ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Module 1: Introduction to Programming and Development Tools

INTRODUCTION TO PHP PROGRAMMING LANGUAGE:

- PHP stands for Hypertext Preprocessor, and it is a server-side scripting language that is used for web development.
- PHP is open-source, free, and widely used, which means that there are plenty of resources available online for learning and troubleshooting.
- PHP is often used in combination with other web technologies, such as HTML, CSS, JavaScript, and databases.
- PHP is used to create dynamic web pages, process forms, manage databases, and perform other server-side tasks
- Opening and closing PHP tags: <?php and ?>
- HTML and PHP code separation: PHP code can be embedded within HTML code.

```php
<?php
// This is a comment in PHP
echo "Hello, World!"; // Print "Hello, World!" to the screen
?>
```

ADVANTAGES OF USING PHP

- Easy to learn and use
- Fast and efficient
- Cross-platform compatibility
- Large and supportive community

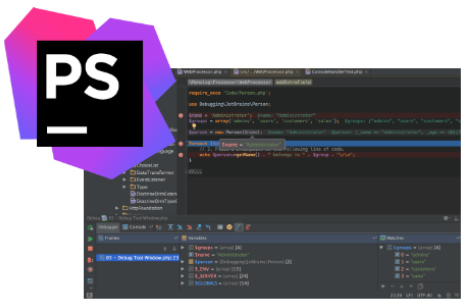
Examples of popular websites built with PHP (Facebook, Wikipedia, Yahoo, Flickr)

DEVELOPMENT TOOLS:

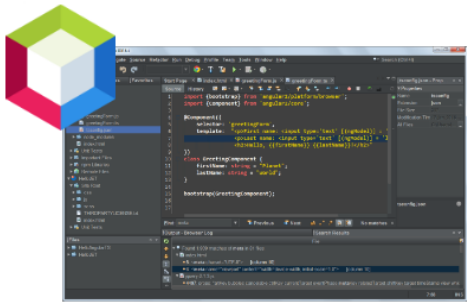
Integrated Development Environment (IDE) is a software application that provides a comprehensive environment for writing, testing, and debugging code.

Examples of PHP IDEs include:

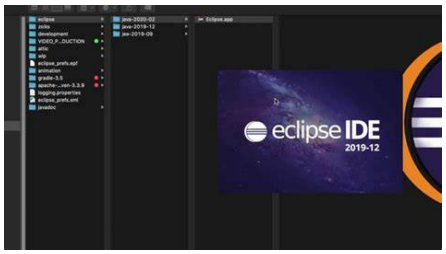
- **PHPStorm** - a commercial IDE with a focus on PHP development, developed by JetBrains.
    - *Download : https://www.jetbrains.com/phpstorm/*



- **NetBeans** - an open-source IDE with support for various programming languages including PHP, developed by Apache Software Foundation.
    - *Download:https://netbeans.apache.org/front/main/download/index.html*



- **Eclipse**- a popular open-source IDE that supports multiple languages including PHP, developed by the Eclipse Foundation.



BENEFITS OF USING AN IDE

- Syntax highlighting
- Code completion
- Debugging tools
- Integrated version control

BASIC SYNTAX AND STRUCTURE OF PHP CODE

Comments: single-line and multi-line comments are supported in PHP. Comments

Inline Comments

```php
<?php
    echo “test”; //this is inline comment
?>
```

Block Comments

```php
<?php
    /* This is a block comment,
    it can take several lines.
    */
?>
```

Variables: variables are used to store values and can be declared using the $ sign.

