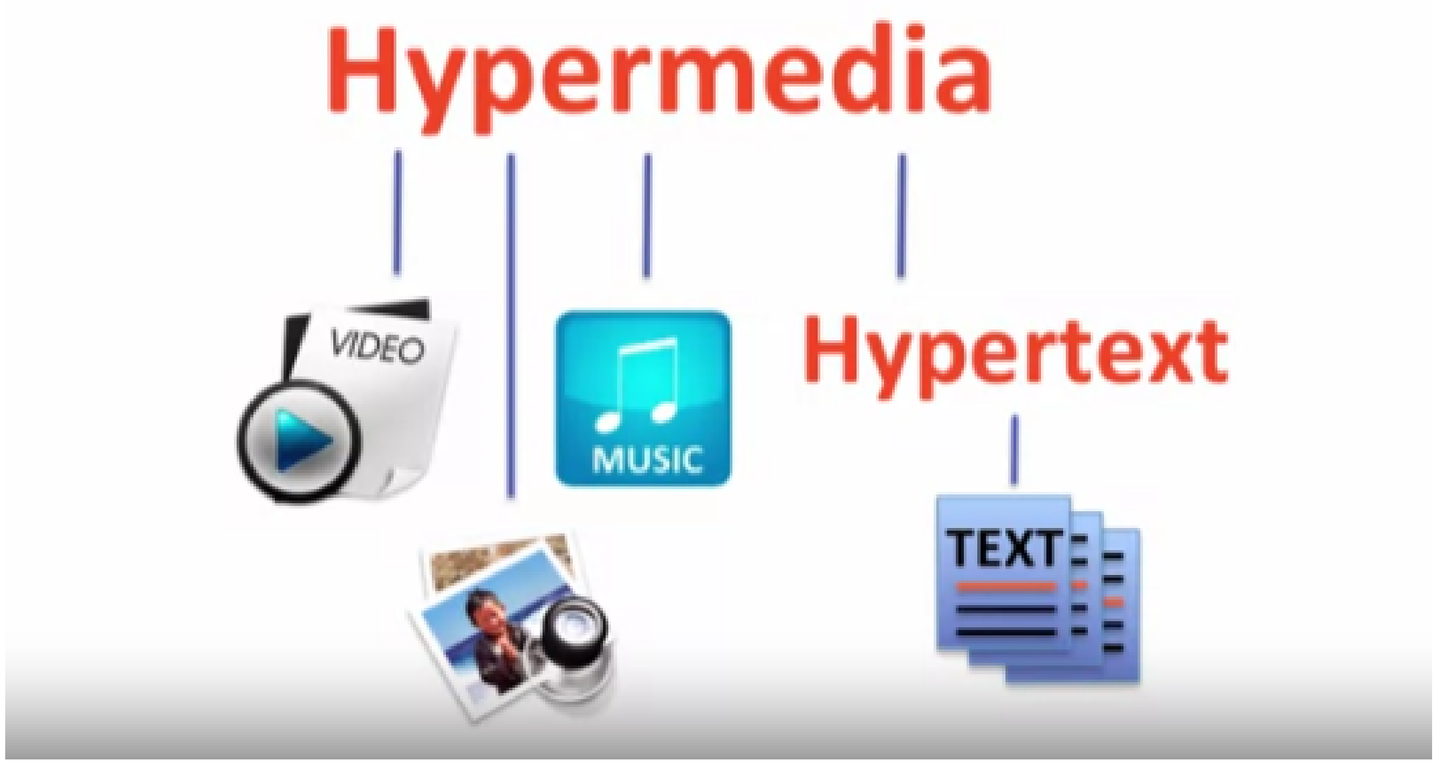
A computer running some code that serves the website

HTML

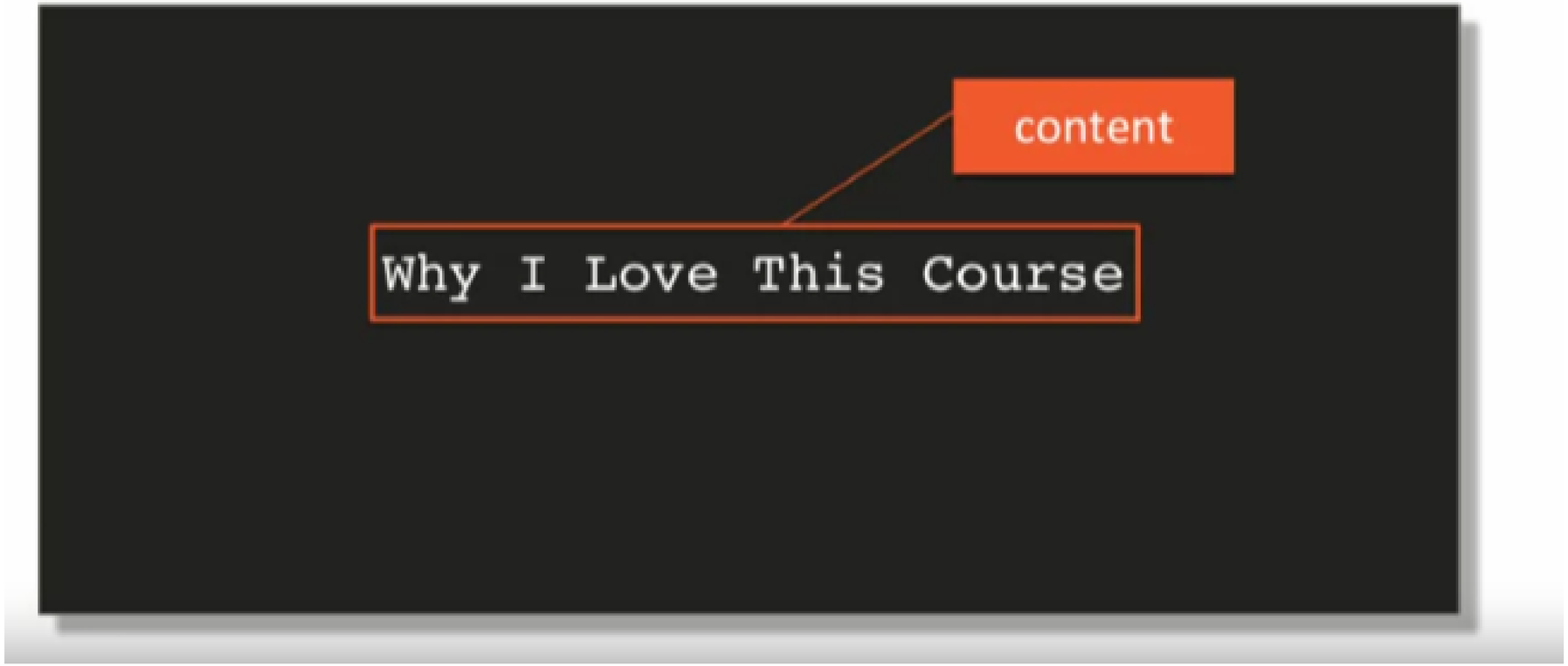
Hypertext Markup Language



Hypertext Markup Language



Hypertext **Markup** Language



Hypertext **Markup** Language

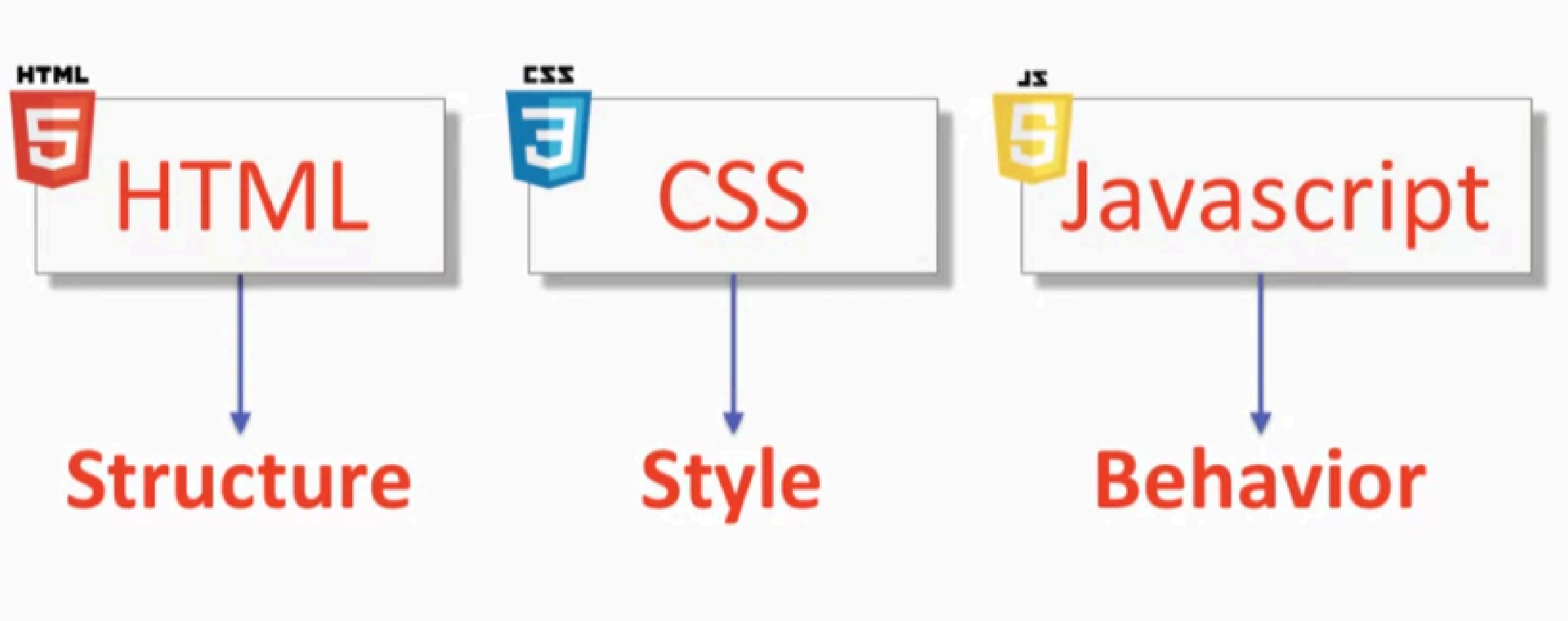
```
<!doctype html>
<html>
<head>
  <title> Why I Love This Course </title>
</head>
<body> [...]
</body>
</html>
```

Hypertext Markup **Language**

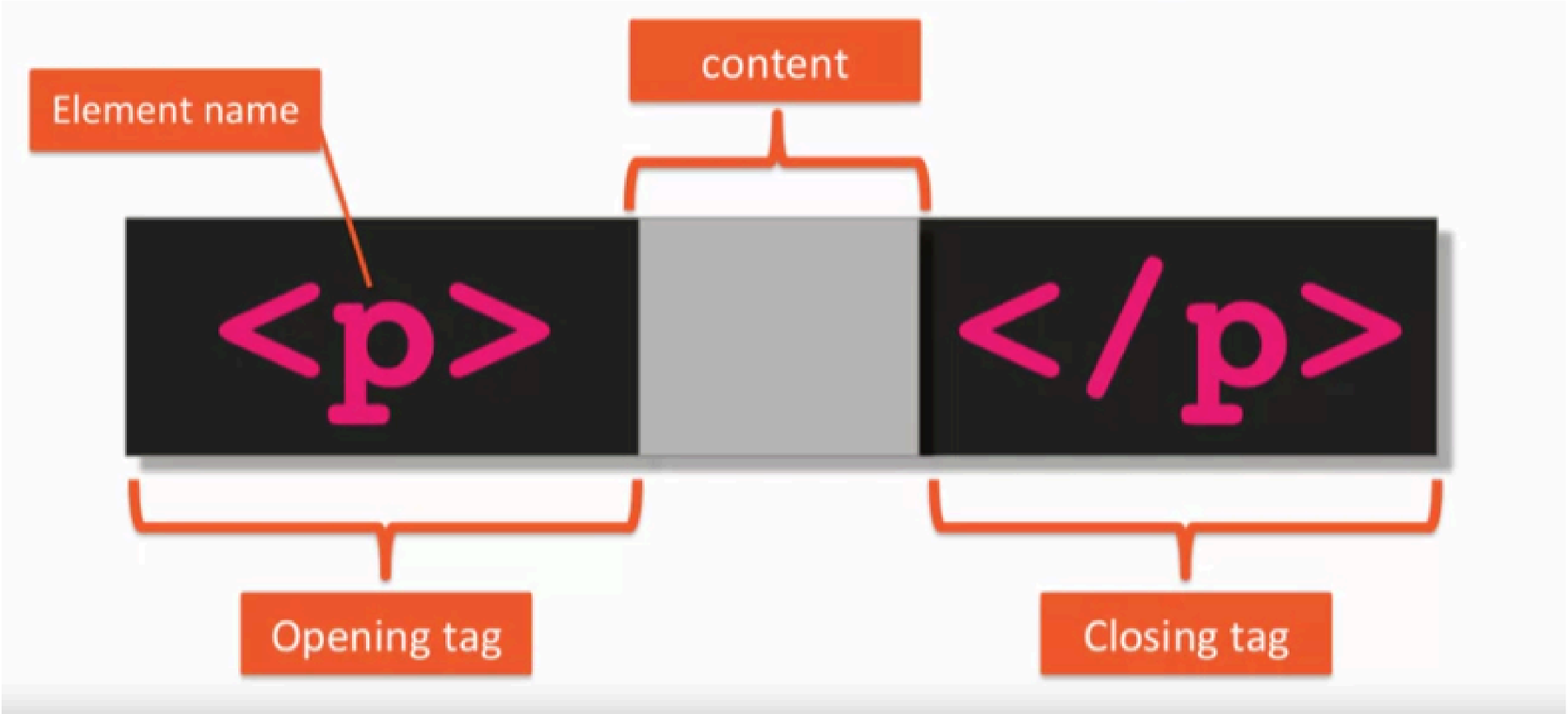
```
<h1>
  <div>Hello World!</h1>
</div>
```

```
<h1>
  <div>Hello World!</div>
</h1>
```

