

I DID NOT DEVELOP THE PHP
WE KNOW TODAY. DOZENS, IF
NOT HUNDREDS OF PEOPLE
DEVELOPED PHP. I WAS SIMPLY
THE FIRST DEVELOPER.

-RASMUS LERDORF-



Introduction to PHP

PHP හැඳින්වීම

PHP is a server-side scripting language that is widely used for web development.

PHP යනු වෙබ් අඩවි සංවර්ධනය සඳහා බහුලව කාවිතා වන සේවාලායක අගුරෙහි ක්‍රියාත්මක වන ස්ක්‍රීප්ට්‍රේත් භාෂාවකි.

PHP, originally known as **Personal Home Page**, was created by Rasmus Lerdorf in 1994.

අරමිනයේදී Personal Home Page මෙසේ හැඳින්වූ PHP , 1994 දී Rasmus Lerdorf විසින් තීර්මාණය කරන ලදී.

Initially, it consisted of Perl scripts designed to manage his personal website and track visits.

මුදලී, විය ඔහුගේ පුද්ගලික වෙබ් අඩවිය කළමනාකරණය කිරීම සහ පැමිණීම් තීර්ක්ෂණය කිරීම සඳහා තීර්මාණය කරන ලද භාෂාවෙන් මිය ස්ක්‍රීප්ට්‍රේත් වලින් සමන්විත විය.

Recognizing its potential beyond personal use, Lerdorf rewrote these tools in C language and released PHP/FI (Forms Interpreter) 1.0 in 1995.

පුද්ගලික භාවිතයෙන් ඔබිබට විෂි විනවය හැඳුනාගෙන, Lerdorf විසින් මෙම මෙවලම් C භාෂාවෙන් නැවත මියා 1995 දී PHP/FI (Forms Interpreter) 1.0 මෙසින් නිකුත් කළේය.

It provided basic tools for web form processing and database communication. විය වෙබ් ආකෘති සැකසීම සහ දත්ත සමුදා සමග සන්නිවේදනය සඳහා මූලික මෙවලම් සපයන ලදී.

Later versions

PHP 3.0 (1998)

Introduced a modular architecture and expanded database support.

මොඩුලර් තීර්මිතියක් සහිත සහ පුත්ල් දත්ත සමුදා සහායන් හඳුන්වා දෙන ලදී.

PHP 4.0 (2000)

Powered by the Zend Engine This version brought significant performance improvements and advanced features like sessions and output buffering.

Zend Engine මගින් බලගත්වන ලද මෙම අනුවාදය සැලකිය යුතු කාර්ය සාධන වැඩිදියුණු කිරීම් සහ සැකසීම සහ ප්‍රතිදාන වැනි උසස් විශේෂාංග ගෙන වලින් සමන්විත විය.

PHP 5.0 (2004)

Enhanced support for object-oriented programming (OOP) with the introduction of classes, interfaces, exceptions, and improved XML handling.

classes, interfaces, exceptions, සහ වැඩිදියුණු කළ XML හැකිරවීම හඳුන්වාදීම සමග වස්තු-නැමුරු ක්‍රමලේඛකරණය OOP සඳහා වැඩි දියුණු කළ සහාය.

PHP 7.0 (2015)

Delivered major performance enhancements with Zend Engine 3.0, adding scalar type declarations, return type declarations, and better error handling.

Zend Engine 3.0 සමඟින් ප්‍රධාන කාර්ය සාධන වැඩි දියුණු කිරීම් ලබා දී, අදින ආකාරයේ ප්‍රකාශ විකුත් කිරීම, return type ප්‍රකාශයන් සහ වඩා තොදු ලේඛ හැකිරවීමක් ගෙන දුන්නේය.

PHP 8.0 (2020)

Introduced the JIT compiler for further performance gains, along with new features like union types, named arguments, and match expressions.

union types, named arguments, සහ match ප්‍රකාශන වැනි නව විශේෂාංග සමගින් තවදුරටත් කාර්ය සාධනයක් සඳහා (න්ස්ට්) සම්පාදකය හඳුන්වා දෙන ලදී.

Static webpages

ස්ථිරික වෙබ් පිටු

Static webpages display fixed content. They are built using HTML and sometimes CSS and JavaScript.

ස්ථිරික වෙබ් පිටු ස්ථාවර අන්තර්ගතයන් පෙන්වයි. එවා HTML සහ සමඟර අවස්ථා වලදී CSS සහ JavaScript හාඩිතයෙන් ගොඩනග ඇත.

The content remains the same for every user and doesn't change unless the web developer manually updates the HTML code.

සම පරිශීලකයෙකුටම අන්තර්ගතය විකම ලෙස පවතින අතර වෙබ් සංවර්ධකය විසින් HTML කේතය අතින් යාවත්කාලීන කරන්නේ නම් මිස වෙනස් නොවේ.

Dynamic webpages

ගතික වෙබ් පිටු

Dynamic webpages display content that can change based on user interactions, preferences, or other factors.

ගතික වෙබ් පිටු පරිශීලක අන්තර්ග්‍රියා, කැමෙන්ත, හෝ වෙනත් සාකි මත පදනම්ව වෙනස් විය හැකි අන්තර්ගත සංදර්ජන කරයි.

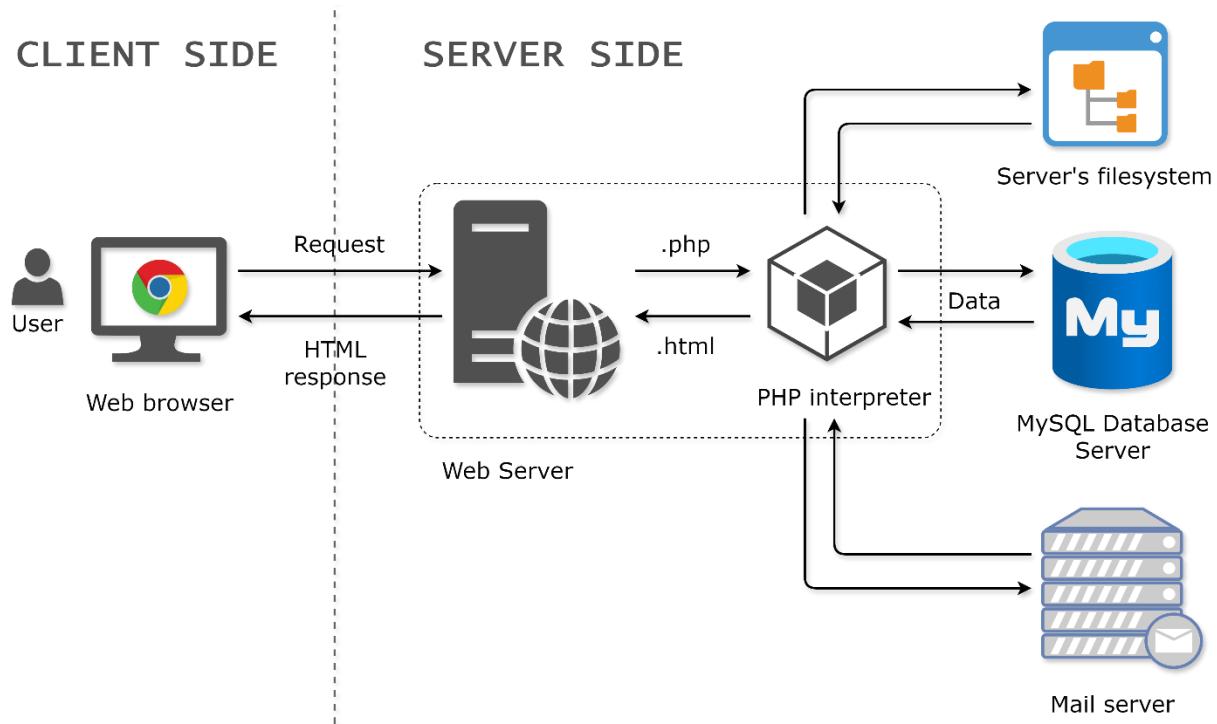
They are created using server-side scripting languages like PHP, Ruby, Python, or Node.js, and often interact with databases to retrieve and store data.