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

```
<?php
$x = 5;
$y = "John";

echo $x;
echo "<br>";
echo $y;
?>
```

Data types:

- Base Type
  - null type -The null type is PHP's unit type, i.e. it has only one value: null.

```
<?php
$var = NULL;
?>
```

  - Scalar types:
    - **bool type** - The bool type only has two values, and is used to express a truth value. It can be either true or false.
    - **int type** - An int is a number of the set $\mathbb{Z}$ = {..., -2, -1, 0, 1, 2, ...}.
    - **float type** -Floating point numbers (also known as "floats", "doubles", or "real numbers") can be specified using any of the following syntaxes:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
$d = 1_234.567; // as of PHP 7.4.0
?>
```

    - **string type** -A string is a series of characters, where a character is the same as a byte.
  - **array type** -A map is a type that associates values to keys. This type is optimized for several different uses; it can be treated as an array, list (vector), hash table (an implementation of a map), dictionary, collection, stack, queue, and probably more.

```
<?php
$array = array("foo", "bar", "hello", "world");
var_dump($array);
?>
```

  - **object type** - An array converts to an object with properties named by keys and corresponding values.

```
<?php
class foo
{
    function do_foo()
    {
        echo "Doing foo.";
    }
}

$bar = new foo;
$bar->do_foo();
?>
```

  - resource type- A resource is a special variable, holding a reference to an external resource. Resources are created and used by special functions

```

```

  - **Value types**- Value types are those which not only check the type of a value but also the value itself. PHP has support for two value types: false as of PHP 8.0.0, and true as of PHP 8.2.0.
    - false
    - True
  - **User-defined types** (generally referred to as class-types)
    - Interfaces
    - Classes
    - Enumerations
  - **callable type**

Operators: arithmetic(`+`,`-`, `/`, `%`, etc.), assignment(`=`,`+=`, `-=`),incerment(`++`, `--`), comparison(`==`, `!=`, `<`, `>`,etc.), and logical operators(`&&`,`||`) are supported in PHP.

Control structures: PHP supports if/else statements, loops, and switch statements.
Functions: PHP has many built-in functions and allows for the creation of custom functions.

- Setting up a development environment

To begin developing in PHP, you need two things: a web server and a text editor/IDE. There are several options available, including XAMPP(Most popular), WAMP(For Windows), MAMP(For Mac). Once installed properly run Apache and MySQL servers should be running under XAMPP control panel. After installing an IDE(text editor/Code Editor), configure it by specifying make sure your editor has syntax highlighting support for PHP files.

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Module 2: Programming Fundamentals

DATA TYPES AND VARIABLES IN PHP

Data types specify the type of data that a variable can store. PHP supports the following data types:

- Integer
  - Whole numbers, positive or negative (-2, 5, 50, -100)
- Float
  - Numbers with decimal points or scientific notation (4.5, 0.0002, 2.5e2)
- - String
  - Series of characters enclosed in quotation marks ("Hello World", '123', "true")
- - Boolean
  - True or False (true, false)

Variables are containers that store values. In PHP, variable names must start with a dollar sign ($). The basic syntax of a variable declaration in PHP is:

$variable_name = value;

Example:

```
$num1 = 5; // Integer variable
$num2 = 4.2; // Float variable
$name = "John"; // String variable
$is_valid = true; // Boolean variable
```

BASIC CONTROL STRUCTURES: CONDITIONALS AND LOOPS

Conditionals allow for decision-making and logic in programming. They are used to execute code based on certain conditions. PHP provides three types of conditionals:
If-else - Executes code if a condition is true, else executes another block of code.

Example :

```
$num = 10;
if ($num > 5) {
    echo "Number is greater than 5.";
} else {
    echo "Number is less or equal to 5.";
}
```

Switch - Allows code execution based on multiple possible values of a variable.