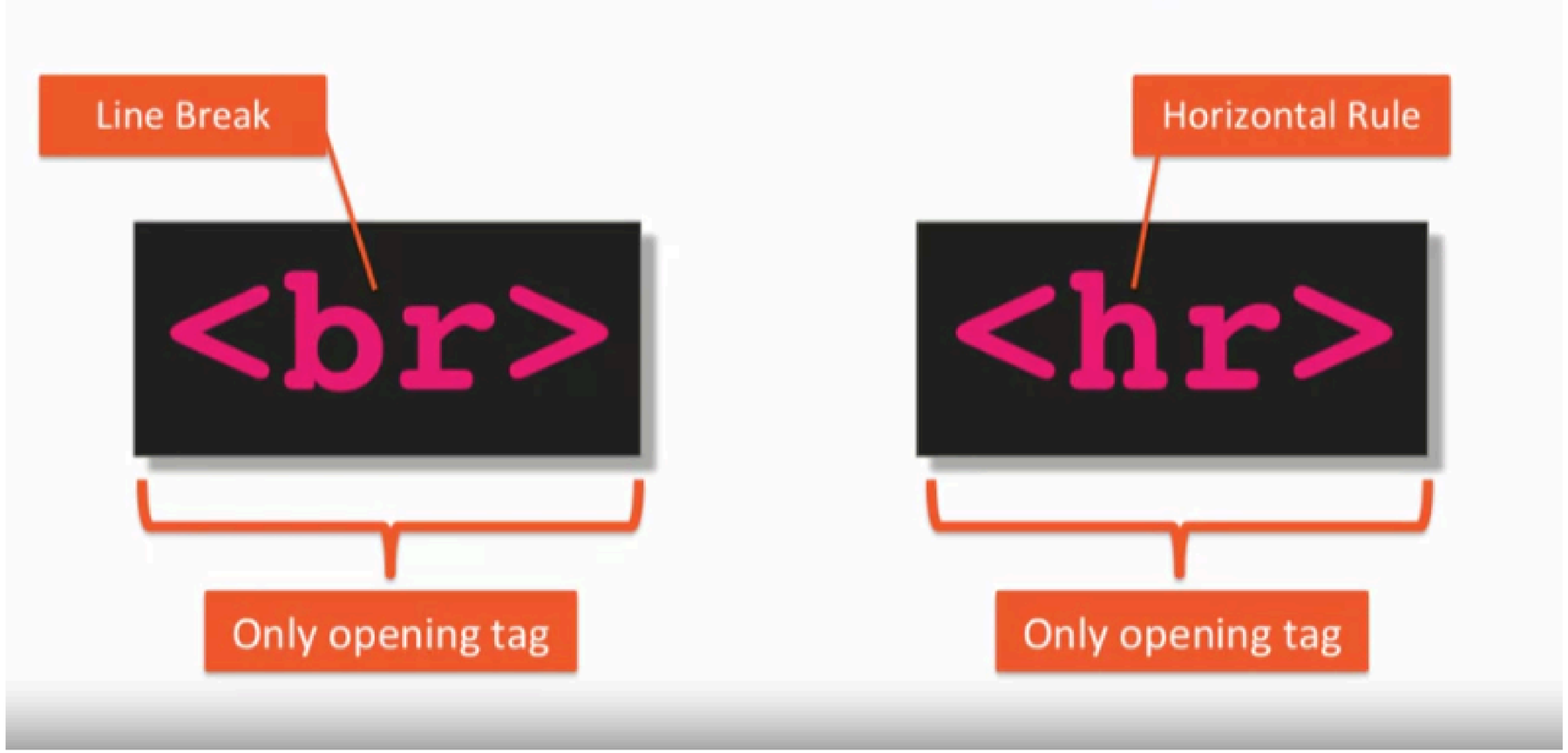
Technologies that drive the Web



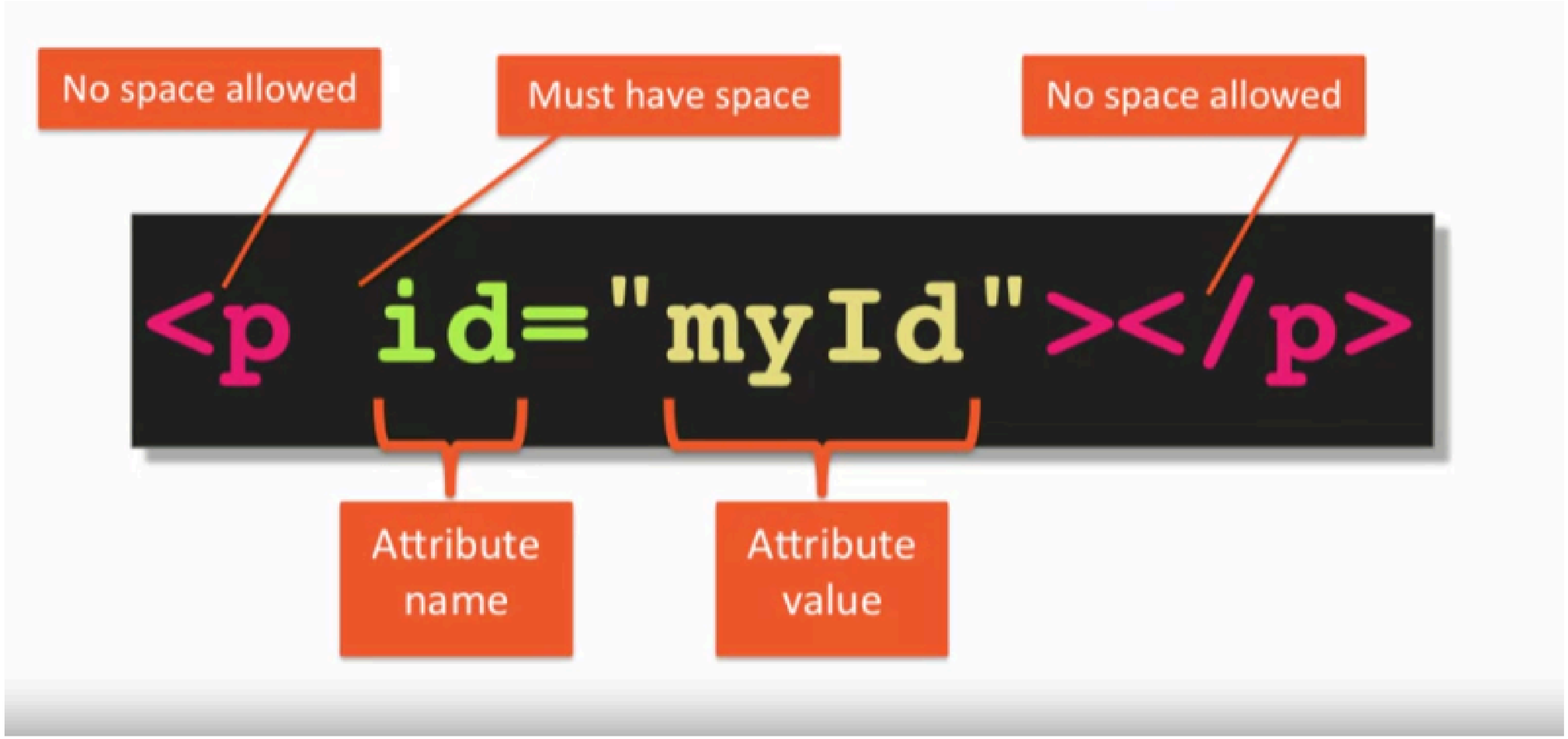
Anatomy of an HTML Tag



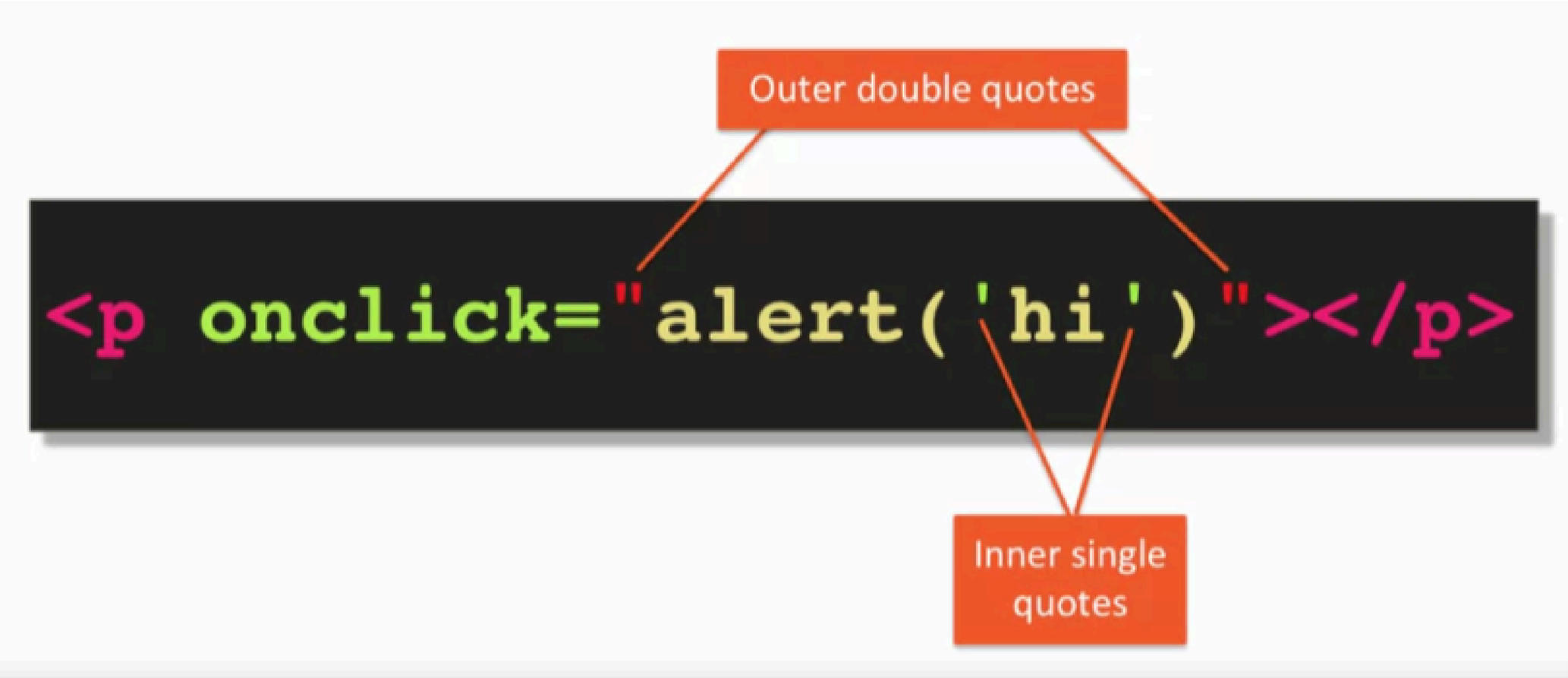
Anatomy of an HTML Tag



Anatomy of an HTML Tag



Anatomy of an HTML Tag



HTML Document Structure

index.html

workshop > index.html > html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1
    <title>HTML Document Structure</title>
</head>
<body>
    Hello World
</body>
</html>
```

top

↓

bottom

HTML Semantic Elements

se·man·tic

Relating to meaning in language or logic

HTML Semantic Elements

semantic html element

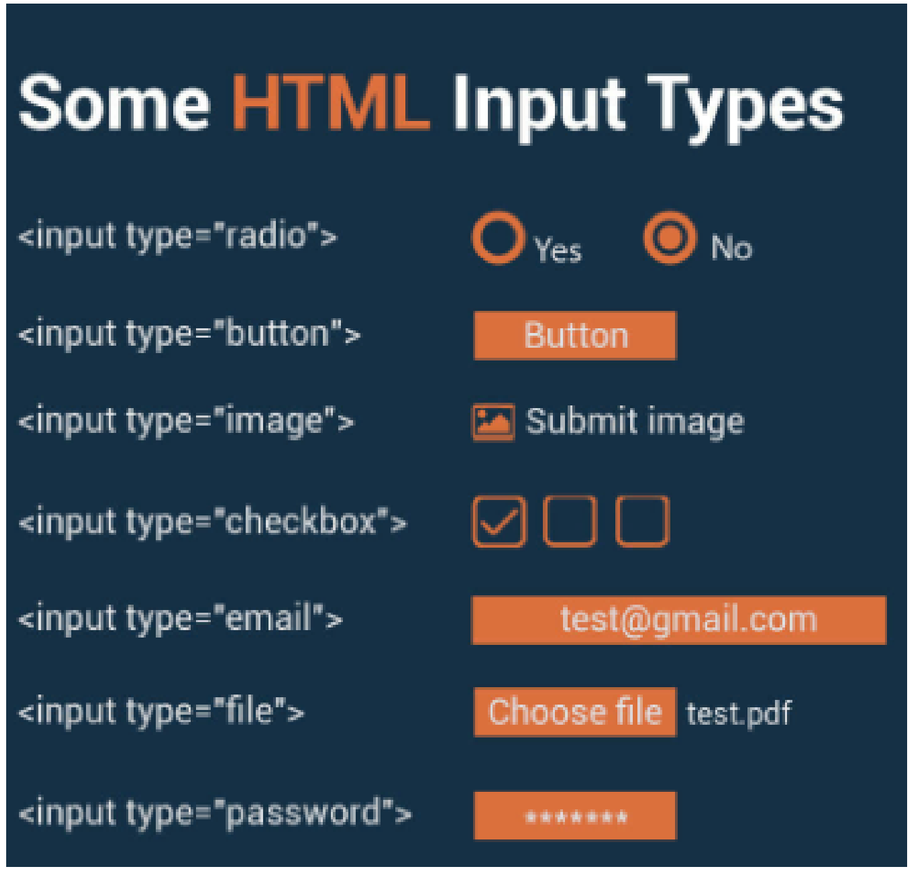
Element that implies some meaning to the content

- Human and/or machine can understand meaning of content surrounded by a semantic element better
- *May* help search engine ranking, i.e., SEO

HTML Input Element

Text Input Type

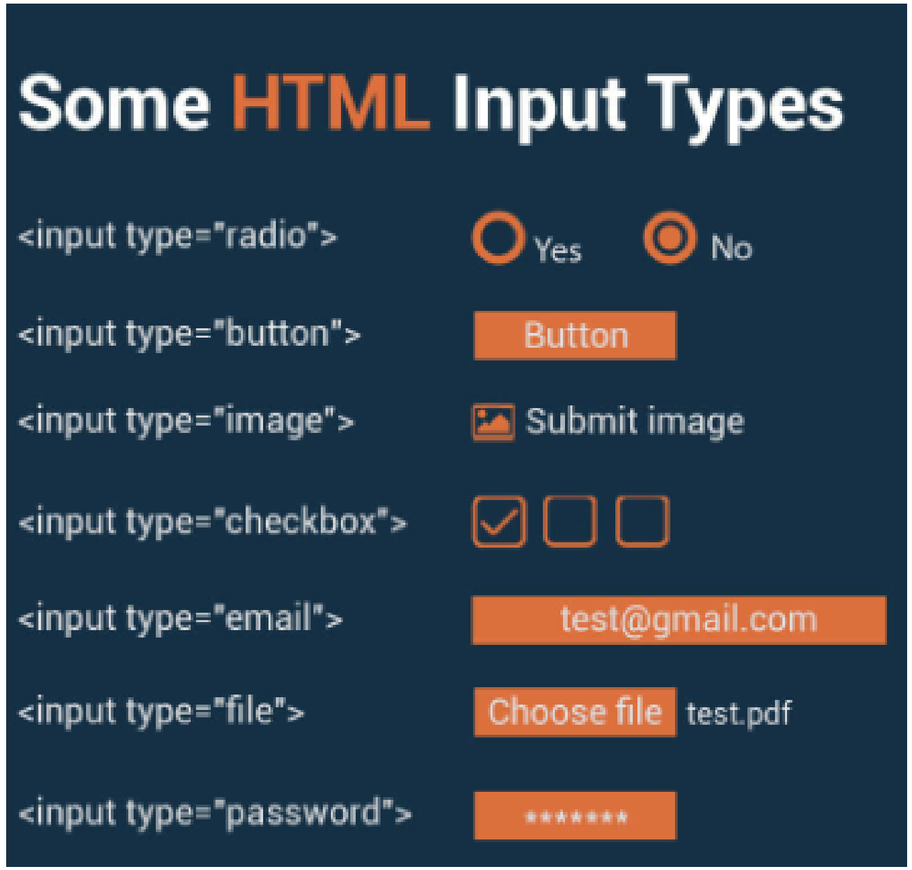
This is the default type. It allows users to enter any text into a field, such as name, email, password, etc.



HTML Input Element

Radio and Checkbox

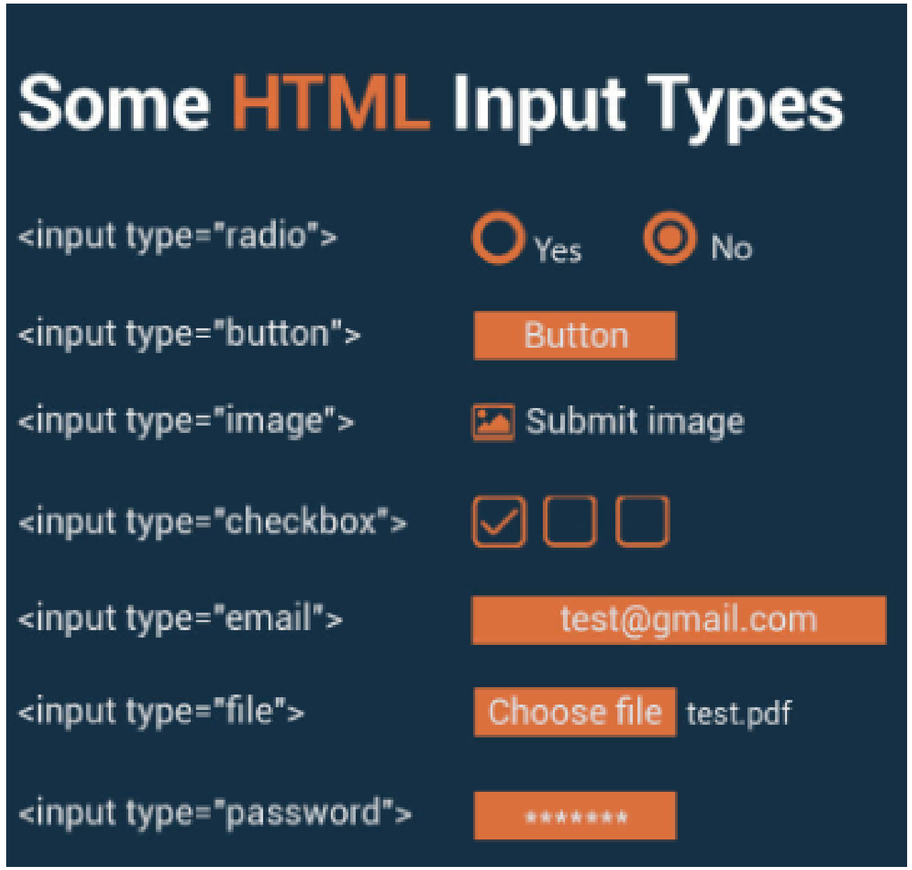
Allows users to select one option from a group, while checkboxes allow multiple selections.



HTML Input Element

File and Button

File input lets users upload files, Button input type is used for click funtions



List