එවා PHP, Ruby, Python, හෝ Node.js වැනි සේවාදායක අගුයෙහි ක්‍රියාත්මක වන ස්ක්‍රීප්‍රේන් හාමා හාඩිතයෙන් තිරිමාත්‍ය කර ඇති අතර, දැන්ත ලබා ගැනීමට සහ ගබඩා කිරීමට බොහෝ විට දැන්ත සමුදායන් සමඟ අන්තර් ක්‍රියා කරයි.

| Feature | Static Webpages | Dynamic Webpages |
|--|--|---|
| Content අන්තර්ගතය | Fixed ස්ථිරිකය | Changes based on user interaction පරිශීලක අන්තර්ග්‍රියා මත පදනම් වී වෙනස් වෙයි |
| Generation නිර්මාණය | Pre-generated HTML පෙර-ජනනය කරන ලද HTML | Generated with server-side scripts සේවාදායක අගුයෙහි හාමා මගින් ජනනය කර ඇත |
| Complexity සංකීර්ණත්වය | Simple (HTML, CSS) සරලයි | Complex (server-side languages, databases) සංකීර්ණයි (සේවාදායක හාමා, දැන්ත සමුදායන්) |
| Interactivity අන්තර් ක්‍රියාකාර්ත්වය | Limited සීමා සහිතයි | High (forms, user accounts) ඉහළයි (පෝර්ම, පරිශීලක ගිණුම්) |
| Performance කාර්ය සාධනය | Fast වේගවත් | Slower (server processing required) මන්දගාලීය (සේවාදායක සැකසුම් අවශ්‍යයි) |
| Maintenance නඩත්තු කිරීම | Manual updates අතින් යාවත්කාලීන කරයි | Backend interface or automatic updates පසුඩ්‍රීම් අනුරූ මුහුණාත හෝ ස්වයංක්‍රීය යාවත්කාලීන කිරීම් |
| Examples | Wikipedia, portfolios  WIKIPEDIA The Free Encyclopedia | E-commerce, social media, Streaming-services    |

How PHP works

PHP ක්‍රියා කරන ආකාරය



Request Handling

ඉල්ලීම් හැසේරවීම

When a client (usually a web browser) requests a PHP page from a web server, the server identifies the file as a PHP script based on its .php extension.

සේවාලාභීයක (සාමාන්‍යයෙන් වෙබ් බූෂරුයක්) වෙබ් සේවාලායකයකින් PHP පිටුවක් ඉල්ලා සිටින විට, සේවාලායකය විනි .php දිගුව මත පදනම්ව ගොනුව PHP ස්ක්‍රීප්ට් විකක් ලෙස හඳුනා ගනී.

PHP Processing

PHP සැකසුම

The web server then sends the PHP file to the PHP interpreter, which is installed on the server. The PHP interpreter processes the PHP code within the file.

විවිධ වෙබ් සේවාලායකය මගින් PHP ගොනුව සේවාලායකයේ ස්ථාපනය කර ඇති PHP පරිවර්තකය වෙත යවයි. PHP පරිවර්තකය ගොනුව තුළ ඇති PHP කේතය සකසයි.

Database Interaction

දත්ත සමුදා අන්තර්ක්‍රියා

If the PHP script includes database operations, PHP communicates with the database (Ex. MySQL) to retrieve or manipulate data.

PHP කේතයේ දත්ත සමුදා මෙහෙයුම් ඇතුළත් වේ නම්, PHP විසින් දත්ත ලබා ගැනීමට හෝ හැසේරවීමට දත්ත සමුදාය (උදා: MySQL) සමග සන්නිවේදනය කරයි.

Dynamic Content Generation

ගතික අන්තර්ගත උත්පාදනය

PHP generates dynamic content based on the script's logic and any data retrieved from databases or other sources.

PHP ස්ක්‍රීප්ට් වල තර්ක සහ දත්ත සමුදායන් හෝ වෙනත් මූලාශ්‍ර වලින් ලබා ගන්නා විනෑම දත්තයක් මත පදනම්ව ගතික අන්තර්ගතයන් ජනනය කරයි.

HTML Output

HTML ප්‍රතිදානය

The PHP interpreter converts the processed PHP code into an HTML output, which is the language understood by web browsers. PHP පරිවර්තනය විසින්, සැකසු PHP කේතය HTML ප්‍රතිදානයක් බවට පරිවර්තනය කරයි, විය වෙඩි අතරක්සුව විසින් තේරෑම් ගන්නා හාඡාව වේ.

Server Response

සේවාදායක ප්‍රතිචාරය

The web server receives the generated HTML output from the PHP interpreter and sends it back to the client's browser.

වෙඩි සේවාදායකය PHP පරිවර්තනයෙන් ජනනය කරන ලද HTML ප්‍රතිදානය ලබාගෙන විය සේවාදායකයාගේ අතරක්සුව වෙත යවයි.

Client Display

සේවාදායී සංදුර්ශකය

The client's browser renders the received HTML output, displaying the final web page to the user.

සේවාදායකයාගේ අතරක්සුව ලැබූණු තම්බ ප්‍රතිදානය ලබා දෙන අතර අවසාන වෙඩි පිටුව පරිශීලකයාට පෙන්වයි.

Client-side processing and Server-side processing

සේවාදායී අගුරෝධි සැකසුම සහ සේවාදායක අගුරෝධි සැකසුම

Client-side

HTML and CSS codes are executed directly on the client's computer, utilizing the browser to parse and render the content.

HTML සහ CSS කේත අන්තර්ගතය විගුහ කිරීමට සහ විදුලුම් කිරීමට අතරක්සුව හාවිතා කරමින් සපුරාවම සේවාදායීයාගේ පරිගණකය මත ක්‍රියාත්මක වේ.

This process leverages the client's CPU and memory resources, causing the client's computer to consume power and generate heat as it processes and displays the webpage.

මෙම ක්‍රියාවලිය සේවාදායීයාගේ සකසනය සහ මතක සම්පත් ප්‍රයෝගනයට ගන්නා අතර, වෙඩි පිටුව සැකසීමේදී සහ පුද්ගලය කිරීමේදී සේවාදායීයාගේ පරිගණකයෙහි බලය පර්හෝජනය වීමට සහ තාපය ජනනය වීමට හේතු වේ.

Server-side

PHP scripts are executed on the server before any content is sent to the client.

සේවාදායකය වෙත කිසියම් අන්තර්ගතයක් යැවීමට පෙර PHP කේත සේවාදායක පරිගණකය මත ක්‍රියාත්මක වේ.