```
Switch:

$num = 2;
switch ($num) {
    case 1:
        echo "Number is one.";
        break;
    case 2:
        echo "Number is two.";
        break;
    default:
        echo "Number is not one or two.";
}
```

Ternary- Executes code based on the result of a comparison.

Loops are used to execute a block of code repeatedly. PHP provides four loop structures:

For - Executes a block of code a specified number of times.
While - Executes a block of code while a condition is true.
Do-while - Executes a block of code at least once, then repeatedly while a condition is true.
Foreach - Executes a block of code for each item in an array.

3. Functions and Procedures

Functions and procedures are blocks of code that can be reused throughout the program. They allow for modular and organized programming. In PHP, functions are declared using the keyword function, followed by the function name and the parameters enclosed in parentheses. Procedures are functions without a return value.

Example:

```
function calculate_sum($num1, $num2) {
    $sum = $num1 + $num2;
    return $sum;
}
$num1 = 5;
$num2 = 8;
$result = calculate_sum($num1, $num2);
echo "The sum of $num1 and $num2 is $result.";
```

4. Arrays and Lists

Arrays and lists are used to store a collection of similar data types. In PHP, arrays can be indexed or associative. Indexed arrays use numeric keys and start at 0, while associative arrays use string keys.

Example:

```
$my_array = array(2, 4, 6, 8);
echo $my_array[1]; // Outputs 4

$my_assoc_array = array("name" => "John", "age" => 25);
echo $my_assoc_array["name"]; // Outputs John
```

5. PHP Form and Superglobals

The HTML form is used to collect user input. In PHP, form data is accessed using superglobal variables. Superglobals are global variables that are predefined in PHP and can be accessed from anywhere in the code.

Example:

```
<form method="post" action="process_form.php">
    Name: <input type="text" name="name"><br>
    Email: <input type="email" name="email"><br>
    <input type="submit" value="Submit">
</form>

In process_form.php:

if (isset($_POST["name"]) && isset($_POST["age"])) {
    $name =$_POST["name"];
```

```
    $age =$_POST["age"];
            echo “Hi $name you are $age” ;
}
```

or The $_SERVER superglobal contains information about the server environment and the current request. Some of the information stored in this superglobal includes the current script file name, the IP address of the client, and the request method used to access the current script.

Sample syntax:

```php
echo $_SERVER['REQUEST_METHOD']; // Prints the request method (e.g. GET, POST)
echo $_SERVER['REMOTE_ADDR']; // Prints the IP address of the client
echo $_SERVER['PHP_SELF']; // Prints the current script file name
```

$_GET

The $_GET superglobal contains any data that was sent to the script through a GET request. In other words, any data that was included in the URL after a question mark (?).

Sample syntax:

Assuming the URL is http://example.com/?name=John&age=30:

```php
echo $_GET['name']; // Prints 'John'
echo $_GET['age']; // Prints '30'
```

$_POST

The $_POST superglobal contains any data that was sent to the script through a POST request. This data is not visible in the URL, as it is sent through the request headers.

Sample syntax:

Assuming a form with an input field named 'username':

```php
echo $_POST['username']; // Prints the value of the 'username' input field
```

$_FILES

The $_FILES superglobal contains any uploaded files that were sent to the script through a POST request. This superglobal is used when handling file uploads.

Sample syntax:

Assuming a form with an input field named 'file':

```php
$filename = $_FILES['file']['name']; // Retrieves the name of the uploaded file
$filesize = $_FILES['file']['size']; // Retrieves the size of the uploaded file
$tmpname = $_FILES['file']['tmp_name']; // Retrieves the temporary location of the uploaded file
```

Key Elements of `$_FILES`

When a file is uploaded, `$_FILES` will have the following components:

- **`$_FILES['input_name']['name']`: The original name of the uploaded file.**
- **`$_FILES['input_name']['tmp_name']`: The temporary file path on the server.**
- **`$_FILES['input_name']['type']`: The MIME type of the uploaded file (e.g., `image/jpeg`).**
- **`$_FILES['input_name']['size']`: The size of the uploaded file in bytes.**
- **`$_FILES['input_name']['error']`: An error code indicating the upload status.**

$_COOKIE

The $_COOKIE superglobal contains any cookies that were sent to the script by the client browser.

Sample syntax:

Assuming a cookie named 'username':

```php
echo $_COOKIE['username']; // Prints the value of the 'username' cookie
```

$_SESSION

The $_SESSION superglobal contains any session variables that have been set. Session variables are used to store information between page loads.