List provide a natural and commonly used grouping of contents

List are used for structuring navigation portions of the webpage

Links

Internal Linking to other pages in the site

External Linking to other websites

Linking to sections of a document

Images

Images can enhance websites design and presentation

Specificy image width and heights attributes whenever it is possible

List, Links, and Images

index.html

workshop > index.html > html > body > nav

```html
<!DOCTYPE html>
<html lang="en">
<head>···
</head>
<body>
    <header>···
    </header>
    <nav>
        <h2>Links List</h2>
        <ul>
            <li>Link 1</li>
            <li><a href="#">Link 2</a></li>
            <li>Link 3</li>
            <li><a href="#">Link 4</a></li>
            <li>Link 5</li>
        </ul>
    </nav>
    <article>
        <h2>Image</h2>
        <img src="https://via.placeholder.com/400" alt="Placeholder Image">
    </article>
</body>
</html>
```

CSS

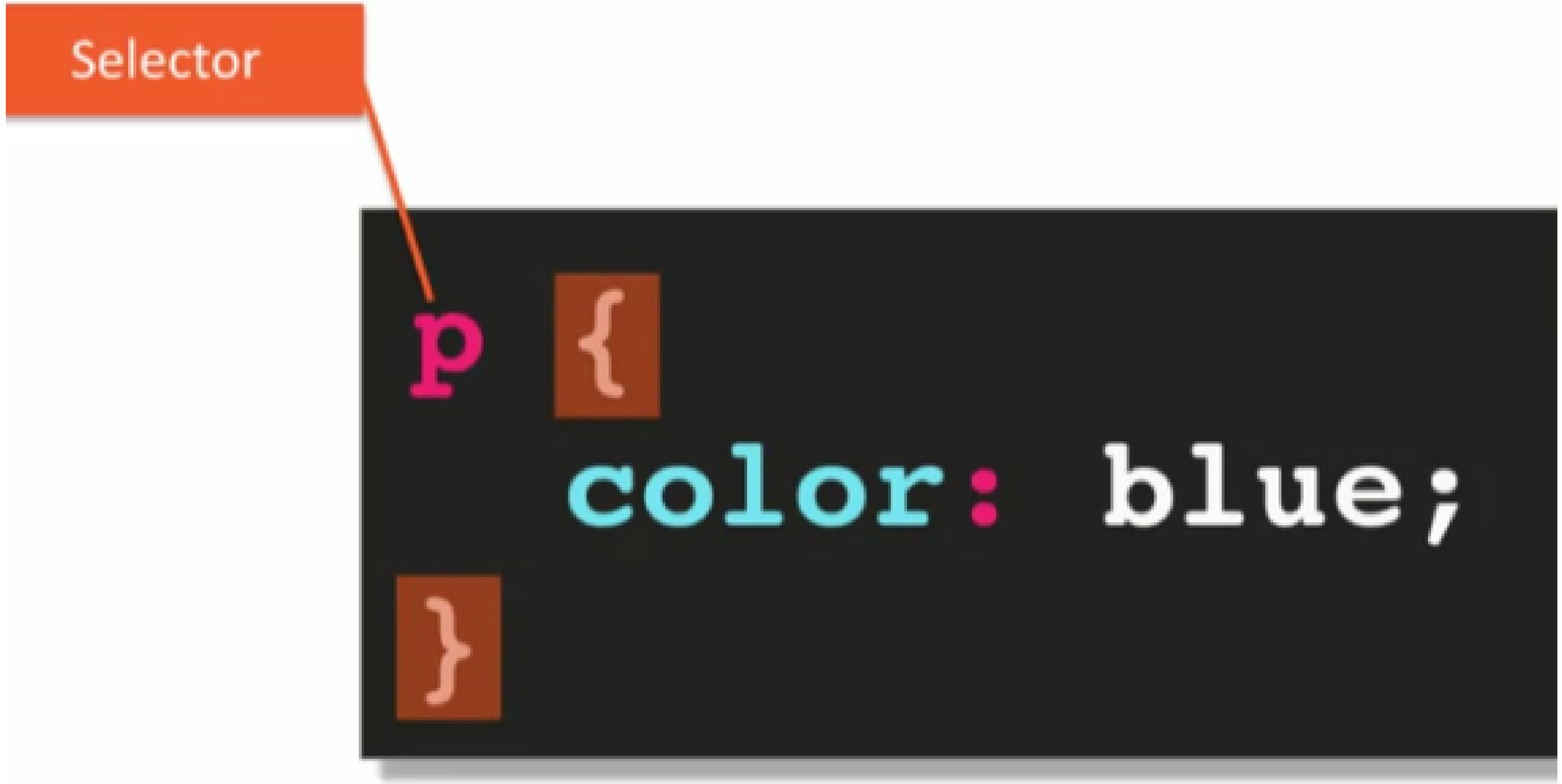
Cascading Style Sheet

CSS is an incredibly **powerful** technology

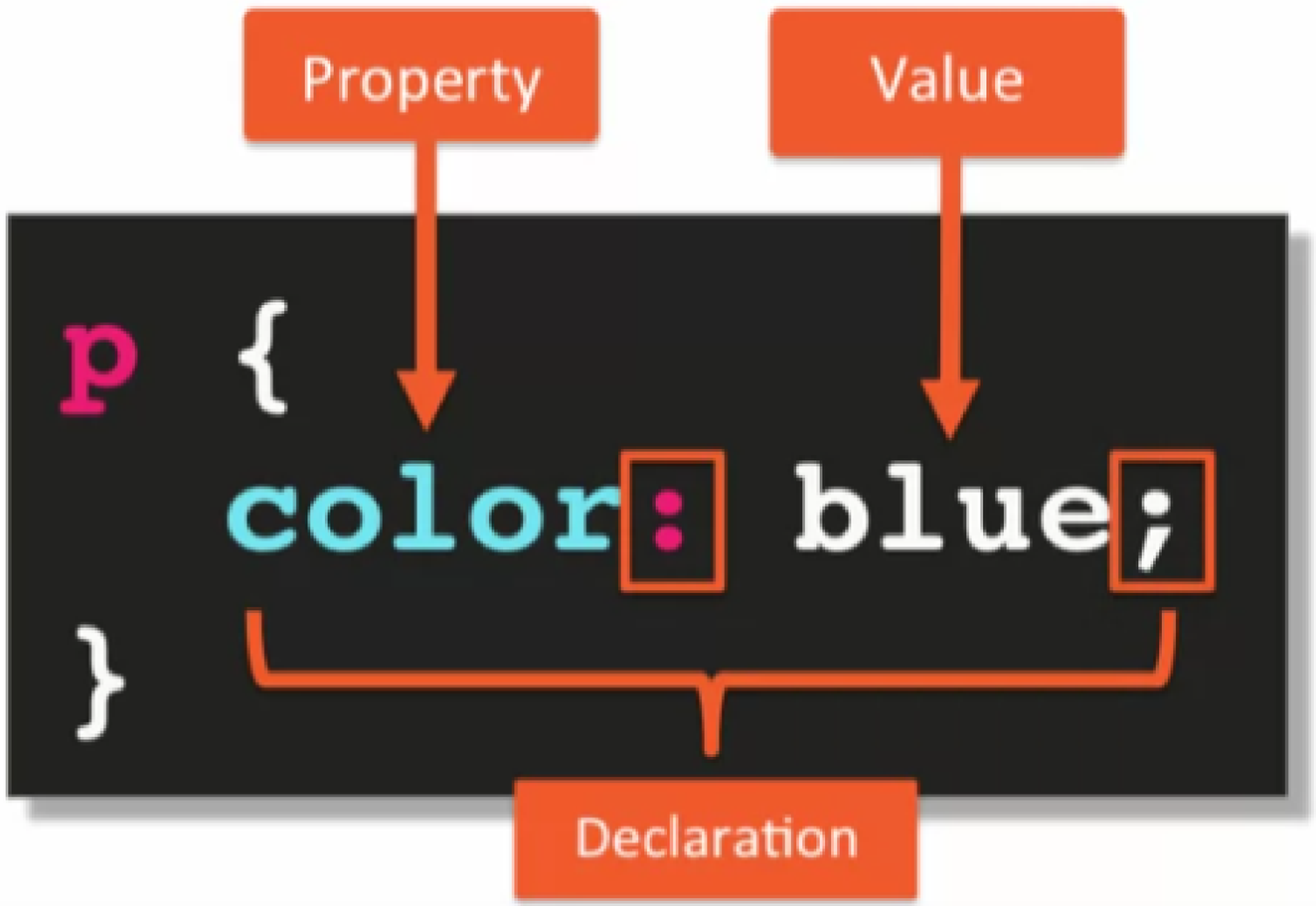
User experience of content matters

csszengarden.com

Anatomy of CSS Rule



Anatomy of CSS Rule



Anatomy of CSS Rule

```
p {
  color: blue;
  font-size: 20px;
  width: 200px;
}

h1 {
  color: green;
  font-size: 36px;
  text-align: center;
}
...
```

Stylesheet

CSS Selector - Element