This means the server's resources (its CPU, memory, and power) are used during the PHP execution, causing the server to consume power and generate heat.

මෙයින් අදහස් කරන්නේ සේවාදායකයේ සම්පත් (විනි සකසනය, මතකය සහ බලය) PHP ක්‍රියාත්මක කිරීමේදී හාවිතා වන අතර විය සේවාදායකයෙහි බලය පර්හෝජනය වීමට සහ තාපය ජනනය වීමට හේතු වේ.

Use cases of PHP

PHP හි භාවිත

Content Management Systems (CMS)

අන්තර්ගත කළමනාකරණ පද්ධති

Powering platforms like WordPress, Joomla, and Drupal for website and blog management.

වෙබ් අඩවි සහ බිලොග් කළමනාකරණය සඳහා WordPress, Joomla සහ Drupal වැනි වේදිකා බල ගැන්වීම.

E-commerce Platforms

ඊ-වාණිජය වේදිකා

Used in Magento and WooCommerce for building online stores with product management and secure payment gateways.

තිපේපාදන කළමනාකරණය සහ ආරක්ෂණ ගෙවීම ද්වාර සමග සඛැලු වෙළඳසැල් ගොඩනැගීම සඳහා Magento සහ WooCommerce හි භාවිත වේ.

Frameworks

Frameworks like Laravel, Symfony, and CodeIgniter streamline web application development with reusable components and database management.

Laravel, Symfony, සහ CodeIgniter වැනි නැවත භාවිත කළ හැකි සංරචක සහ දත්ත සමුළු කළමනාකරණය සමග වෙබ් යෝදුම් සංවර්ධනය විධිමත් කරයි.

Social Media Platforms

සමාජ මාධ්‍ය වේදිකා

Utilized by Facebook and parts of Twitter and LinkedIn for backend services and API responses.

පසුපෙළ නොවා සහ API ප්‍රතිචාර සඳහා Facebook සහ Twitter සහ LinkedIn කොටස් මගින් භාවිත කෙරේ.

Online Forums

Powering forum software like phpBB and vBulletin for community interactions.

පහා අන්තර්ඩ්‍රොයිඩ සඳහා phpBB සහ vBulletin වැනි සංස්කීර්ණ මෘදුකාංග බල ගැන්වීම.

Web Services and APIs

වෙබ් නොවා සහ APIs

Developing RESTful APIs and web services that process requests and deliver dynamic content.

මුළුම් සකසන සහ ගතික අන්තර්ගතයන් බඟ දෙන RESTful API සහ වෙබ් නොවා සංවර්ධනය කිරීම.

Custom Web Applications

අකුරත් වෙබ් යොදුම්

Creating bespoke applications for specific business needs like CRMs and inventory management systems.

CRM සහ ඉන්වෙන්ටර් කළමනාකරණ පද්ධති වැනි විශේෂීත ව්‍යාපාරික අවශ්‍යතා සඳහා සුදුසු යොදුම් නිර්මාණය කිරීම.

Setting up a virtual server

අතරින සේවාදායකයක් සකසා ගැනීම

PHP runs on servers, which means we can't directly run PHP scripts on our local computer like we do with HTML or JavaScript.

PHP සේවාදායක මත ක්‍රියා කරයි, විනම් අපට HTML හෝ JavaScript මෙන් අපගේ සේවානීය පරිගණකයේ සංස්කීරුවම PHP ශේෂ බාධනය කළ නොහැක.

To execute PHP code, we need a server environment that can process PHP scripts and handle HTTP requests.

PHP සේෂය ක්‍රියාත්මක කිරීමට, අපට PHP ස්ක්‍රීප්ට් සැකසීමට සහ HTTP ඉල්ලීම් හැසිරවීමට හැකි සේවාදායක පරිසරයක් අවශ්‍ය වේ.

Since setting up an actual remote server can be costly and impractical for development and testing purposes, we temporarily turn our local computer into a virtual server.

යෙදින දිරීස්ථ සේවාදායකයක් පිහිටුවීම, සංවර්ධන සහ පරීක්ෂණ අරමුණු සඳහා මිල අධික සහ ප්‍රායෝගික නොවන බැවෙන්, අප අපගේ සේවානීය පරිගණකය තාවකාලිකව අතරින සේවාදායකයක් බවට පත් කරනු ලැබේ.

By creating a virtual host on our own computer, we simulate a web server environment, allowing us to send and receive HTTP requests using a web browser.

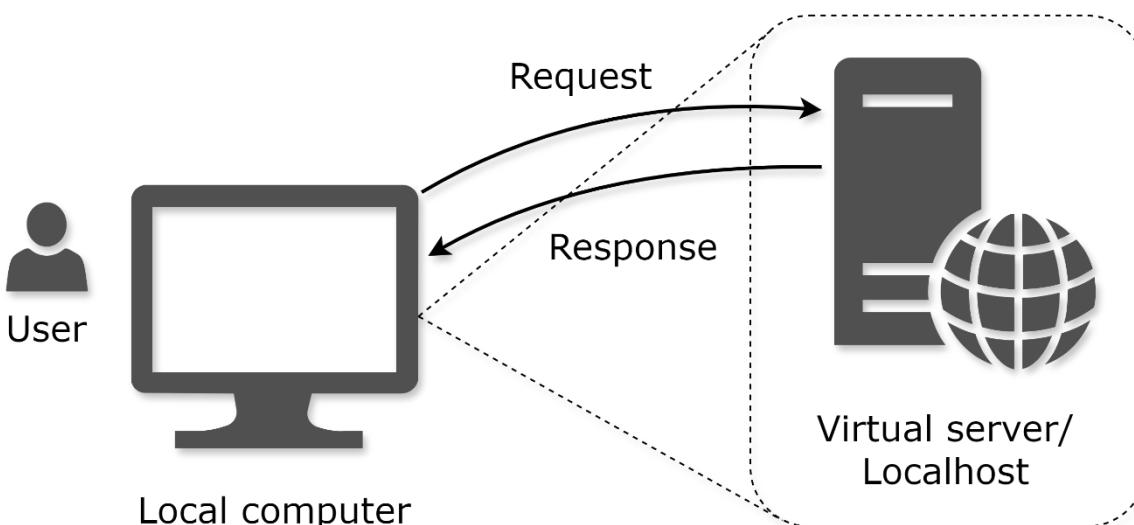
අපගේ පරිගණකයේ අතරින සත්කාරකයක් නිර්මාණය කිරීමෙන්, වෙබ් බුවුසරයක් හා විශයෙන් HTTP ඉල්ලීම් යැවීමට සහ ලැබීමට ඉඩ සම්පූර්ණ වෙබ් සේවාදායක පරිසරයක් අප පරිගණකය තුළ අනුකරණය කරයි.

This setup involves installing web server software, such as Apache or Nginx, along with PHP and a database server like MySQL. මෙම සැකසුම PHP සහ MySQL වැනි දත්ත සම්ඟ සේවාදායකයක් සමඟ Apache හෝ Nginx වැනි වෙබ් සේවාදායක මෘදුකාංග සේවානීය පරිගණකය කිරීම ඇතුළත් වේ.

Tools like XAMPP, WAMP, or MAMP make this process easier by bundling these components into a single package. Once installed, we can start the server software, which listens for incoming requests and processes PHP scripts.