```php
session_start(); // Initializes the session
$_SESSION['username'] = 'John'; // Sets the 'username' session variable to 'John'
echo $_SESSION['username']; // Prints 'John'
```

$_REQUEST
The $_REQUEST superglobal contains the contents of $_GET, $_POST, and $_COOKIE.

Sample syntax:

Assuming a form with an input field named 'username':

```php
echo $_REQUEST['username']; // Prints the value of the 'username'
 input field (if sent through a GET or POST request) or the value of the 'username'
 cookie (if set)
```

$_ENV
The $_ENV superglobal contains any environment variables that have been set on the server.

Sample syntax:

Assuming an environment variable named 'DB

```php
echo $_ENV['DB_HOST']; // Prints the value of the 'DB_HOST' environment variable
```

$GLOBALS

The $GLOBALS superglobal contains all global variables that have been defined in the script. It can be used to access variables from within a function or other local scope.

ITEC 60 - Integrated Programming and Technologies 1

Prepared by : Benedict G Bautista

Sample syntax:

```
$var = 'Hello, world!';
function printVar() {
    echo $GLOBALS['var']; // Prints 'Hello, world!'
}
printVar();
```

Built in Function

STRING FUNCTION

strlen()

The strlen() function is used to find the length of a string.
Syntax: strlen(string)

Example:

```
$string = "Hello World!";
echo strlen($string); // Output: 12
```

str_replace()

The str_replace() function is used to replace a string with another string.
Syntax: str_replace(find, replace, string)

Example:

```
$string = "Hello World!";
echo str_replace("World", "Universe", $string); // Output: Hello Universe!
```

Trim

Strips whitespace from the beginning and end of a string.

echo trim(" Hello world! "); // Outputs: Hello world!

6. **strtolower() - Converts a string to lowercase.**
```php
echo strtolower("Hello WORLD!"); // Outputs: hello world!
```

7. **strtoupper() - Converts a string to uppercase.**
```php
echo strtoupper("Hello world!"); // Outputs: HELLO WORLD!
ARRAY FUNCTION
count()

The count() function is used to count the number of elements in an array.
Syntax: count(array, mode)

Example:

```
$array = array("red", "green", "blue");
echo count($array); // Output: 3
```

array_push()

The array_push() function is used to add one or more elements to the end of an array.
Syntax: array_push(array, value1, value2, ...)

Example:

```
$array = array("red", "green", "blue");
array_push($array, "yellow", "orange");
print_r($array); // Output: Array ( [0] => red [1] => green [2] =>
 blue [3] => yellow [4] => orange )
```

array_pop()

The array_pop() function is used to remove the last element from an array.
Syntax: array_pop(array)

Example:

```
$array = array("red", "green", "blue");
array_pop($array);
print_r($array); // Output: Array ( [0] => red [1] => green )
```

DATE FUNCTION

date()

The date() function is used to format a date and time.
Syntax: date(format, timestamp)

Example:

```
echo date("Y-m-d H:i:s"); // Output: 2023-03-11 14:05:00
```

strtotime()

The strtotime() function is used to convert a string into a Unix timestamp.
Syntax: strtotime(time, now)

Example:

```
echo strtotime("now"); // Output: 1647038700
```

date_diff()

The date_diff() function is used to calculate the difference between two dates.
Syntax: date_diff(datetime1, datetime2)

Example:

```
$date1 = date_create("2022-01-01");
$date2 = date_create("2023-03-11");
$diff = date_diff($date1, $date2);
echo $diff->format("%a days"); // Output: 434 days
```

Math Functions

abs()

The abs() function is used to return the absolute value of a number.