```
p {
  color: blue;
}
```

Blue text

```
...
<p>...</p>
...
<p>...</p>
```

Blue text

CSS Selector - Class

```
.blue {
  color: blue;
}
```

```
...
<p class="blue">...</p>
<p>...</p>
<div class="blue">...</div>
...
```

Blue text

Blue text

CSS Selector - Id

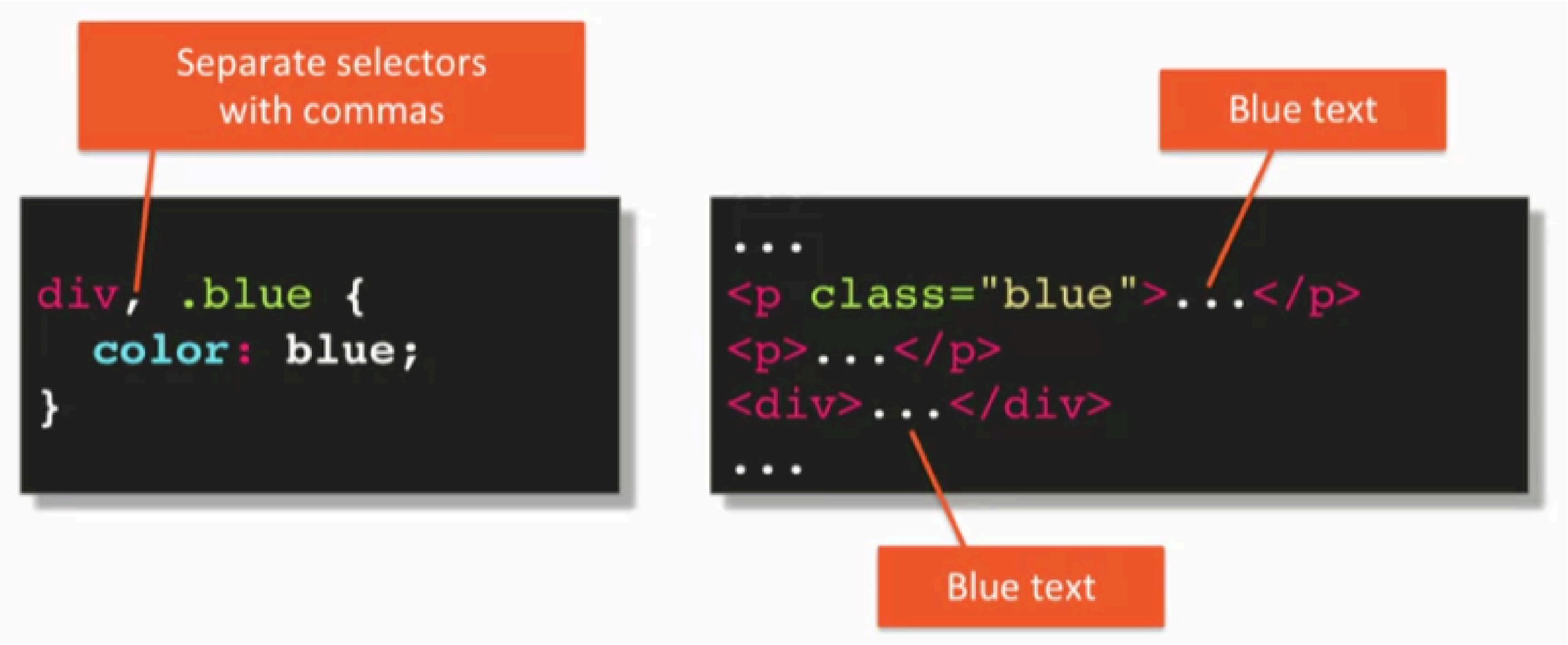
```
#name {
   color: blue;
}
```

```
...
<p>...</p>
<div id="name">...</div>
...
```

Unaffected

Blue text

CSS Group Selector



CSS Combining Selector

Element Class Selectors

```
p.big {
  font-size: 20px;
}
```

```
...
<p class="big">...</p>
<div class="big">...</div>
...
```

Text size 20px

Unaffected text

Child Selectors

Every **p** that is a **direct** child of **article**

```
article > p {
  color: blue;
}
```

Child Selectors

```
article > p {
  color: blue;
}
```

```
<article>...
  <p>...</p>
</article>
...
<p>...</p>
<article>...
  <div><p>...</p></div>
</article>
```

Blue text

Unaffected text

Unaffected text

Descendant Selectors

Every **p** that is inside (at any level) of **article**

```
article p {
   color: blue;
}
```

Descendant Selectors

```
article p {
  color: blue;
}
```

```
<article>...
  <p>...</p>
</article>
...
<p>...</p>
<article>...
  <div><p>...</p></div>
</article>
```

Blue text

Unaffected text

Blue text

Not limited to Element Selectors

```
.colored p {
  color: blue;
}
```

Every **p** that is inside (at any level) an element with **class="colored"**

```
article > .colored {
  color: blue;
}
```

Every element with **class="colored"** that is a <u>direct</u> child of **article** element

CSS Combining Selector

Combining selectors

- Element with `class` selector (`selector.class`)
- Child (direct) selector (`selector > selector`)
- Descendent selector (`selector selector`)

Pseudo Class Selectors

: link	: active
: visited	: nth-child(...)
: hover	

Pseudo Class Selectors

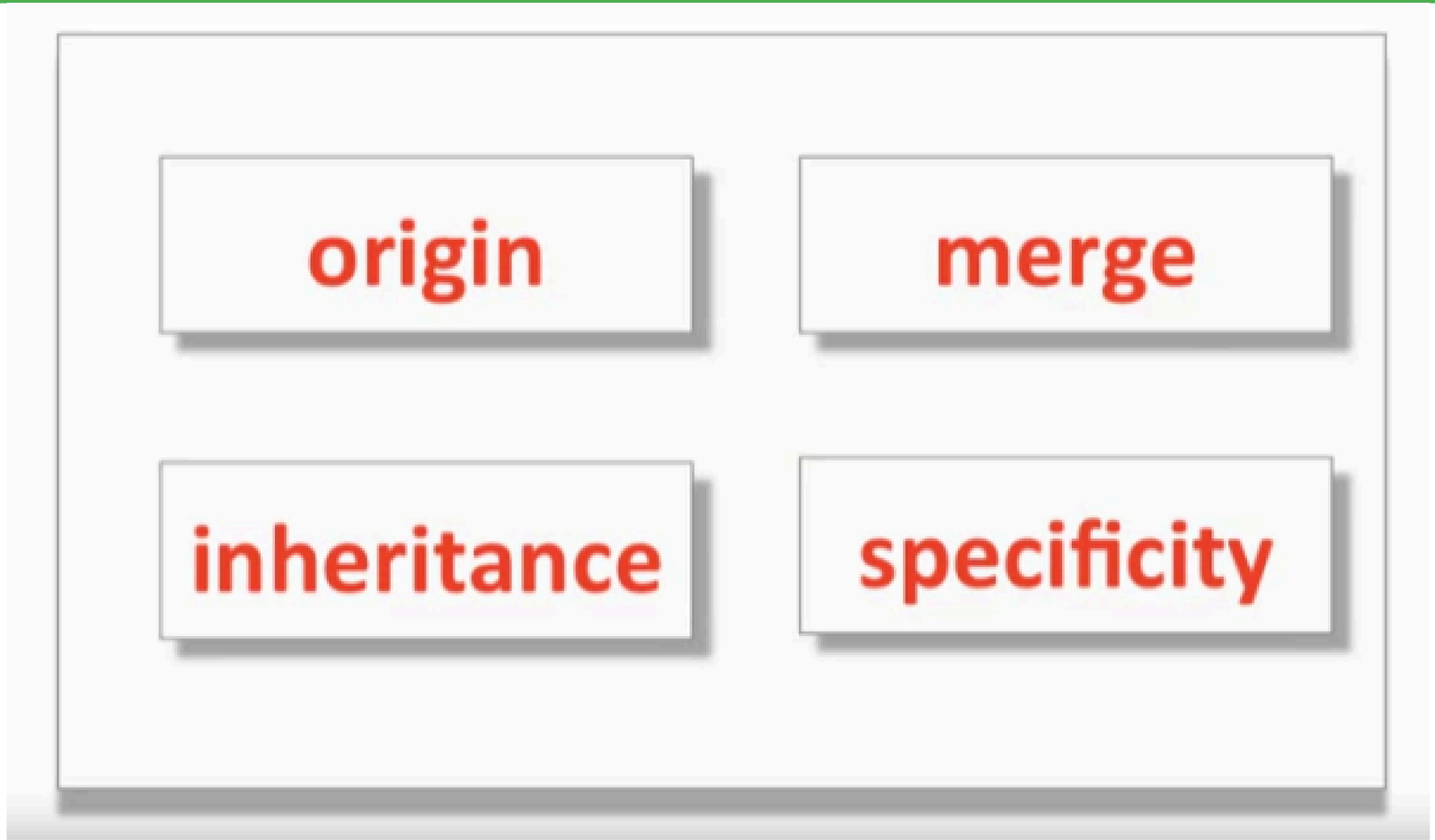
- Pseudo-class selectors are very powerful
- Make sure selectors are still readable
- Simple/Readable > Complicated/Tricky

Style Placement

- External Style Sheet (Reusability & Consistency)
- Head Style (Usually to override external)
- In-line Style (Testing purposes)

CSS Conflict Resolution

Cascading Algorithm Concepts & Conflict Resolutions



Origin Precedence (when in conflict)

simple rule

Last Declaration Wins

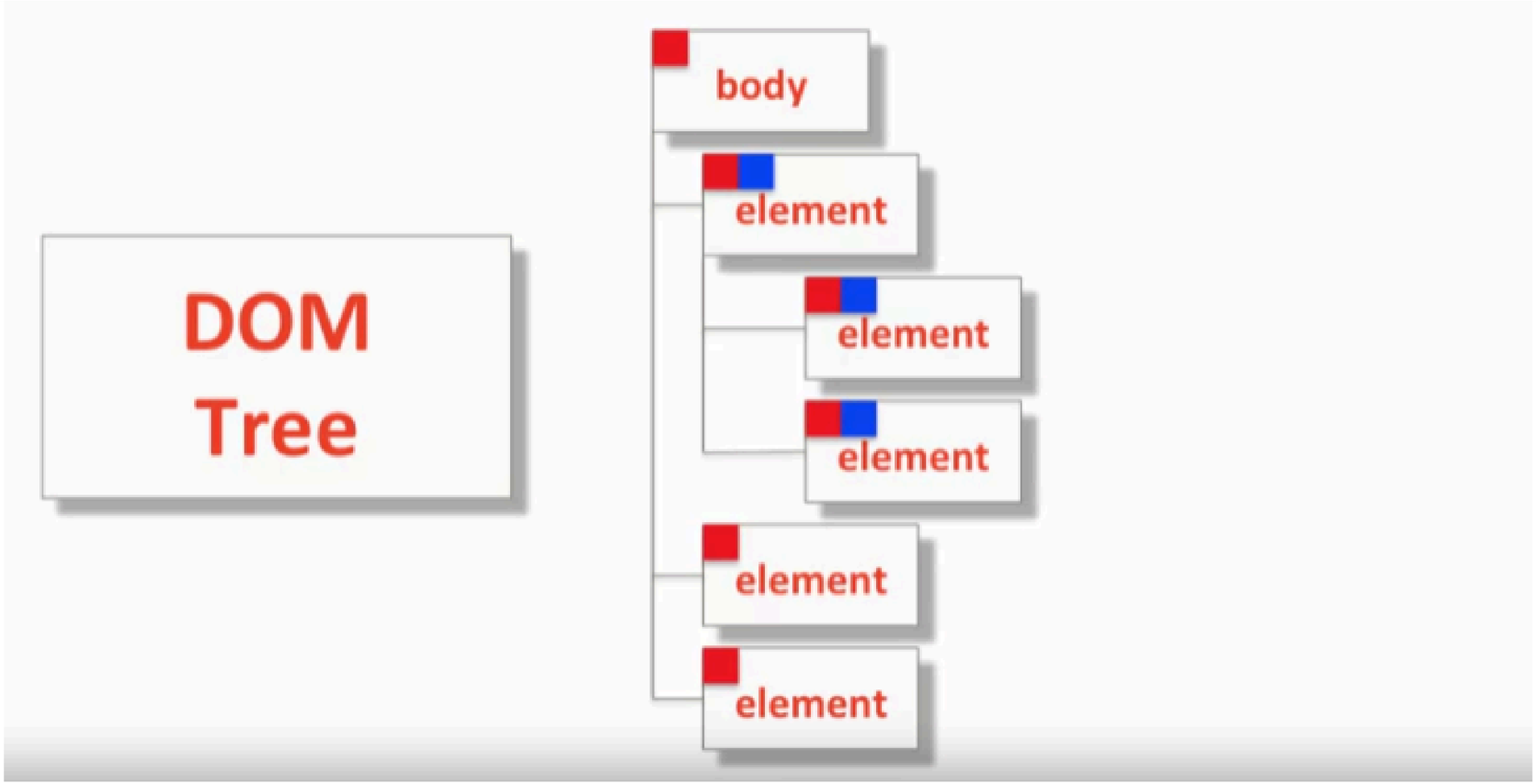
- Remember, HTML is processed sequentially, top to bottom.
- For precedence, think of external CSS as declared at the spot where it's linked to.

Origin Precedence (when no conflict)

even simpler rule

Declarations Merge

Inheritance



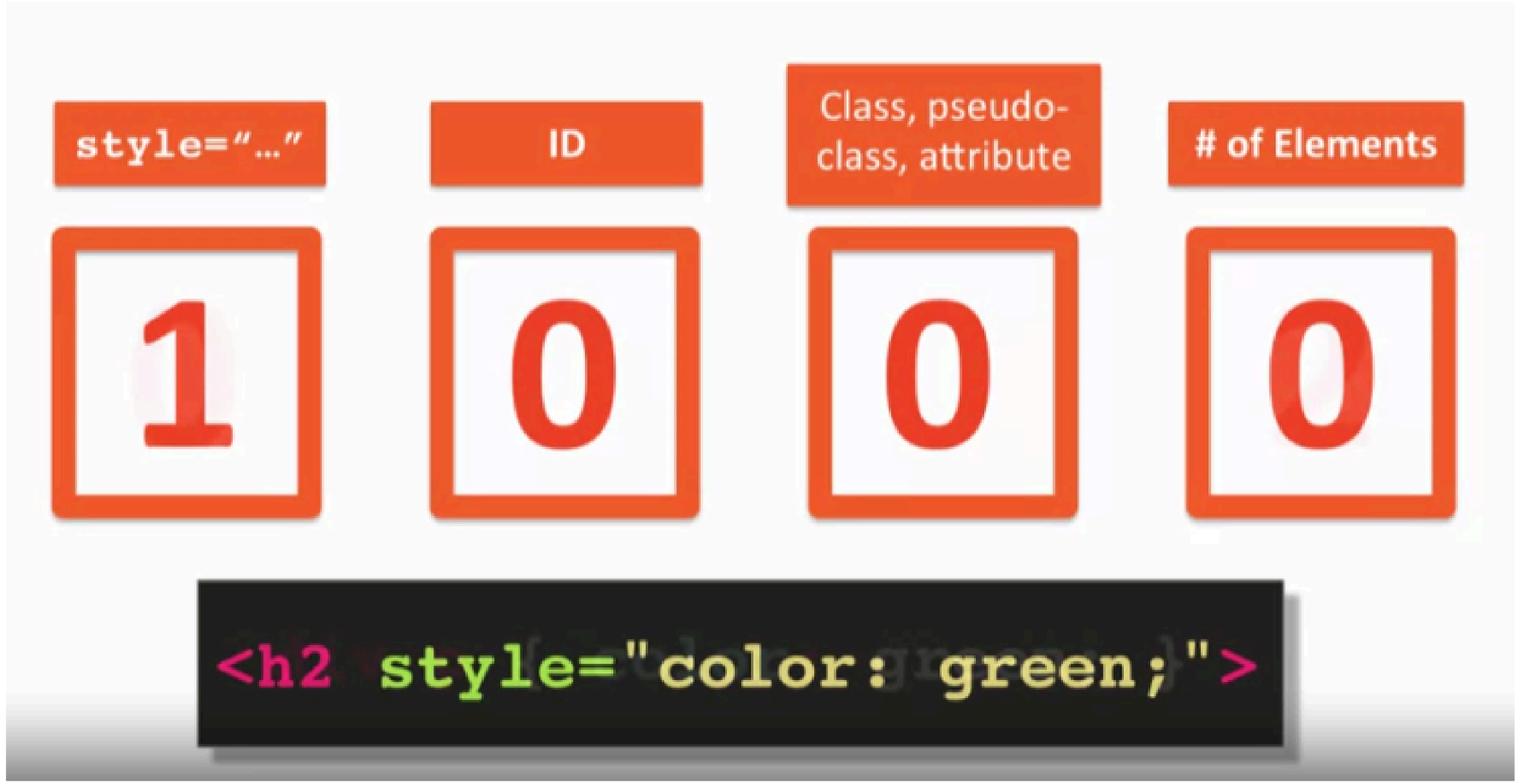
Specificity

simple rule

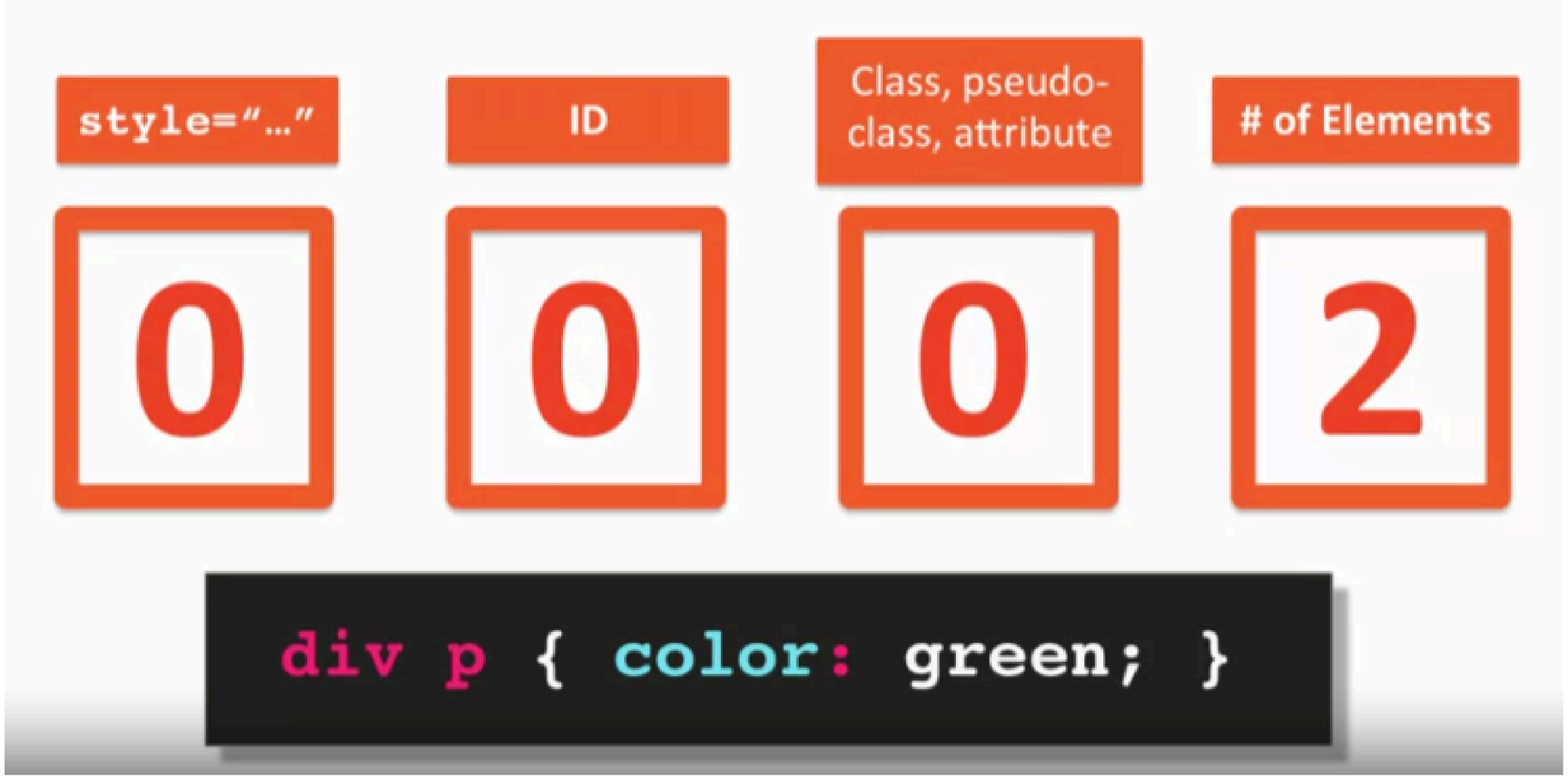
Most Specific Selector Combination Wins

Specificity

Specificity



Specificity



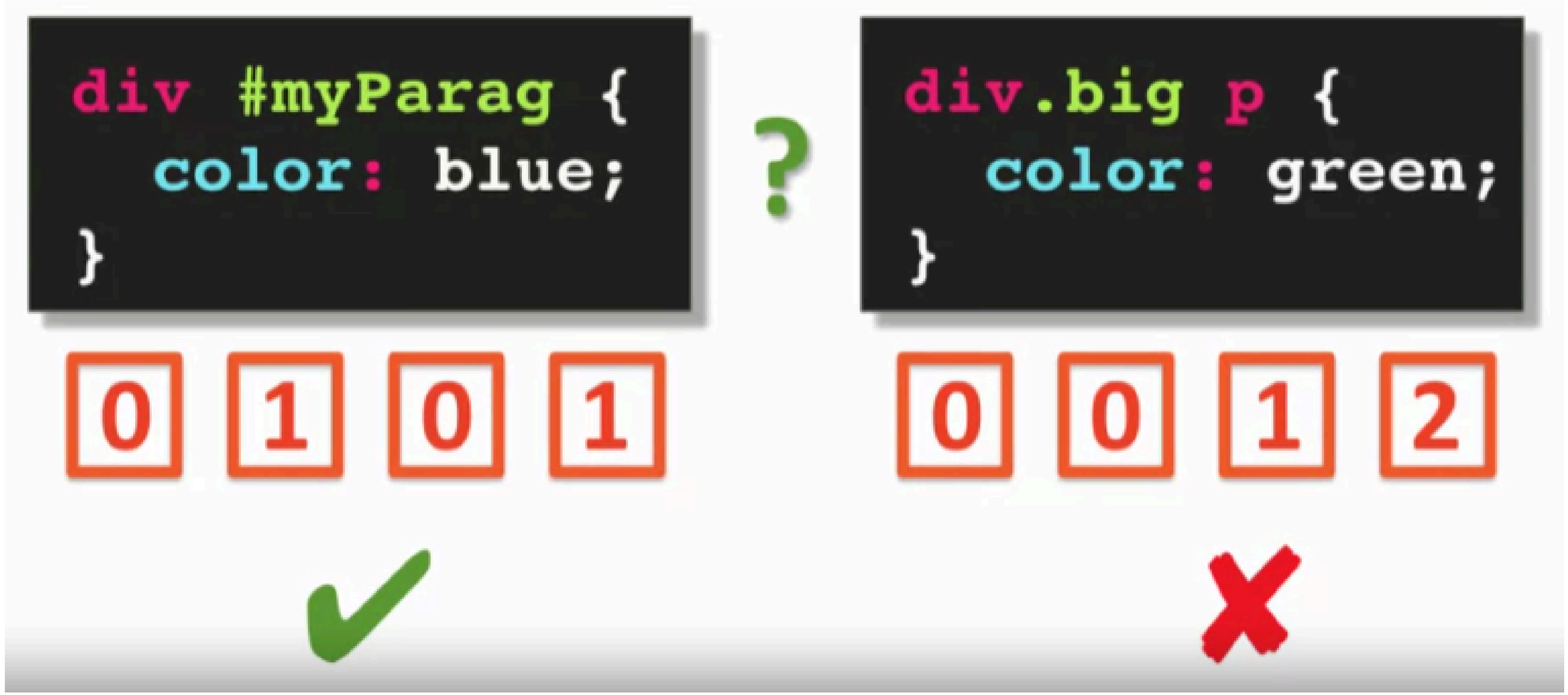
Specificity

