JAN RUSSEL E. PEÑAFIEL BSIT – 2A

Developing a website involves a combination of **frontend**, **backend**, and **full-stack technologies**. Here's a breakdown of the programming languages and scripts commonly used, along with examples:

**Frontend Languages**

1. **HTML (HyperText Markup Language)**
   * **Purpose**: Structures the content of a webpage.
   * **Example**:

html

<h1>Welcome to My Website</h1>

<p>This is a sample paragraph.</p>

1. **CSS (Cascading Style Sheets)**
   * **Purpose**: Styles the content (e.g., colors, fonts, layout).
   * **Example**:

css

h1 {

color: blue;

font-size: 24px;

}

1. **JavaScript**
   * **Purpose**: Adds interactivity and dynamic functionality to webpages.
   * **Example**:

javascript

document.getElementById('button').addEventListener('click', () => {

alert('Button clicked!');

});

**Backend Languages**

1. **PHP (Hypertext Preprocessor)**
   * **Purpose**: Server-side scripting for dynamic content and database interaction.
   * **Example**:

php

<?php

echo "Hello, World!";

?>

1. **Python**
   * **Purpose**: Backend development using frameworks like Django or Flask.
   * **Example**:

python

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Welcome to the backend!"

1. **Ruby**
   * **Purpose**: Backend development using Ruby on Rails framework.
   * **Example**:

ruby

class WelcomeController < ApplicationController

def index

render plain: "Hello, World!"

end

end

1. **Java**
   * **Purpose**: Backend systems for enterprise websites using frameworks like Spring.
   * **Example**:

java

@RestController

public class HelloController {

@RequestMapping("/hello")

public String hello() {

return "Hello, World!";

}

}

1. **Node.js**
   * **Purpose**: JavaScript runtime for server-side scripting.
   * **Example**:

javascript

const http = require('http');

http.createServer((req, res) => {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello, World!');

}).listen(3000);

**Database Query Languages**

1. **SQL (Structured Query Language)**
   * **Purpose**: Managing and querying data in databases.
   * **Example**:

sql

SELECT \* FROM users WHERE username = 'admin';

**Full-Stack Languages**

1. **JavaScript (via frameworks)**
   * **Purpose**: Both frontend and backend development using frameworks like React.js (frontend) and Node.js (backend).
   * **Example**:
     + Frontend: React.js component:

javascript

function App() {

return <h1>Hello, React!</h1>;

}

* + - Backend: Node.js API:

javascript

const express = require('express');

const app = express();

app.get('/', (req, res) => {

res.send('Hello from the server!');

});

app.listen(3000);

1. **TypeScript**
   * **Purpose**: A typed superset of JavaScript for large-scale applications.
   * **Example**:

typescript

let message: string = "Hello, TypeScript!";

console.log(message);

**Other Technologies**

1. **Framework-Specific Languages**
   * **Example**: Razor (C# for ASP.NET applications).
2. **Scripting for Automation**
   * **Example**: Shell scripts for server-side automation.

Combining these languages creates dynamic, functional, and visually appealing websites!