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Syntax: abs(number)

Example:

```
echo abs(-4); // Output: 4
```

pow()

The pow() function is used to raise a number to a power.
Syntax: pow(base, exponent)

Example:

```
echo pow(2, 3); // Output: 8
```

rand()

The rand() function is used to generate a random number.
Syntax: rand(min, max)
Example:

```
echo rand(1, 10); // Output: a random number between 1 and 10
```

DATABASE FUNCTIONS
mysqli_connect()

The mysqli_connect() function is used to establish a connection with a MySQL database.
Syntax: mysqli_connect(servername, username, password, dbname)

Example:

```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "database";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
```

mysqli_query()

The mysqli_query() function is used to execute a SQL query.
Syntax: mysqli_query(connection, query)

Example:

```
$sql = "SELECT * FROM users";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"] . " - Name: " . $row["name"] . "<br>";
    }
} else {
    echo "0 results";
}
```

mysqli_fetch_assoc()
The mysqli_fetch_assoc() function is used to fetch a result row as an associative array from a MySQL query. The function takes a result object returned by mysqli_query().

Example:

```
$sql = "SELECT * FROM users WHERE id = 1";
$result = mysqli_query($link, $sql);
$row = mysqli_fetch_assoc($result);
echo "ID: " . $row["id"] . " - Name: " . $row["name"];
```

mysqli_num_rows()
The mysqli_num_rows() function is used to get the number of rows in a result set returned by a MySQL query. The function takes a result object returned by mysqli_query().

Example:

```
$sql = "SELECT * FROM users";
$result = mysqli_query($link, $sql);
$num_rows = mysqli_num_rows($result);
echo "Number of rows: " . $num_rows;
```

mysqli_close()

The mysqli_close() function is used to close a MySQL database connection.
Syntax: mysqli_close(connection)

Example:

```
mysqli_close($conn);
```

VARIABLE HANDLING
isset()
The isset() function is used to check if a variable is set and not null.
Syntax: isset(variable)

Example:

```
$name = "John";
if (isset($name)) {
  echo "The variable is set.";
} else {
  echo "The variable is not set.";
}
```

empty()
The empty() function is used to check if a variable is empty.
Syntax: empty(variable)

Example:

var_dump()
The var_dump() function is used to display the type and value of a variable.

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Syntax: var_dump(variable)

```php
$name = "";
if (empty($name)) {
  echo "The variable is empty.";
} else {
  echo "The variable is not empty.";
}
```

Example:

```php
$name = "John";
var_dump($name); // Output: string(4) "John"
```

settype()
The settype() function is used to set the type of a variable.
Syntax: settype(variable, type)

Example:

```php
$age = "30";
settype($age, "integer");
echo gettype($age); // Output: integer
```

intval()
The intval() function is used to convert a variable to an integer.
Syntax: intval(variable)

Example:

```php
$age = "30";
echo intval($age); // Output: 30
```

trim()
The trim() function is used to remove whitespace or other characters from the beginning and end of a string.
Syntax: trim(string, characters)

Example:

```php
$message = "   Hello World!   ";
echo trim($message); // Output: Hello World!
```

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Module 3: Web Development Basics

HTML (Hypertext Markup Language): HTML is used for creating the structure and content of web pages. It provides a set of tags that are used to define headings, paragraphs, images, links, and other content.

CSS (Cascading Style Sheets): CSS is used for styling the HTML content. It provides a set of rules that are used to define the color, font, size, layout, and other visual aspects of the web page.

Bootstrap : Bootstrap is a popular CSS framework that provides a responsive grid system, pre-designed UI components, and customizable CSS styles

Materialize :Materialize is a front-end CSS framework based on Google's Material Design language. It provides a set of CSS and JavaScript components for building responsive web pages.

Foundation : Foundation is a responsive CSS framework that provides a set of customizable CSS styles and pre-designed UI components.

Semantic UI :Semantic UI is a CSS framework that uses human-friendly HTML to create responsive web pages. It provides a set of customizable CSS styles and pre-designed UI components.