```
div #myParag {
  color: blue;
}
```

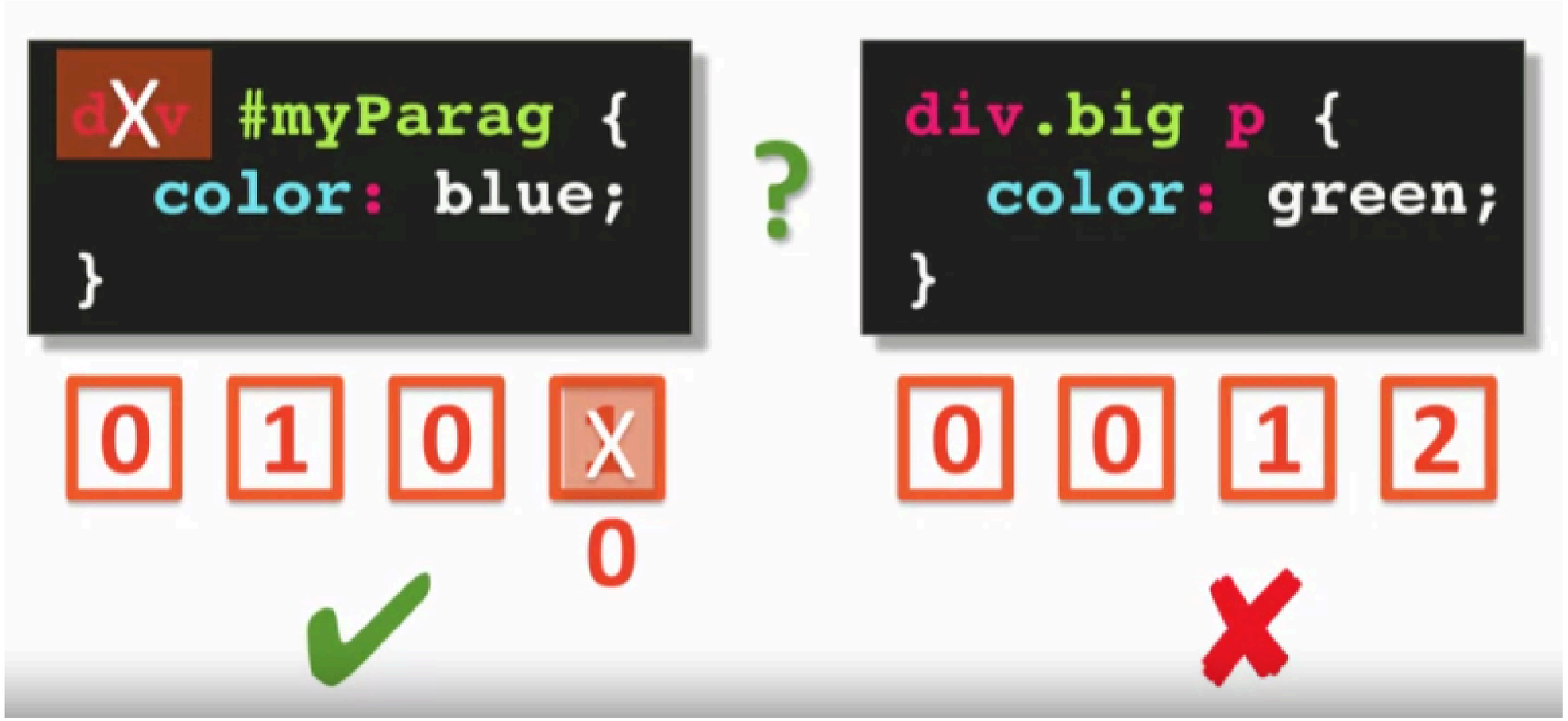
?