XAMPP, WAMP, හෝ MAMP වැනි මෙවලම් මෙම සංවර්ධන තනි පැක්ෂයකට විකතු කිරීමෙන් මෙම ක්‍රියාවලිය පහසු කරයි. සේවානීය පෑස්, ලැබේන ඉල්ලීම්වලට සවන් දෙන සහ PHP ස්ක්‍රීප්ට් ක්‍රියාවට නංවන සේවාදායක මෘදුකාංගය ආරම්භ කළ හැක.

CLIENT SIDE



Server software packages

සේවාදායක මෘදුකාංග පැකේෂ

Tools like WAMP, XAMPP, MAMP, and LAMP are essential software packages that provide everything you need to create a virtual server on your local computer, enabling easy website development and application testing.

WAMP, XAMPP, MAMP, සහ LAMP වැනි මෙවලම් යනු ඔබේ ස්ථානීය පරිගණකයේ අතර් සේවාදායකයක් නිර්මාණය කිරීමට අවශ්‍ය සියලුම සපයනු ලබන අත්‍යවශ්‍ය මෘදුකාංග පැකේෂ වන අතර, පහසු වෙබ් අඩවි සංව්‍යෝගීතා සහ යෙදුම් පරික්ෂා කිරීම සඳහා කරයි.

They create a virtual server on your local machine, mimicking the environment of a live web server.

එමගින් සඡේට් වෙබ් සේවාදායකය පරිසරය අනුකරණය කරමින් ඔබේ ස්ථානීය පරිගණකයේ අතර් සේවාදායකයක් නිර්මාණය කරයි.

They bundle together a web server (Apache), a database server (MySQL or MariaDB), and server-side scripting languages (PHP, Perl, or Python).

එවා මගින් වෙබ් සේවාදායකයක් (Apache), දැන්ත සමුදාය සේවාදායකයක් (MySQL හෝ MariaDB) සහ සේවාදායක අගුරෙක් ස්ක්‍රීප්‍රේන් හාමා (PHP, Perl හෝ Python) විකට විකතු කරයි.

They simplify the setup process by providing pre-configured packages that are easy to install and manage, reducing the complexity of setting up each component individually.

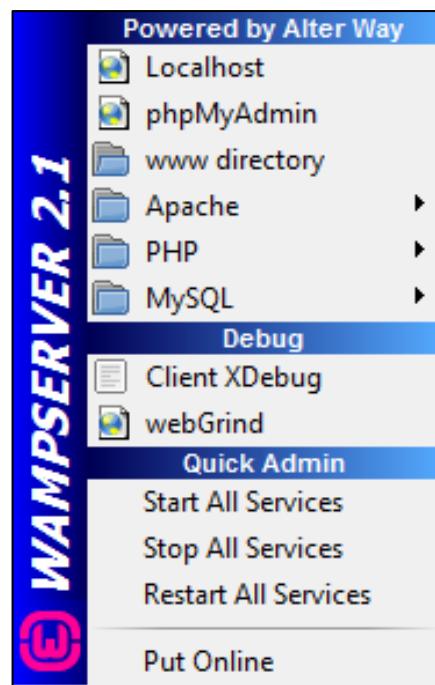
වික් වික් සංරචක තනි තනිව සැකසීමේ සංකීර්ණත්වය අඩු කරමින්, ස්ථාපනය කිරීමට සහ කළමනාකරණය කිරීමට පහසු වන පෙර-වින්කාසගත පැකේෂ ලබා දීමෙන් මෙවා මගින් සැකසුම් ක්‍රියාවලිය සරල කරයි.

WAMP (Windows, Apache, MySQL, PHP)

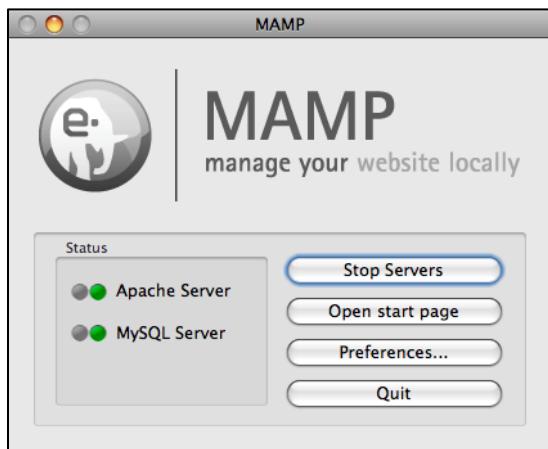


WAMP is tailored for Windows systems, combining Apache (web server), MySQL (database), and PHP (server-side scripting) to create a local server environment for developing PHP-based web applications on Windows.

WAMP Windows පද්ධති සඳහා සකස් කර ඇත, Apache (වෙබ් සේවාදායකය), MySQL (දැන්ත සමුදාය) සහ PHP (සේවාදායක අගුරෙක් ස්ක්‍රීප්‍රේන්) එකාබද්ධ කරමින් Windows මත ජ්‍යාල් මත පදනම් වූ වෙබ් යෙදුම් සංව්‍රේධනය කිරීම සඳහා ස්ථානීය සේවාදායක පරිසරයක් නිර්මාණය කරයි.



MAMP (macOS, Apache, MySQL, PHP)



MAMP is designed for macOS, featuring Apache, MySQL (or MariaDB), and PHP to facilitate the creation of local server environments on Apple computers, ideal for developing and testing PHP applications.

MAMP නිර්මාණය කර ඇත්තේ MacOS සඳහා වන අතර, Apache, MySQL (හෝ MariaDB) සහ PHP විශේෂාංගීව Apple පරිගණකවල ස්ථාපිය සේවාදායක පරිසරයන් නිර්මාණය කිරීමට පහසුකම් සැලසීමට, PHP යෙදුම් සංවර්ධනය කිරීම සහ පරික්ෂා කිරීම සඳහා වඩාත් සුදුසාය.

LAMP (Linux, Apache, MySQL, PHP)

LAMP is widely used on Linux systems, comprising Apache, MySQL (or MariaDB), and PHP.

Apache, MySQL (හෝ MariaDB) සහ PHP වෙළුන් සමන්විත Linux පද්ධති මත LAMP බහුවත භාවිත වේ.



It serves as a robust platform for deploying and testing PHP applications locally or on Linux-based servers, known for its flexibility and stability.

විය නම්සැල් බව සහ ස්ථාපිතාවය සඳහා ප්‍රසිද්ධ PHP යෙදුම් ස්ථාපිය හෝ Linux මත පදනම් වූ සේවාදායක මත යෙදුම්මට සහ පරික්ෂා කිරීමට ශක්තිමත් වේදිකාවක් ලෙස සේවය කරයි.

XAMPP (Cross-Platform, Apache, MySQL, PHP, Perl)



XAMPP is cross-platform, supporting Windows, macOS, and Linux. It includes Apache, MySQL (or MariaDB), PHP, and optionally Perl, providing developers with a unified stack for local web server setups across different operating systems.

XAMPP යනු Windows, macOS සහ Linux සඳහා සහය දක්වන හරක් වේදිකාවක් වේ. වියට Apache, MySQL (හෝ MariaDB), PHP සහ විකල්ප වශයෙන් Perl ඇතුළත් වේ, විවිධ මෙහෙයුම් පද්ධති තරඟා ස්ථාපිය වෙබ් සේවාදායක සැකසුම් සඳහා සංවර්ධකයින්ට එකාබද්ධ සංරචක කිරීමෙන් සපයයි.