Tailwind CSS :Tailwind CSS is a utility-first CSS framework that provides a set of pre-designed CSS classes for building responsive web pages.

Bulma: is a modern CSS framework that is designed to be flexible and easy to use. It is a lightweight alternative to larger CSS frameworks like Bootstrap, and it provides a simple set of customizable styles and components for building responsive web pages.

JavaScript : JavaScript is the most widely used client-side scripting language. It is a lightweight programming language that can be embedded into HTML pages to create dynamic effects and interactive user interfaces.

React.js: Developed and maintained by Facebook, used for building user interfaces with a virtual DOM, and allows for reusable UI components.

AngularJS: Developed and maintained by Google, used for building dynamic web applications with features like two-way data binding and dependency injection.

Vue.js: A progressive JavaScript framework for building user interfaces that is lightweight and easy to learn, providing reactive data binding and component-based architecture.

Node.js: A server-side JavaScript runtime built on top of Google's V8 JavaScript engine that allows for scalable network applications with non-blocking I/O.

Express.js: A popular web application framework built on top of Node.js that provides middleware support, routing, and templating engines.

Ember.js: A JavaScript framework for building ambitious web applications that includes features like two-way data binding, routing, and component-based architecture.

Meteor.js: A full-stack JavaScript framework for building real-time web applications with reactive data binding, server-side rendering, and client-side caching.

Backbone.js: A lightweight JavaScript framework that provides a minimal set of features for building client-side applications with models, views, and collections for organizing code.

Server-side scripting languages: Server-side scripting languages are used to create web applications that run on a server. These languages include PHP, Python, and Ruby on Rails, among others.

PHP: Widely used open-source scripting language for web development, can be embedded into HTML, easy to learn, large community of developers.

Python: General-purpose programming language used for web development, known for simplicity, readability, versatility.

Ruby: Dynamic, object-oriented programming language used for web development, clean syntax, emphasizes productivity and simplicity.

JavaScript: Primarily used for client-side scripting in web development, but can also be used on the server-side with Node.js.

Java: Popular programming language used for a wide range of applications, including web development, known for scalability, security, cross-platform compatibility.

C#: Modern programming language used for building Windows desktop applications, web applications on the Microsoft .NET platform, game development, and mobile development.

Perl: General-purpose scripting language used for web development, system administration, and network programming, known for text processing capabilities and ability to handle complex tasks.

Database technology tools are essential in managing, organizing, and analyzing large amounts of data efficiently.

Oracle Database: Oracle is a relational database management system (RDBMS) used for storing and managing structured data. It is known for its scalability, security, and high performance.

Microsoft SQL Server: SQL Server is another RDBMS that runs on Windows-based operating systems. It is commonly used in enterprise environments for managing large amounts of data and providing business intelligence solutions.

MySQL: MySQL is an open-source RDBMS widely used in web applications due to its ease of use, low cost, and scalability.

MongoDB: MongoDB is a NoSQL database that stores data in a JSON-like format. It is known for its ability to handle unstructured data and scalability.

PostgreSQL: PostgreSQL is an open-source RDBMS known for its robustness, extensibility, and SQL compliance. It is commonly used in enterprise environments for mission-critical applications.

Cassandra: Cassandra is a distributed NoSQL database known for its high scalability and fault tolerance. It is commonly used for storing large amounts of data across multiple servers.

Redis: Redis is an in-memory data structure store used for caching, real-time data processing, and messaging. It is commonly used in web applications and for storing session data.

HTML, or Hypertext Markup Language, is a markup language used to create web pages. It consists of a set of tags and attributes that define the structure and content of a web page.