```
div.big p {
  color: green;
}
```

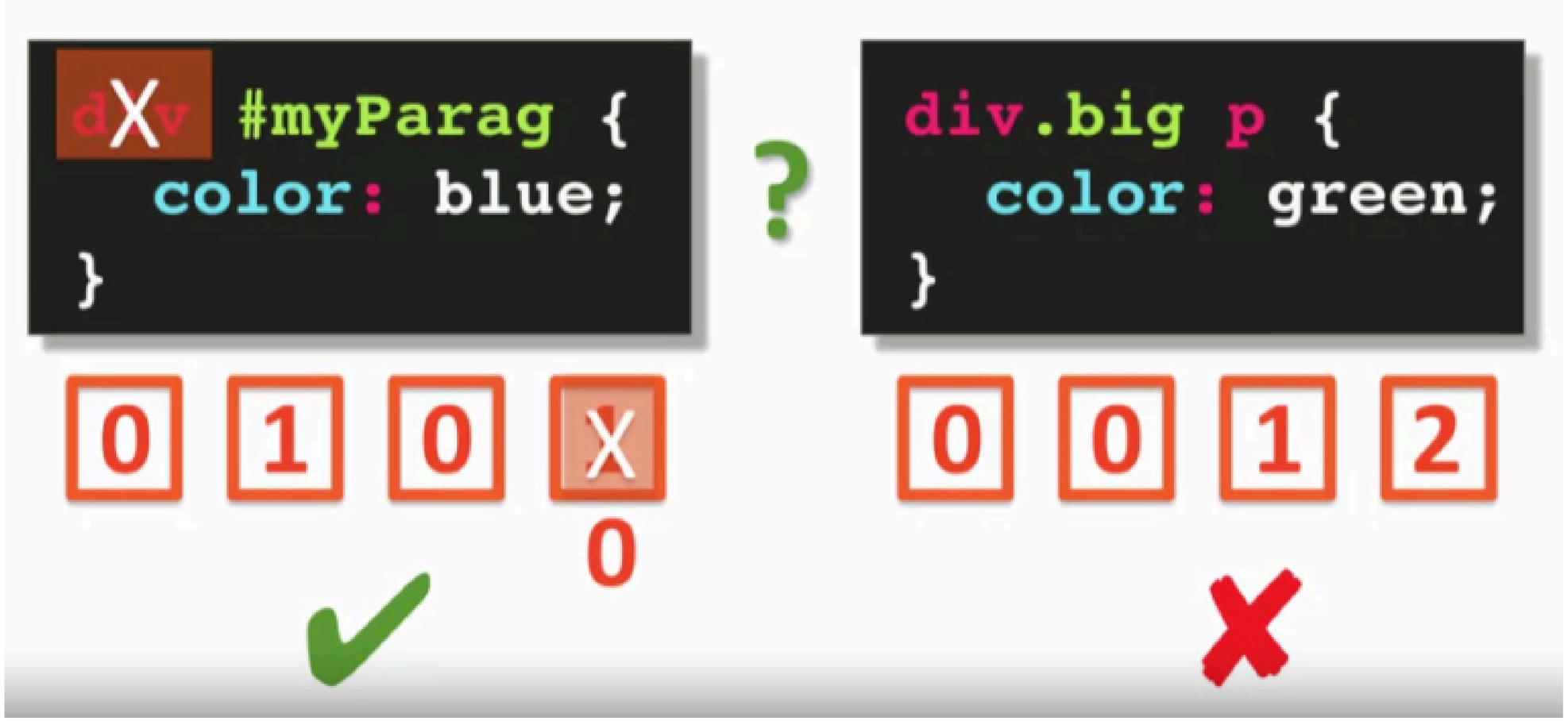
Specificity



Specificity



Specificity



JAVASCRIPT

Variables

```
var message = "hi";
```

- Variable definition should always start with ‘var’
- No types are declared
  - JS is dynamically typed language
  - Same variable can hold different types during the life of the execution

Data Types

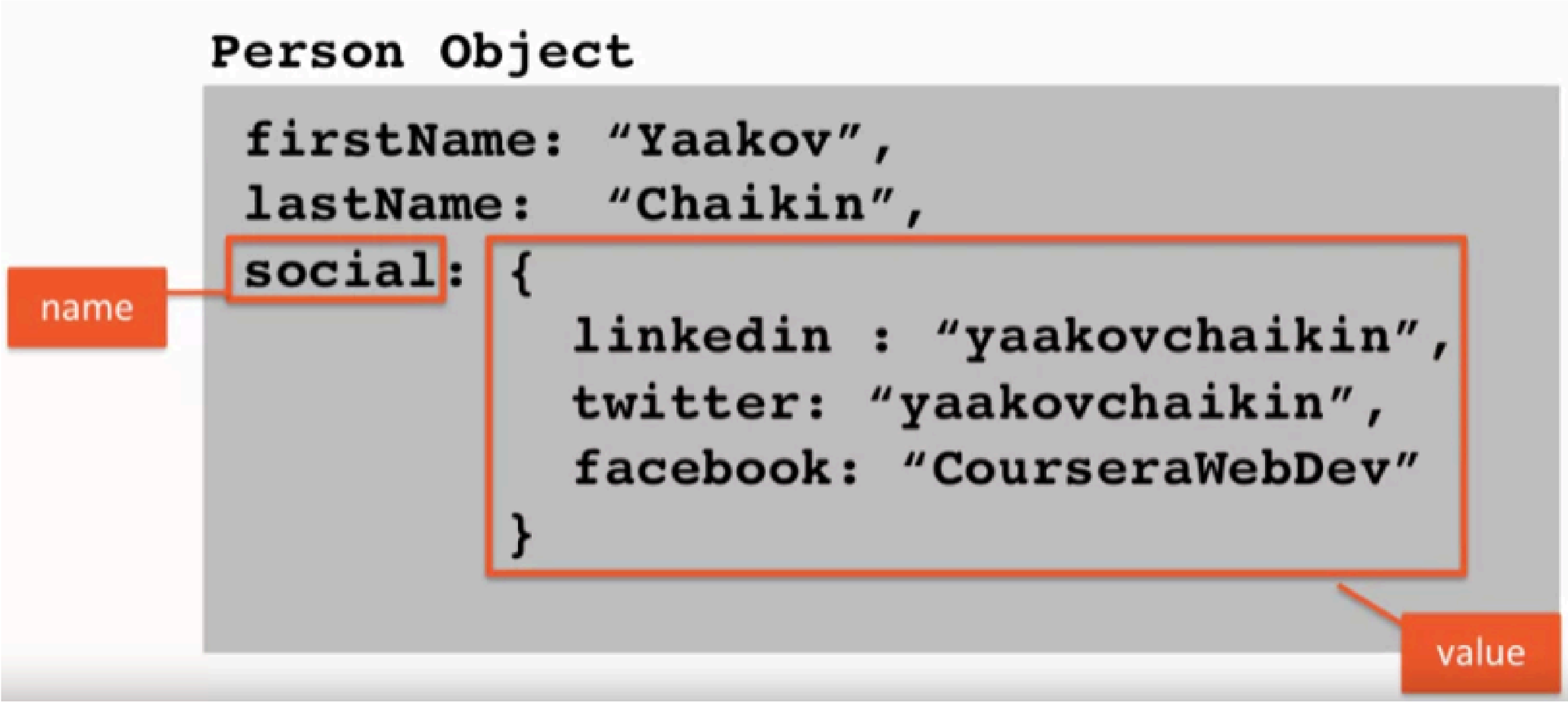
A type is a particular data structure.

- Each language defines some built-in types
- Built-in types can be used to build other data structures
- JS has 7 built-in types: 6 primitive and 1 Object type

Object Type

Object is a collection of name/value pairs

Object Type


Person Object
firstName: "Yaakov",
lastName: "Chaikin",
social: {
linkedin : "yaakovchaikin",
twitter: "yaakovchaikin",
facebook: "CourseraWebDev"
}
name
value

Primitive Types

Primitive type represents a *single, immutable* value

- Single value, i.e., *not* an object
- Immutable means once it's set, it can't be changed
  - Value becomes read-only
  - You can create another value based on an existing one

Primitive Type: Boolean

Boolean can only have 2 values: `true` or `false`

Primitive Type: Undefined

Undefined signifies that no value has ever been set

- Can only have one value: `undefined`
- You *can* set a variable to `undefined`, but you *should NEVER do it*
  - Its meaning is that it's never been defined, so defining it to `undefined` is counter to its core meaning

Primitive Type: Null

Null signifies lack of value

- As opposed to `undefined`, which is lack of definition
- Can only have one value: `null`
- It's ok to explicitly set a variable to `null`

Primitive Type: Number

Number is the only numeric type in Javascript

- Always represented under the hood as double-precision 64-bit floating point
- JS does *not* have an integer type
  - Integers are a subset of doubles instead of a separate data type

Primitive Type: String

String is sequence of characters used to represent text

- Use either single or double quotes, i.e., ‘text’ or “text”

Functions