Setting up WAMP

Download the WAMP server software from <https://sourceforge.net/projects/wampserver/>.

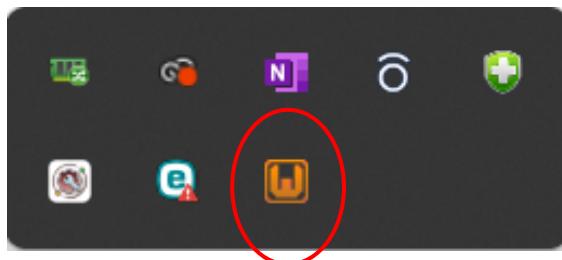
WAMP සේවාදායක මෘදුකාංගය

<https://sourceforge.net/projects/wampserver/>
වෙතින් බාගත කරගන්න.



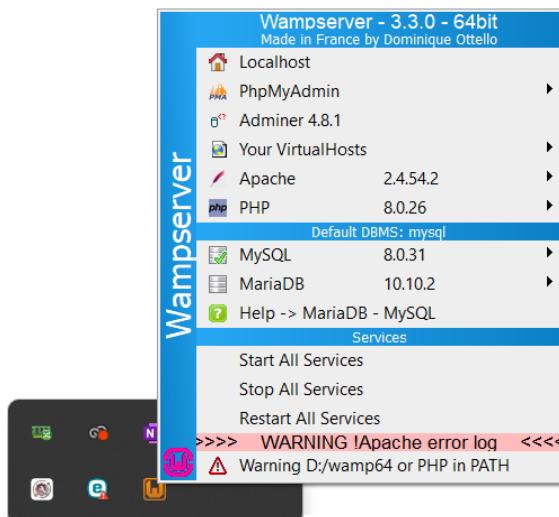
After installation, you should find a WAMP shortcut on your desktop or in your Start menu.

සේවාපනය කිරීමෙන් පසු, ඔබේ desktop විශේ හෝ ආර්ථික මෙනුවේ WAMP සඳහා කෙටිමගක් සොයා ගත හැකි විය යුතුය.



Click on the WAMP shortcut to launch the application.

යෙදුම දියත් කිරීමට WAMP කෙටිමග මත ක්ලික් කරන්න.



Open your web browser and type <http://localhost/> into the address bar.

You should see the WAMP server homepage, indicating that the server is running correctly.

ඔබගේ වෙබ් ඉවිසරය විවහ කර ලිපින තීරුවේ <http://localhost/> ලෙස රිස් කරන්න. විවිධ සේවාදායකය නිවැරදිව ක්‍රියාත්මක වන බව පෙන්වනු ඕම් කරමින් WAMP සේවාදායකයේ මූල් පිටුව දැකිය හැකි වේ.

The screenshot shows the WampServer configuration page. At the top, it displays system information: Apache 2.4, MySQL 5 & 8, MariaDB 10, and PHP 5, 7 & 8. The page title is "Wampserver". Below the title, it says "Version 3.3.0 - 64bit" and has language selection dropdowns for "english" and "classic".

Server Configuration

- Apache Version: 2.4.54.2 - [Documentation Apache](#)
- Server Software: Apache/2.4.54 (Win64) PHP/8.0.26 mod_fcgid/2.3.10-dev - Port defined for Apache: 80
- PHP Version: 8.0.26 - [Documentation PHP](#) - Loaded PHP extensions - Use of PHP versions
[FCGI] 7.4.33 - 8.0.26 - 8.1.13 - 8.2.0 - FCGI mode help
- MySQL Version: 8.0.31 - Port defined for MySQL: 3306 - default DBMS - [Documentation MySQL](#)
- MariaDB Version: 10.10.2 - Port defined for MariaDB: 3307 - [Documentation MariaDB](#) - MySQL - MariaDB

Tools

- [phpinfo\(\)](#)
- [xdebug_info\(\)](#)
- [PhpSysInfo 3.4.2](#)
- [Add a Virtual Host](#)

Your Projects (1)

- packagemanager

These are your folders in D:/wamp64/www To use them as an http link, you must declare them as VirtualHost. However, you can use Right-Click, Wamp Settings, Warning: risky..., Allow links on projects...

Your Aliases (2)

- [adminer 4.8.1](#)
- [PhpMyAdmin 5.2.0](#)

Your VirtualHost (1)

- [localhost](#)

[Wampserver Forum](#)

Once your set-up is done, you can start writing PHP programs

ඔබගේ සැකසුම අවසන් වූ පසු, ඔබට PHP වැඩසටහන් ලිඛීම ආරම්භ කළ හැක

By default, WAMP installs in C:\wamp64\ on 64-bit systems or C:\wamp\ on 32-bit systems.

පෙරහිමියෙන්, WAMP මෘදුකාංගය, 64-bit පද්ධති මත C:\wamp64\ හෝ 32-bit පද්ධති මත C:\wamp\ යන ස්ථානයන්හි ස්ථාපනය වේ.

Inside this directory, you'll find a folder named **www**. This is your virtual server for web projects.

මෙම නාමාවලිය තුළ, ඔබට **www** නම් ගොනු නාමාවලියක් හමුවනු ඇත. මෙය වෙබ් ව්‍යාපෘති සඳහා ඔබගේ අතත් සේවාදායකයයි.

Inside the **www** folder, create a new directory for your PHP project.

www ගේශ්ලීඩිය තුළ, ඔබේ PHP ව්‍යාපෘති සඳහා නව නාමාවලියක් සාදාගන්න.

For example, create a folder named myFirstProject.

ලදාහරණයක් ලෙස, myFirstProject නම් නාමාවලියක් සාදන්න.

The path should be something like

C:\wamp64\www\myproject.

විවිධ ගොනු මාර්ගය

C:\wamp64\www\myproject ලෙස වේ.

Open a text editor (such as, Visual Studio Code, Sublime text or any IDE of your choice).

පෙළ සංස්කාරකයක් විවෘත කරන්න (such as, Visual Studio Code, Sublime text වැනි ඔබ කෙමති සිනසම IDE එකක්).



Write your PHP code in the editor.

සංස්කාරකයේ ඔබේ PHP කේතය ලියන්න.

For example, create a file named index.php with the following content

ලදාහරණයක් ලෙස, පහත අන්තර්ගතය සමඟ index.php නම් ගොනුවක් සාදන්න

```
<?php  
echo "Hello, World!";  
?>
```

Save this file in the myFirstProject folder you created earlier.

මෙම ගොනුව ඔබ කළින් නිර්මාණය කළ myFirstProject නාමාවලියේ පුරකින්න.

(C:\wamp64\www\myproject\index.php)

Ensure WAMP is running, and the icon in the system tray is green, indicating all services are running.

WAMP ක්‍රියාත්මක වන බව සහ සියලු සේවාවන් ක්‍රියාත්මක වන බව පෙන්වනු මි කරමින් විෂ අයිකනය කොළ පාට බව සහතික කර ගන්න.

Open your web browser and type http://localhost/myproject/index.php in the address bar.

ඔබගේ වෙබ් ඉවිසරය ව්‍යවහර කර ලිපින තීරුවේ http://localhost/myproject/index.php යන්න විස්තර කරන්න.