Headings: HTML provides six levels of headings, from <h1> to <h6>, which can be used to create headings of different sizes and levels of importance. The syntax for headings is as follows:

```
<h1>This is a level 1 heading</h1>
<h2>This is a level 2 heading</h2>
<h3>This is a level 3 heading</h3>
<h4>This is a level 4 heading</h4>
<h5>This is a level 5 heading</h5>
<h6>This is a level 6 heading</h6>
```

Paragraphs: The <p> tag is used to create paragraphs of text. The syntax is as follows:

```
<p>This is a paragraph of text.</p>
```

Links: The <a> tag is used to create links to other web pages or resources. The syntax is as follows:

```
<a href="https://www.example.com">Click here to visit Example.com</a>
```

Images: The <img> tag is used to insert images into a web page. The syntax is as follows:

```
<img src="image.jpg" alt="Description of the image">
```

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Lists: HTML provides two types of lists: ordered lists (<ol>) and unordered lists (<ul>). The syntax for creating lists is as follows:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>

<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Forms: HTML provides a number of elements for creating forms, which are used to collect user input. The <form> tag is used to create a form, and different types of input elements can be used within the form, such as text boxes, radio buttons, checkboxes, and dropdown menus. Here is an example:

```
<form action="submit-form.php" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="message">Message:</label>
  <textarea id="message" name="message" rows="5" required></textarea>

  <input type="submit" value="Submit">
</form>
```

Tables: HTML provides the <table> tag for creating tables on a web page. Tables can be used to display data in an organized manner. Here is an example:

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>Gender</th>
  </tr>
  <tr>
    <td>John</td>
    <td>30</td>
    <td>Male</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>25</td>
    <td>Female</td>
  </tr>
</table>
```

Divs and Spans: The and tags are used to group elements together for styling purposes. is a block-level element, while is an inline element. Here is an example:

```
<div class="container">
  <h1>Welcome to my website</h1>
  <p>Here you will find information about my products and services.</p>
  <span class="highlight">Contact us</span> for more information.
</div>
```

Audio and Video: HTML provides <audio> and <video> tags for embedding audio and video files in a web page. Here is an example:

```
<audio controls>
  <source src="music.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

<video controls>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video element.
</video>
```

Meta Tags: HTML provides a set of meta tags that can be used to provide additional information about a web page, such as its title, description, and keywords. Here is an example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="description" content="This is a sample web page.">
    <meta name="keywords" content="HTML, CSS, JavaScript">
    <meta name="author" content="John Doe">
    <title>Sample Web Page</title>
  </head>
  <body>
    ...
  </body>
</html>
```

Comments: HTML allows you to add comments to your code using the <!-- --> syntax. Comments are not visible on the web page and are only intended for human readers. Here is an example:

```
<!-- This is a comment -->
```

Buttons: HTML provides the <button> tag for creating buttons on a web page. Here is an example:

```
<button type="button">Click me</button>
```

Horizontal Rule: HTML provides the <hr> tag for inserting a horizontal rule or line on a web page. Here is an example:

```
<hr>
```

Abbreviations: HTML provides the <abbr> tag for indicating an abbreviation or acronym. Here is an example:

```
<p>The <abbr title="World Health Organization">WHO</abbr> is a specialized agency of the United Nations.</p>
```

Subscripts and Superscripts: HTML provides the <sub> and <sup> tags for creating subscripts and superscripts, respectively. Here are some examples:

ITEC 60 - Integrated Programming and Technologies 1

Prepared by : Benedict G Bautista

```
<p>H<sub>2</sub>O</p>
<p>E = mc<sup>2</sup></p>
```

CSS

Responsive web design makes your web page look good on all devices.
Responsive web design uses only HTML and CSS.
Responsive web design is not a program or a JavaScript.

Viewport

- The viewport is the user's visible area of a web page.
- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

HTML5 introduced a method to let web designers take control over the viewport, through the **<meta>** tag.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Without the viewport meta tag

With the viewport meta tag

1. **Do NOT use large fixed width elements** - For example, if an image is displayed at a width wider than the viewport it can cause the viewport to scroll horizontally. Remember to adjust this content to fit within the width of the viewport.