```
function a () {...}

a();
```

Defines function

Executes function (aka invokes function)

Function Scope

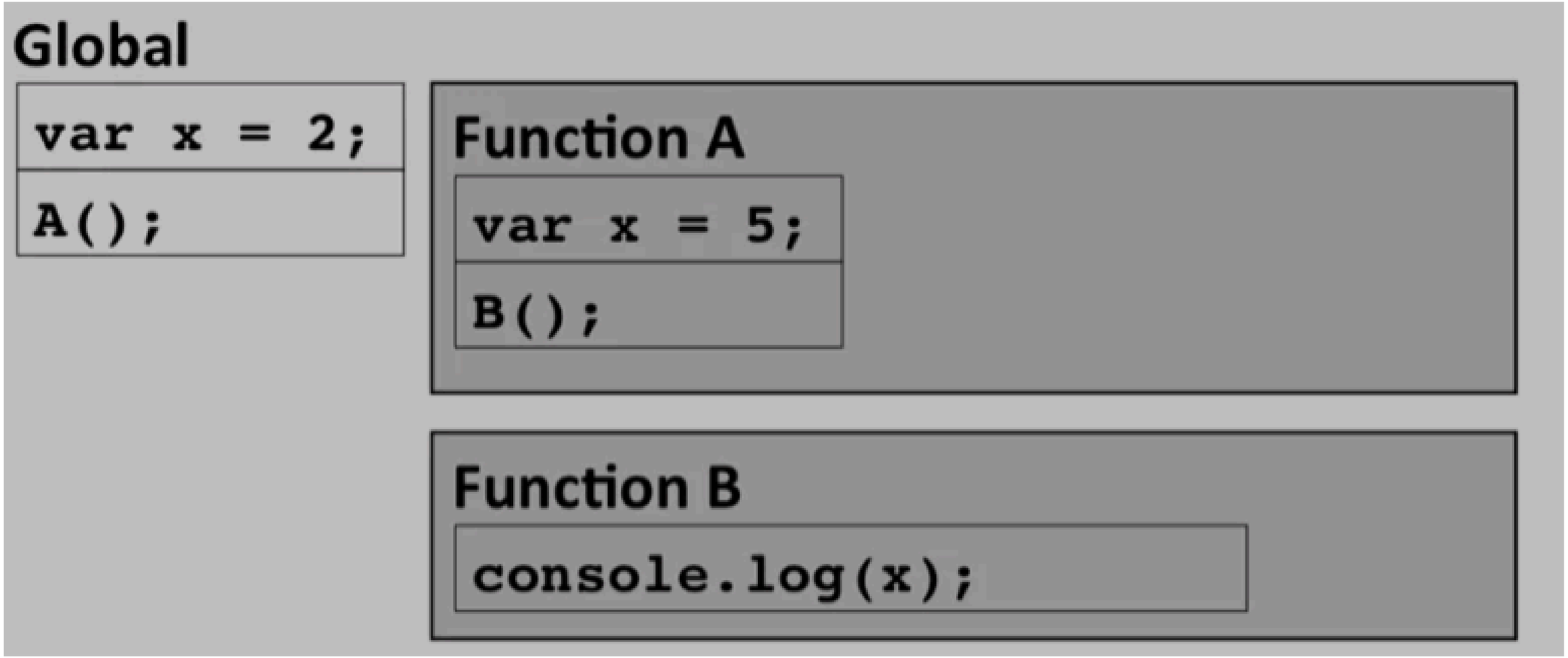
Global

- Variables and functions defined here are available everywhere

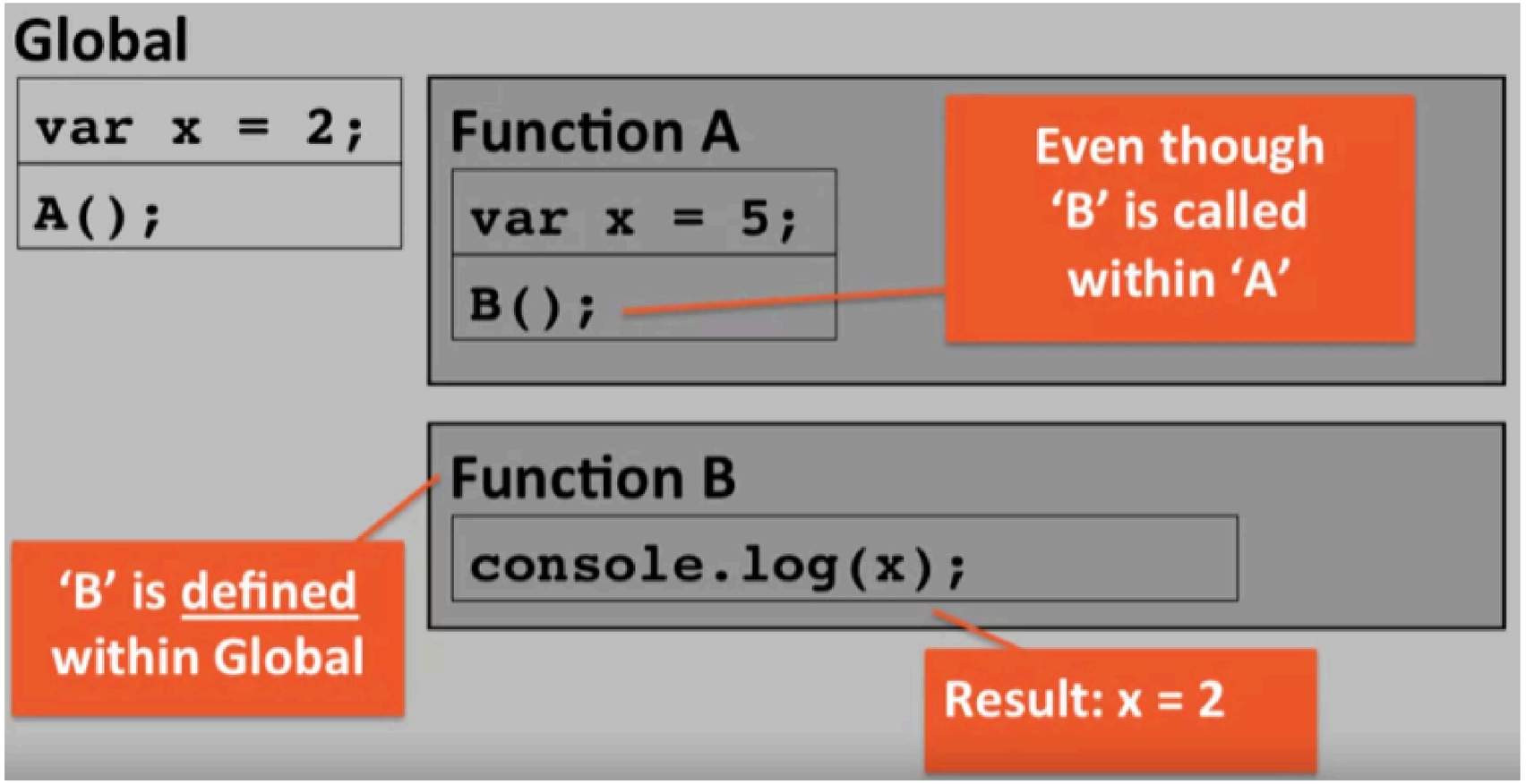
Function
aka lexical

- Variables and functions defined here are available only within this function

Function Scope



Function Scope



DOM Manipulation

Document Object represents the entire HTML document.

DOM Manipulation

Accessing Elements:

document.getElementById(),
document.getElementsByClassName(),
document.getElementsByTagName(),
document.querySelector(),
document.querySelectorAll()

DOM Manipulation

Manipulating Elements:

element.classList,

element.innerHTML,

element.innerText,

element.textContent,

element.setAttribute(),

element.style,

element.addEventListener()

Events

Mouse Events:

mouseover,

mouseout,

mousemove,

mousedown,

mouseup

Keyboard Events:

keydown,

keyup,

keypress

Events

Form Events:

submit,

change,

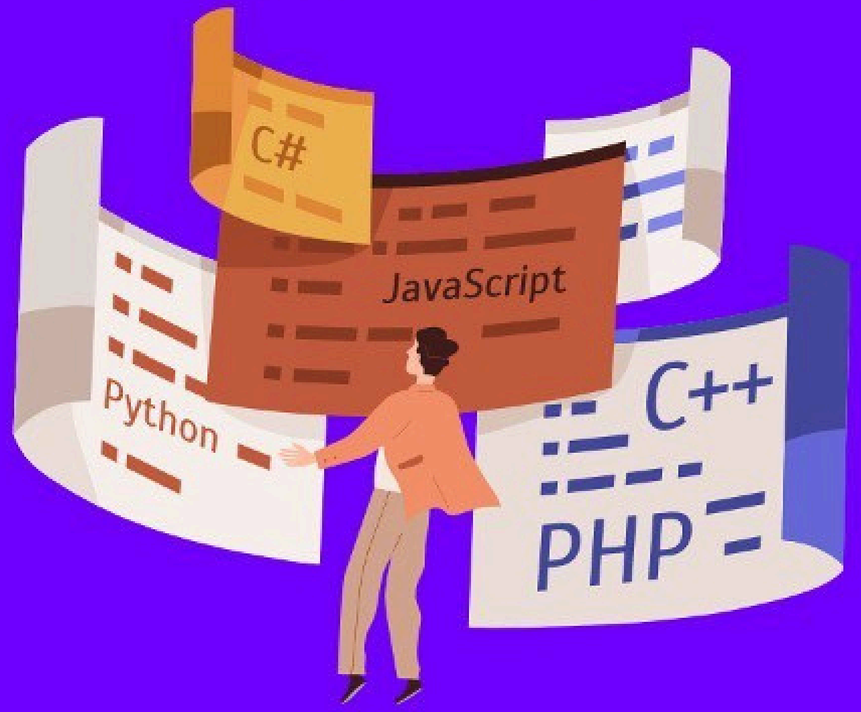
input,

focus,

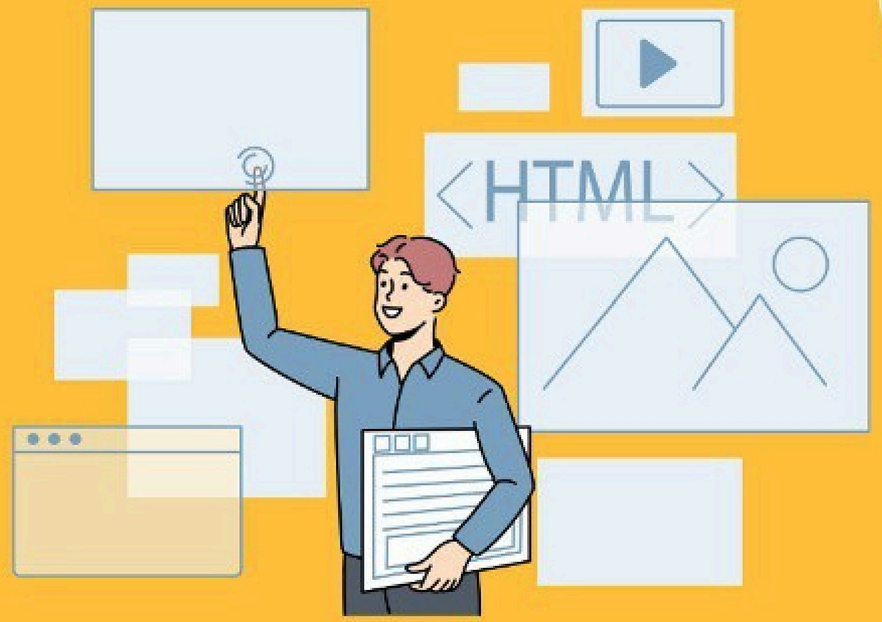
blur

heydevs

WHICH DEV ARE YOU ?

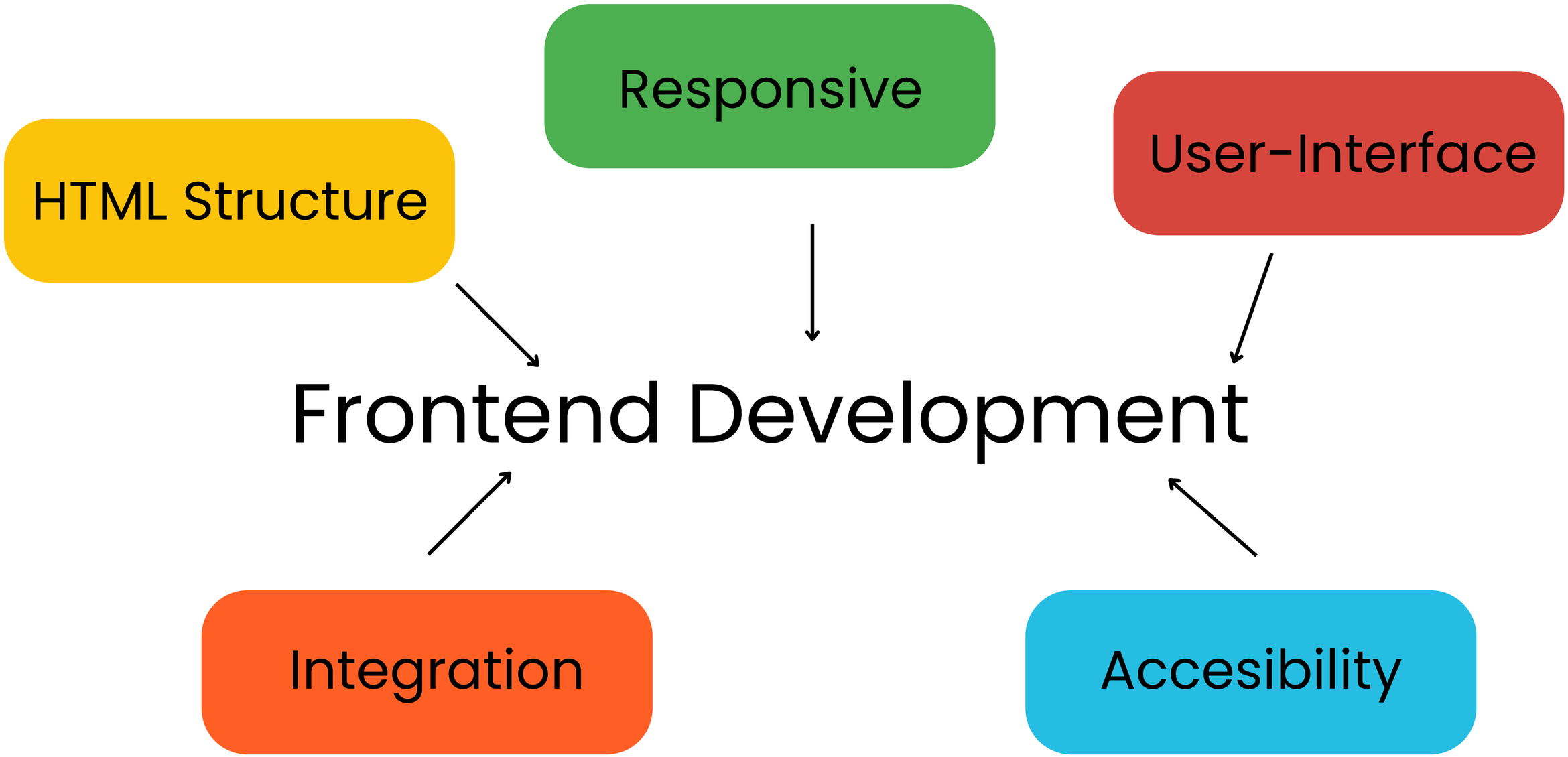


BACKEND DEVELOPER



FRONTEND DEVELOPER

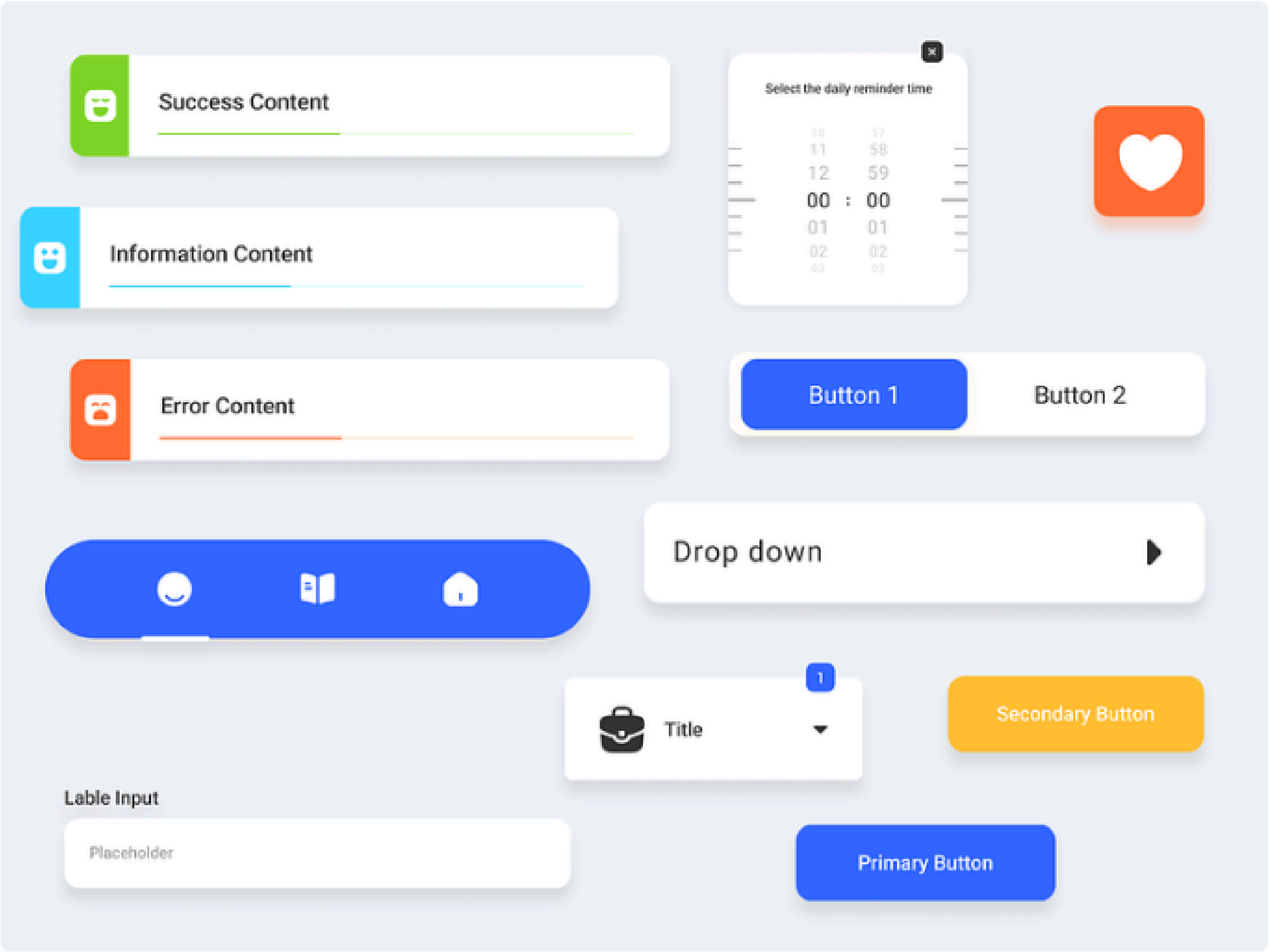
Frontend Development


Responsive
HTML Structure
User-Interface
Frontend Development
Integration
Accesibility

Frontend Developers

Focuses on the development of user-visible website elements and functionalities.

It involves in making the web visual aspects functional.



Buttons,
Designs,
Images,
Navigations,
Illustrations,
Animations

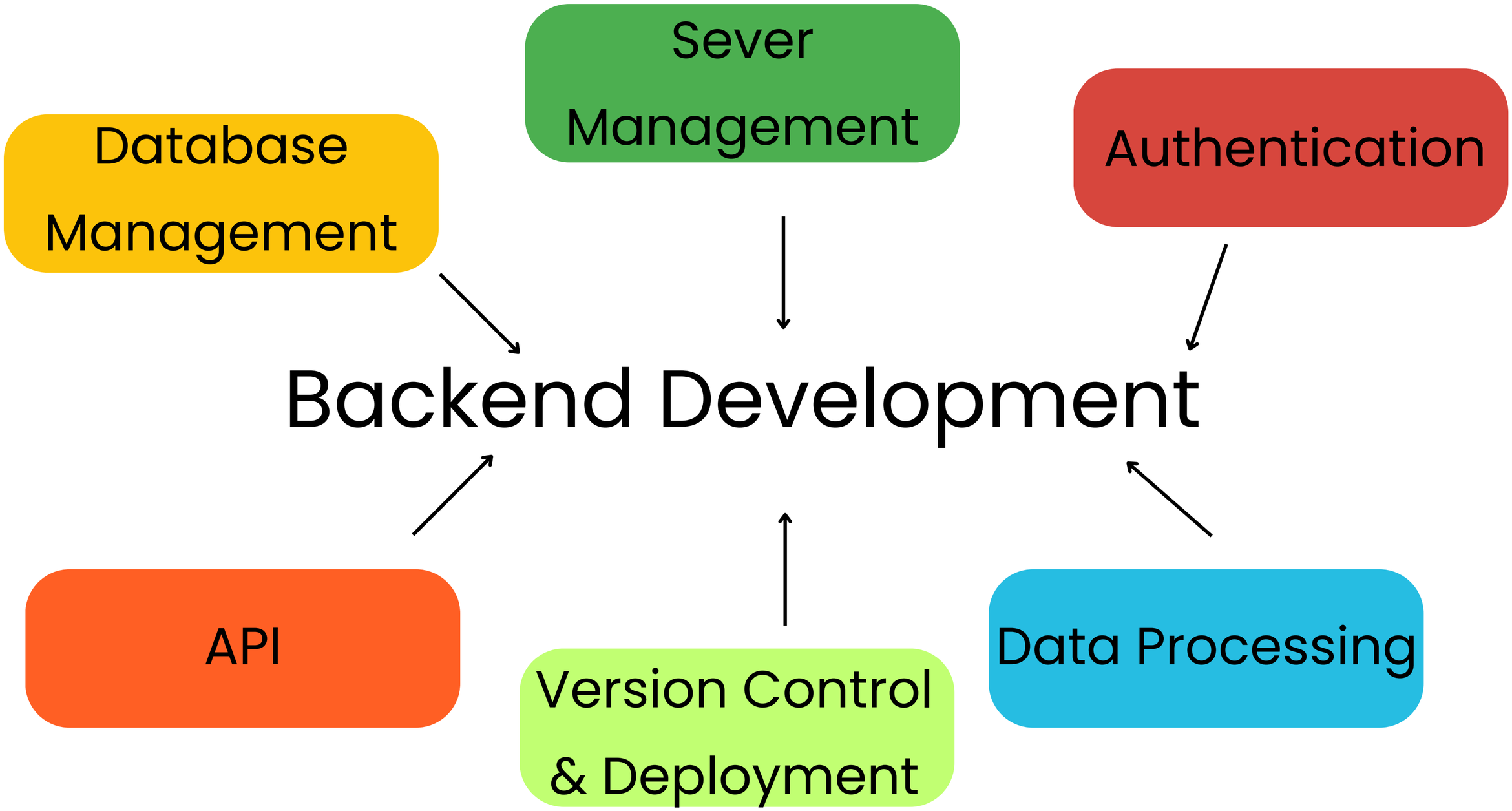
HTML
JS
CSS
WORKHARD
ANYWHERE

Frontend Development

Frontend Development makes use of its own libraries and frameworks.

REQUIRE·JS
npm
TS
REQUIRE·JS
REQUIRE·JS
BABEL
JS
JS
A
REQUIRE·JS

Backend Development


Sever Management
Database Management
Authentication
Backend Development
API
Version Control & Deployment
Data Processing