You should see the output of your PHP code: Hello, World!.

දැන් ඔබේ PHP කේතයේ ප්‍රතිඵ්‍යානය Hello" World! ලෙසින් දිස් වනු ඇත.

NOTE: A file with a .php extension can contain HTML, CSS, and PHP code.

.php දිගුවක් සහිත ගොනුවක HTML, CSS සහ PHP කේත අඩංගු විය හැක.

Setting up XAMPP

Download the appropriate version for your operating system (Windows, macOS, Linux) ඔබගේ මෙහෙයුම් පද්ධතිය සඳහා සුදුසු අනුවාදය බාගත කරන්න (Windows, macOS, Linux)
<https://www.apachefriends.org/download.html>

Run the downloaded installer.
බාගත කළ ස්ථාපකය ධාවනය කරන්න.

Follow the installation instructions. Choose the components you need (usually Apache, MySQL, and PHP).
ස්ථාපන උපදෙස් අනුගමනය කරන්න. ඔබට අවශ්‍ය සර්වරවක තෝරන්න (සාමාන්‍යයෙන් Apache, MySQL, සහ PHP).

Complete the installation.
ස්ථාපනය සම්පූර්ණ කරන්න.

Launch the XAMPP Control Panel from the start menu or installation directory.
අරමිනක මෙහෙවත් හේ ස්ථාපන නාමාවලියෙන් XAMPP පාලන පැනවය දියත් කරන්න.

In the control panel, click the "Start" buttons next to Apache and MySQL modules.

පාලක පැනවයේ, Apache සහ MySQL මොඩුල අසළ ඇති "Start" බොත්තම් ක්ලික් කරන්න.

Navigate to the **htdocs** directory inside your XAMPP installation folder (usually C:\xampp\htdocs on Windows).

ඔබගේ XAMPP ස්ථාපන ගෝල්බරය තුළ ඇති **htdocs** නාමාවලිය වෙතට ප්‍රවේශ වෙන්න.
(සාමාන්‍යයෙන් Windows වල C:\xampp\htdocs)

Create a new folder for your project (Ex. my_project).

ඔබගේ ව්‍යාපෘතිය සඳහා නව නාමාවලියක් සාදන්න (ලඟ. (පහිචරදුවැජ්වී).)

Inside your project folder, create a new PHP file (Ex. index.php).

ඔබගේ ව්‍යාපෘති නාමාවලිය තුළ, නව (ඡ්‍යාප්) ගොනුවක් සාදන්න (ලඟ. (සංඛ්‍යා වියව).)

Type the following code

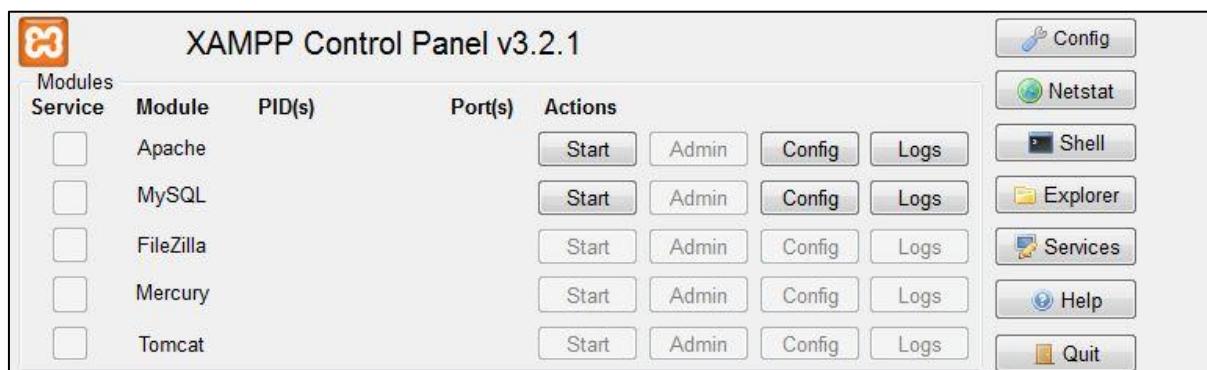
```
<?php  
echo "Hello, world!";  
?>
```

Launch your preferred web browser and in the address bar, type http://localhost/my_project/index.php and press Enter.

ඔබ කැමති වෙබ් ඕවිසරය දියත් කර ලිපින තිරැවේ,
http://localhost/my_project/index.php යන්න වසිල් කර Enter ඔබන්න.

You should see the output displayed in your browser.

ඔබගේ බිරුවිසරයේ දුරක්ෂය වෙත "Hello, world!" නම් ප්‍රතිදානය දැකිය යුතුය.



PHP syntax

PHP code should be written in-between `<?php` and `?>` tags

PHP කේත `<?php` සහ `?>` යන උපුමන අතර ලිවිය යුතුය

```
<?php  
// PHP code goes here  
?>
```

PHP code is embedded within HTML using PHP tags.

PHP කේත PHP උපුමන හාවිතයෙන් HTML තුළ අන්තර්ගත කර ඇත.

```
<!DOCTYPE html>  
<html>  
<head></head>  
<body>  
  
<?php  
    echo "Hello World!";  
?>  
  
</body>  
</html>
```

Semicolon (;)

In PHP, a semicolon is used to notify the computer that one instruction has finished and a new instruction is about to begin.

PHP හිදි, වික් උපදෙසක් අවසන් වූ පසු එම පරිගණකයට දැනුම් දීම සඳහාත් නව උපදෙසක් ආරම්භ වීමට යන බව දැනුම් දීම සඳහාත් semicolon විකක් හාවිතා වේ.

echo and print in PHP

In PHP, both echo and print are used to output data to the browser or command line.

PHP හිදි, ඉවුසරයට හෝ විභාග පේලියට දත්ත ප්‍රතිල්‍යුතු කිරීමට echo සහ print යන දෙකම හාවිතා වේ.

echo

Type:

Language construct

Language construct විකක්

Usage:

Can take multiple parameters

පරාම්ත කිහිපයක් ලබා ගත හැක

Return Value:

No return value (always returns void)
අපසු අයෙක් නැත (සංම විටම විය හිස් වේ)

Performance:

Slightly faster than print
print ට වඩා තරමක් වේගවත්

print

Type:

Language construct

Language construct විකක්

Usage:

Can only take one argument

ගත හැක්කේ වික් පරාම්තියක් පමණි

Return Value:

Returns 1 (can be used in expressions)
1 අපසු ලබාදේ (ප්‍රකාශනවල හාවිතා කළ හැක)

Performance:

Slightly slower than echo due to the
return value

අපසු අය හේතුවෙන් echo ට වඩා තරමක්
මන්දගාමී වේ

PHP variables

Variables in PHP are used to store data that can be manipulated and retrieved later in the script.

PHP හි විවෘතයන් භාවිතා කරනුයේ කේතයෙහි හැසිරවිය හැකි සහ පසුව එබා ගත හැකි දත්ත ගබඩා කිරීමටය.

NOTE: Variables in PHP should be declared using a dollar sign (\$), informing the server computer that a variable is going to be created.

PHP හි විවෘතයන් බොලට් ලකුණුකින් (\$) ආරම්භ විය යුතු අතර විමූහින් විවෘතයක් සඳහාමට යන වග සේවාදායක පරිගණකය දැනුවත් කරයි.