2. **Do NOT let the content rely on a particular viewport width to render well** - Since screen dimensions and width in CSS pixels vary widely between devices, content should not rely on a particular viewport width to render well.

3. **Use CSS media queries to apply different styling for small and large screens** - Setting large absolute CSS widths for page elements will cause the element to be too wide for the viewport on a smaller device. Instead, consider using relative width values, such as width: 100%. Also, be careful of using large absolute positioning values. It may cause the element to fall outside the viewport on small devices.

Media Query

Media query is a CSS technique introduced in CSS3.
It uses the @media rule to include a block of CSS properties only if a certain condition is true.

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Always Design for Mobile First

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).

```
/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}

@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

Pseudo-classes selector

:**active** - targets the active state of an element (when it is being clicked)
Syntax: selector:active {...}

:**checked** - targets elements that are checked, such as checkboxes or radio buttons
Syntax: input:checked {...}

:**disabled** - targets disabled form elements
Syntax: input:disabled {...}

:**empty** - targets elements that have no children
Syntax: selector:empty {...}

:**enabled** - targets enabled form elements
Syntax: input:enabled {...}

:**focus** - targets the element that has focus
Syntax: selector:focus {...}

:**hover** - targets the element that is being hovered over
Syntax: selector:hover {...}

Function

rgb() and rgba(): These functions are used to specify colorsl in the red-green-blue (RGB) color space.

```
background-color: rgb(255, 0, 0); /* red */
color: rgba(0, 0, 255, 0.5); /* semi-transparent blue */
```

hsl() and hsla(): These functions are used to specify colors in the hue-saturation-lightness (HSL) color space.

```
background-color: hsl(120, 100%, 50%); /* green */
color: hsla(0, 100%, 50%, 0.5); /* semi-transparent red */
```

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

calc(): This function is used to perform arithmetic calculations in CSS. It can be used to calculate values for properties such as width, height, and margin.

```
width: calc(50% - 20px);
height: calc(100vh - 100px);
```

url(): This function is used to specify the location of an external resource such as an image or a font.

```
background-image: url("path/to/image.png");
font-family: url("path/to/font.ttf");
```

transform(): This function is used to apply transformations to HTML elements, such as scaling, rotating, and translating.

```
transform: translate(50px, 0); /* moves the element 50 pixels to the right */
```

linear-gradient(): This function is used to create a gradient background using two or more colors.

```
background-image: linear-gradient(red, yellow); /* creates a gradient from red to yellow */
```

box-shadow(): This function is used to add a shadow effect to an HTML element.

```
box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.5); /* adds a shadow effect to the element */
```

transition(): This function is used to create a smooth transition effect when a CSS property changes value.

```
transition: width 1s ease-in-out; /* creates a smooth transition effect for the width property */
```

Fallback Fonts

Serif

- "Times New Roman", Times, serif

- Georgia, serif

- Garamond, serif

Sans-serif

- Arial, Helvetica, sans-serif

- Tahoma, Verdana, sans-serif

- "Trebuchet MS", Helvetica, sans-serif

- Geneva, Verdana, sans-serif

Monospace

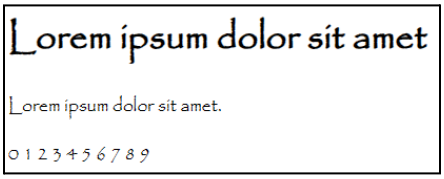
- "Courier New", Courier, monospace

Cursive

- "Brush Script MT", cursive

Fantasy

- Copperplate, Papyrus, fantasy



Units

There are two types of length units: absolute and relative.

Absolute Lengths
The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

ITEC 60 - Integrated Programming and Technologies 1
Prepared by : Benedict G Bautista

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Relative Lengths
Relative length units specify a length relative to another length property.
Relative length units scale better between different rendering medium.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element