Backend Developers

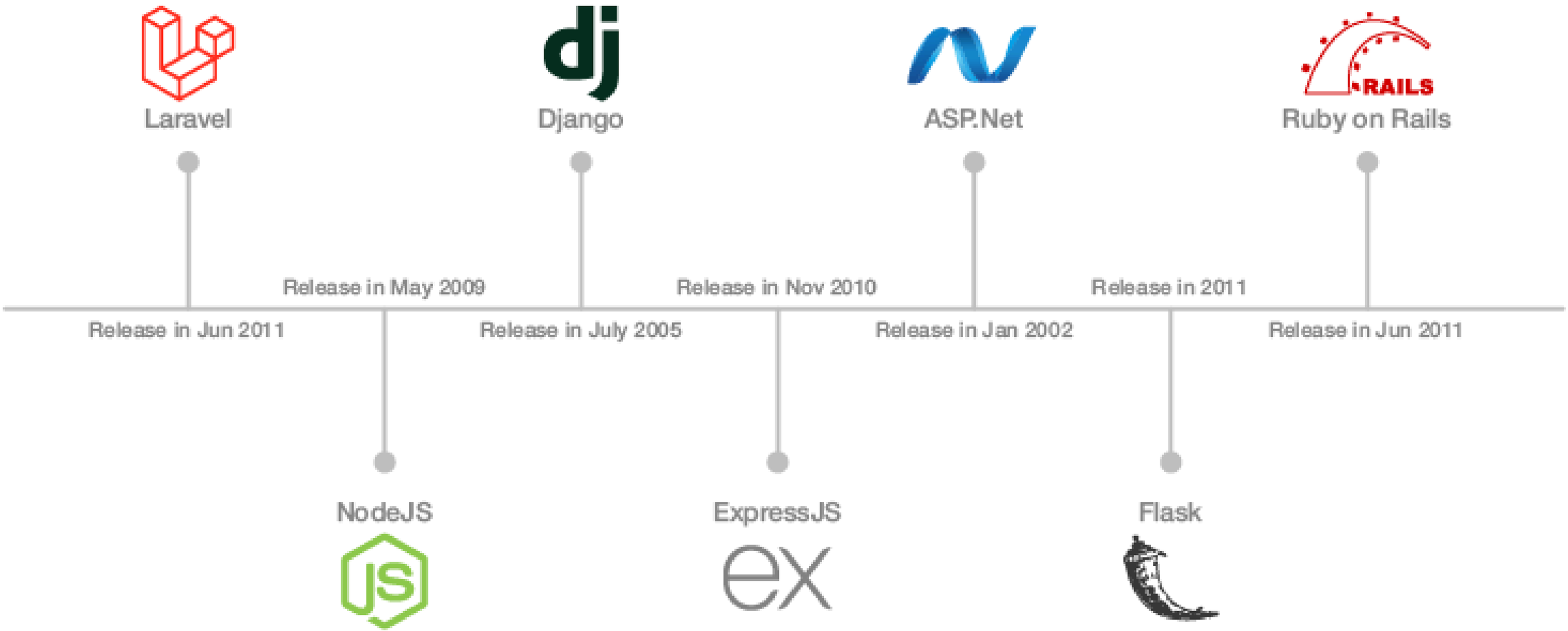
Focuses on the development and maintainance of server-side components of the application.

It involves in designing databases, develops API, manage data processing, security, server infrastructure.

Backend Developers

Backend Development makes use of its own frameworks.

Top Backend Frameworks 2024



Creating Your Portfolio

Speaker's Notes:

Hello, I'm Micah Mustaham

- Computer Science Graduate - ADZU 2018
- Studies MIT in ADZU 2023
- A part-time teacher at ADZU
- 2 years experience as System Developer

System Developer

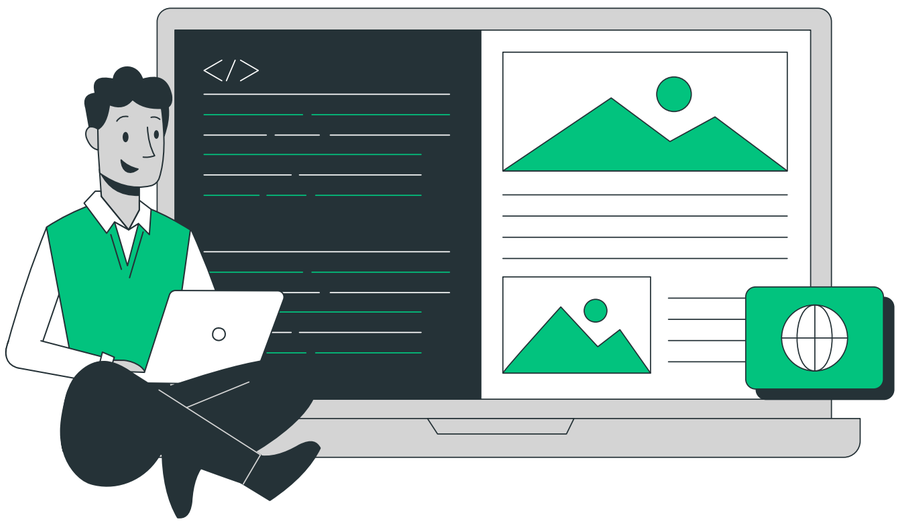
github.com/micah3252625

instagram.com/_micahchrles

https://www.facebook.com/micahmustaham1010/

Portfolio Website

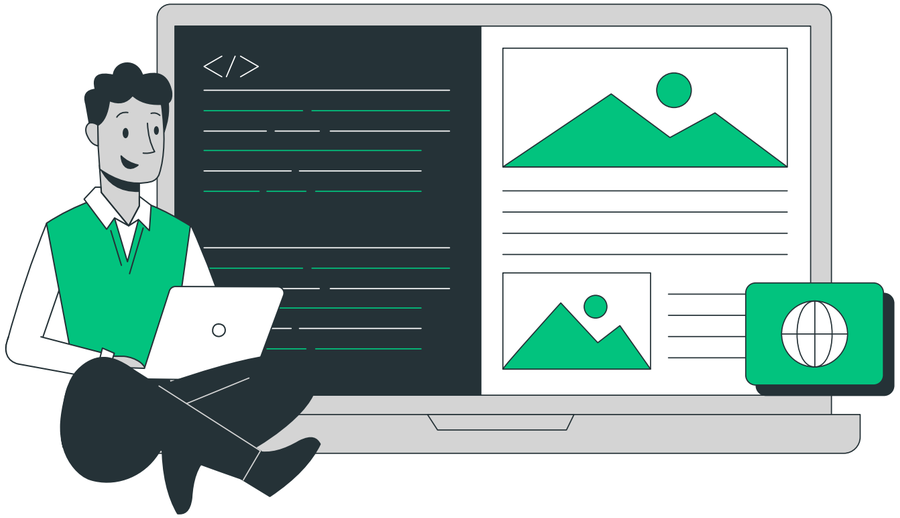
- platform used by individuals to showcased their work, projects, skills, and accomplishments
- It includes images, videos, descriptions, and links to further details about each project
- act as a virtual resume, allowing visitors to gauge the creator's skills and expertise



Content

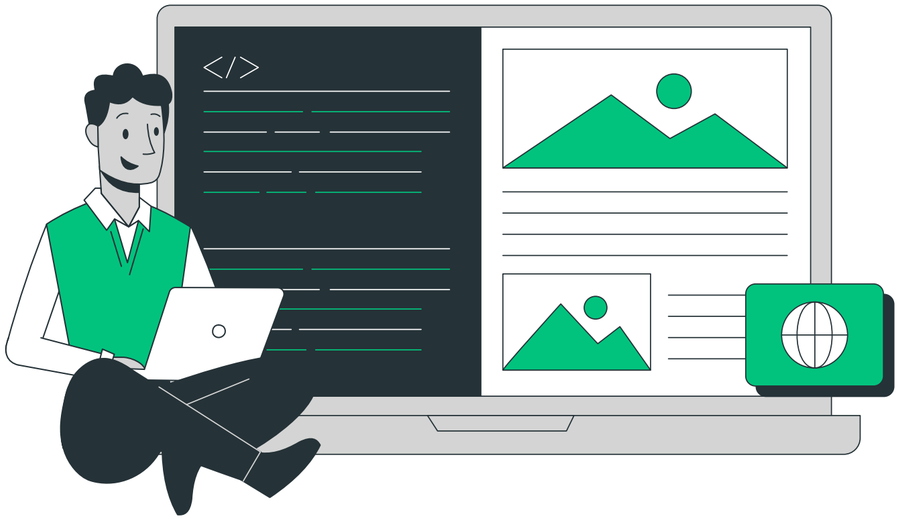
Typically include sections such as:

- Projects
- Services
- About Me/About Us
- Contact Information



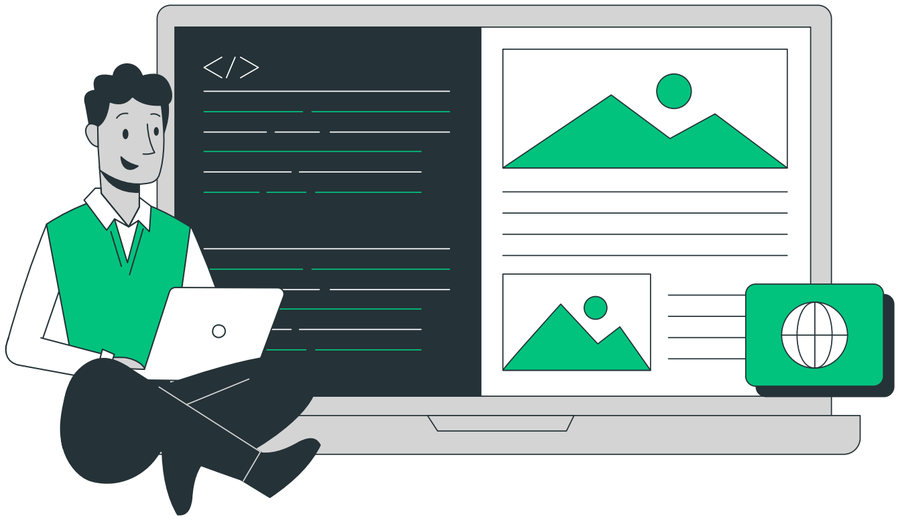
Design and Layout

- Visually appealing
- Layout should be intuitive and easy to navigate
- Responsive Design

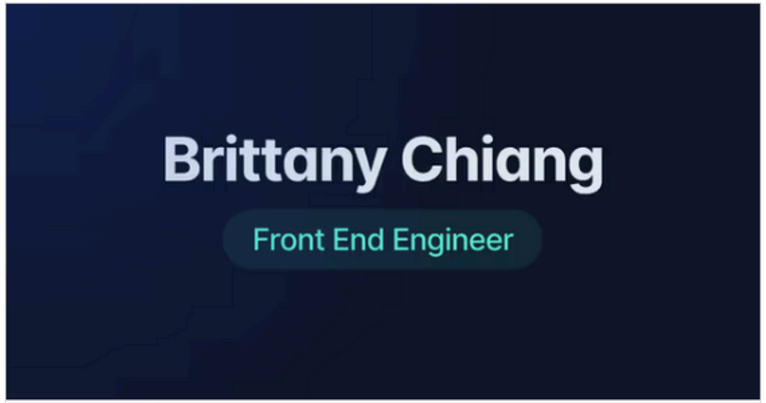


Technologies

- typically build using a combination of front-end and back-end techs
- Content Management System(CMS) like WordPress



Examples



Brittany Chiang

Brittany Chiang is a software engineer who builds accessible, inclusive products and digital experiences for the web.

Brittany Chiang

Frontend developer and UX Engineer.

Olaolu Olawuyi is a UX Engineer and Frontend developer highly skilled at design systems and...

Olaolu Olawuyi

Designer, Frontend Developer & Mentor

I design and code beautifully simple things, and I love what I do.

Matt Farley

DEMO...