PHP variables can hold values of any data type and do not require explicit declaration of the type.

PHP විවෘතයන්ට ඕනෑම දත්ත වර්ගයක අයයන් තබාගත හැකි අතර විම දත්ත වර්ගය ප්‍රථමයෙන් අර්ථ දැක්වීම අවශ්‍ය නොවේ.

Variable names in PHP are case-sensitive, meaning \$variable and \$Variable are treated as different variables.

PHP හි විවෘත භාම අක්ෂර-සංවේදී වේ, විනම් \$variable සහ \$Variable විකිණීකර වෙනස් විවෘතයන් ලෙස සලකනු ලැබේ.

But the PHP language itself is not case-sensitive

නමුත් PHP භාෂාව අක්ෂර-සංවේදී නොවේ

Ex. Echo, echo, eCHo are the same in PHP.
PHP හිදී Echo, echo, eCHo යනු විකම වේ.

Variables have a scope that defines where they can be accessed within the code.

(Scopes will be discussed later)

විවෘතයන්ට කේතය තුළ ප්‍රධාන විය හැකි ස්ථාන තීරුවන්හි කරන පටයක් ඇත.

(පට පිළිබඳව පසුව සාකච්ඡා කරනු ඇත)

Variable naming rules

විවෘත භාමකරණ නීති

Must start with a letter or an underscore _
අකුරකින් හෝ යටි ඉරකින්_ ආරම්භ කළ යුතුය

Cannot start with a number

සංඛ්‍යාවකින් ආරම්භ කළ නොහැක

Can only contain alphanumeric characters and underscores (A-z, 0-9, and _)

අධිංශ විය හැක්කේ අක්ෂරයාංක සමඟතු සහ යටි ඉර (A-z, 0-9, සහ _) පමණි

Reserved words of the language cannot be used as variable names.

භාෂාවේ ඇති ඇවිරුණු පද විවෘත නම් ලෙස භාවිතා කළ නොහැක.

Ex.

```
<?php
/* Area and perimeter of a
Rectangle */
length = 12;
width = 5;
$a = $length * $width;
$p = 2*($length + $width);
echo "Area is $a";
echo "<br>";
echo "Perimeter is $p";
echo "<br>"; echo "<br>";
print "Area is $a";
print "<br>";
print "Perimeter is $p";
?>
```

Area is 60
Perimeter is 34

Area is 60
Perimeter is 34

PHP comments

PHP supports several types of comments which will be treated as comments.

Single-line comments

- **Double Slash (//)**

Everything on the right of // will be treated as a comment.

// 10 + 20 /* 50 */ + 30;
The 50 is treated as a comment.

```
<?php  
// This is a single-line comment  
echo "Hello, World!";  
?>
```

- **Hash (#)**

Similar to //, everything on the right of # will be treated as a comment.

// 10 + 20 # 30;
The 30 is treated as a comment.

```
<?php  
# This is a single-line comment  
echo "Hello, World!";  
?>
```

Multi-line comments

- **Slash-Asterisk (* ... *)**

Encloses a block of text, allowing for multi-line comments.

The text between /* and */ is treated as a comment.

```
<?php  
/*  
This is a multi-line comment  
that spans multiple lines.  
*/  
echo "Hello, World!";  
?>
```

NOTE: Multi-line comments can be used to comment a part of a single-line comment.

බහු-පේලි විවරණ තනි පේලියේ විවරණයක කොටසක් විවරණයක් බවට පත් කිරීමට හාවිතා කළ හැකිය.

```
<?php
```

```
$x = 10 + /* 50 */ + 20;  
echo $x; # 30  
?>
```

Here, /* 50 */ is a comment. Therefore, only 10 and 20 will be added.

මෙහි /* 50 */ යනු විවරණයකි. විභැවීන් 10 සහ 20 පමණක් විකතු වේ.

Why use comments?

විවරණ හාවිතා කරන්නේ ඇයි?

- Readability: Improve code understanding.

කියවේමේ හැකියාව: කේත අවබෝධය වැඩි දියුණු කරයි.

- Debugging: Isolate and identify issues.

ගැටෙන නුදකලා කර නැඹුනා ගනියි.

- Maintenance: Facilitate future updates.

නඩත්තු කිරීම: අනාගත යාවත්කාලීන සඳහා

පහසුකම් සැලැසීම.

- Collaboration: Enhance team communication.

සහයෝගීතාව: කණ්ඩායම් සහ්තිවේදනය

වැඩි දියුණු කරයි.

- Documentation: Provide code context and purpose.

කේත සහ්දාර්යය සහ අරමුණ සැපයීම.

PHP operators

PHP මෙහෙයවන

1. Arithmetic Operators

අංක ගණිතමය මෙහෙයවන

These operators are used to perform common arithmetic operations

මෙම මෙහෙයවන පොදු අංක ගණිත ක්‍රියා සිදු කිරීමට භාවිතා කරයි

Addition (+)

Adds two numbers.

සංඛ්‍යා දෙකක් විශ්‍යම කරයි.

```
<?php  
echo 10 + 5; # 15  
?>
```

Subtraction (-)

Subtracts one number from another.

විශ්‍යම සංඛ්‍යාවකින් තවත් සංඛ්‍යාවක් අඩු කරයි.

```
<?php  
echo 10 - 5; # 5  
?>
```

Multiplication (*)

Multiply two numbers.

සංඛ්‍යා දෙකක් වැශ්‍යම කරයි.

```
<?php  
echo 10 * 5; # 50  
?>
```

Division (/)

Divides one number by another.

සංඛ්‍යාවක් තවත් සංඛ්‍යාවකින් බෙදාහු බඳාපෑම්.

```
<?php  
echo 10 / 5; # 2  
?>
```

Modulus (%):

Returns the remainder of a division.

බේදීමක ඉතිරි අගය ලබා දෙයි.

```
<?php  
echo 11 % 5; # 1  
?>
```

Exponentiation (**):

Raises one number to the power of another.

සංඛ්‍යාවක් තවත් සංඛ්‍යාවක බලය ලෙස ලබා දෙයි.

```
<?php  
$a = 5;  
$b = 2;  
echo $a ** $b; # 25  
?>
```

NOTE: In PHP, python's integer division (//) is not used as an operator. Instead, it is used to insert single-line comments.

PHP පිළි, පසිනහ් වල නිඩිල බේදීම (//) මෙහෙයවනයක් ලෙස භාවිතා නොවේ. ඒ වෙනුවට විය තහි පෝළීයේ විවරණ ඇතුළත් කිරීමට භාවිතා කරයි.

2. Assignment Operators

അടുക്ക മേഖലയിൽ

These operators are used to assign values to variables

മേഖലയിൽ വിവരങ്ങൾ സിദ്ധാന്തം അനുസരിച്ച് പെടുത്തണം.

Basic Assignment (=)

Assigns the value on the right to the variable on the left.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 10;  
$b = $a;  
?>
```

Addition Assignment (+=)

Adds the right operand to the left operand and assigns the result to the left operand.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 10;  
$a += 5;  
echo $a; # 15  
?>
```

Subtraction Assignment (-=)

Subtracts the right operand from the left operand and assigns the result to the left operand.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 10;  
$a -= 5;  
echo $a; # 5  
?>
```

Multiplication Assignment (*=)

Multiplies the right operand by the left operand and assigns the result to the left operand.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 10;  
$a *= 5;  
echo $a; # 50  
?>
```

Division Assignment (/=)

Divides the left operand by the right operand and assigns the result to the left operand.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 10;  
$a /= 5;  
echo $a; # 2  
?>
```

Modulus Assignment (%=)

Calculates the modulus of the left operand and the right operand and assigns the result to the left operand.

ഉദാഹരണം നിരീക്ഷണം ചെയ്യാം.

```
<?php  
$a = 11;  
$a %= 5;  
echo $a; # 1  
?>
```

3. Comparison Operators

සංස්කරණ මෙහෙයවන

These operators are used to compare two values

මෙම මෙහෙයවන අගයන් දෙකක් සංස්කරණය කිරීමට භාවිත කරයි

Equal (==)

Checks if two values are equal.

අගයන් දෙකක් සමාන දැයි පරීක්ෂා කරයි

```
<?php  
$a = 10;  
$b = 10;  
echo $a == $b; # 1  
?>
```

Identical (== =)

Checks if two values are equal and of the same type.

අගයන් දෙකක් සමාන සහ විකම වර්ගයේ දැයි පරීක්ෂා කරයි.

```
<?php  
$a = 10;  
$b = 10;  
echo $a === $b; # 1  
?>
```

Not Equal (!= or <>)

Checks if two values are not equal.

අගයන් දෙකක් සමාන නොවේදයි පරීක්ෂා කරයි.

```
<?php  
$a = 10;  
$b = 5;  
echo $a != $b; # 1  
echo $a <> $b; # 1  
?>
```

Not Identical (!==)

Checks if two values are not equal or not of the same type.

අගයන් දෙකක් සමාන නොවේද නැතහොත් විකම වර්ගයේ නොවේදයි පරීක්ෂා කරයි.

```
<?php  
$a = 10;  
$b = 10.0;  
echo $a !== $b; # 1  
?>
```

Less Than (<)

Checks if the value on the left is less than the value on the right.

මම පස ඇති අගය දකුණේ අගයට වඩා අමු දැයි පරීක්ෂා කරයි.

```
<?php  
$a = 10;  
$b = 5;  
echo $a < $b;  
?>
```

Greater Than (>)

Checks if the value on the left is greater than the value on the right.

මම පස ඇති අගය දකුණේ අගයට වඩා වැඩි දැයි පරීක්ෂා කරයි.

```
<?php  
$a = 10;  
$b = 5;  
echo $a > $b; # 1  
?>
```

Less Than or Equal To (<=)

Checks if the value on the left is less than or equal to the value on the right.

වම් පස ඇති අගය දකුණු පස ඇති අගයට වඩා අඩු හෝ සමානදැයි පරීක්ෂා කරයි.

```
<?php
$a = 10;
$b = 10;
echo $a <= $b; # 1
?>
```

Greater Than or Equal To (>=)

Checks if the value on the left is greater than or equal to the value on the right.

වම් පස ඇති අගය දකුණු පස ඇති අගයට වඩා වැඩි හෝ සමානදැයි පරීක්ෂා කරයි.

```
<?php
$a = 10;
$b = 11;
echo $a >= $b;
?>
```

4. Logical Operators

තාර්කික මෙහෙයුවන

These operators are used to combine conditional statements

කොන්දේසි සහිත ප්‍රකාශ ඒකාබද්ධ කිරීමට මෙම මෙහෙයුවන හාවිතා කරයි

And (&& or and)

Returns true if both operands are true.

මෙහෙයුම් දෙකම සත්‍ය නම් සත්‍ය ලෙස ලබා දේ.

```
<?php
$a = 10;
$b = 5;
$c = 4;
echo $a > $b && $b > $c; # 1
?>
```

Or (|| or or)

Returns true if at least one of the operands is true.

අවම වශයෙන් වික් මෙහෙයුමක්වත් සත්‍ය නම් සත්‍ය ලෙස ලබා දේ.

```
<?php
$a = 10;
$b = 5;
$c = 4;
echo $a > $b || $b < $c; # 1
?>
```

Not (!)

Reverses the logical state of its operand.

එහි operand හි තාර්කික තත්ත්වය ආපසු හරවයි.

```
<?php
$b = 0;
echo !$b; # 1
?>
```

Xor (xor)

Returns true if either operand is true, but not both.

මෙයාන්ස් දෙකෙන් විකක් සත්‍ය නම් සත්‍ය ලබා දෙයි, නමුත් දෙකම නොවේ.

```
<?php  
$a = 10;  
$b = 5;  
$c = 4;  
echo $a > $b xor $b < $c; # 1  
echo $a > $b xor $b > $c;  
?>
```

5. Increment/Decrement Operators

These operators are used to increment or decrement a variable's value:

විවෘතයක අගය වැඩි කිරීමට හෝ අඩු කිරීමට මෙම මෙහෙයුවන හාටිතා කරයි

Increment (++)

Increases a variable's value by one.
විවෘතයක අගය විකතින් වැඩි කරයි.

They are two types.

ආකාර දෙකක්

1. Pre-increment

In pre-increment, the variable's value is incremented first, and then the new value is used in the expression.

Pre-increment හිදී, විවෘතයේ අගය පළමුව වැඩි කරනු ලබන අතර, පසුව නව අගය ප්‍රකාශනයේ හාටිතා වේ.

```
<?php  
$number = 5;  
echo ++$number; // Outputs: 6  
echo $number; // Outputs: 6  
?>
```

2. Post-increment

In post-increment, the current value of the variable is used in the expression first, and then the variable's value is incremented.

Post-increment හිදී, විවෘතයේ වත්මන් අගය ප්‍රකාශනයේ පළමුව හාටිතා කරනු ලබන අතර, පසුව විවෘතයේ අගය වැඩි වේ.

```
<?php  
$number = 5;  
echo $number++; // Outputs: 5  
echo $number; // Outputs: 6  
?>
```

Decrement (--)

Decreases a variable's value by one.
විවෘතයක අගය විකුත් අඩු කරයි.

They are two types.
ආකාර දෙකක්

1. Pre-decrement

In pre-decrement, the variable's value is decremented first, and then the new value is used in the expression.

Pre-decrement හිදී, විවෘතයේ අගය පළමුව අඩු කරනු ලබන අතර, පසුව නව අගය ප්‍රකාශනයේ භාවිතා වේ.

```
<?php
$number = 5;
echo --$number; // Outputs: 4
echo $number;   // Outputs: 4
?>
```

2. Post-decrement

In post-decrement, the current value of the variable is used in the expression first, and then the variable's value is decremented.

Post-decrement හිදී, විවෘතයේ වත්මන් අගය ප්‍රකාශනයේ පළමුව භාවිතා කරනු ලබන අතර, පසුව විවෘතයේ අගය අඩු වේ.

```
<?php
$number = 5;
echo $number--; // Outputs: 5
echo $number;   // Outputs: 4
?>
```

6. String Operators

These operators are used to manipulate strings
මෙම මෙහෙයුවන තන්තු හැසිරීමට භාවිත කරයි

Concatenation (.)

Concatenates two strings.
තන්තු දෙකක් විකට සම්බන්ධ කරයි.

```
<?php
```

```
$firstName = "John";
$lastName = "Hamilton";
$fullName = $firstName . $lastName;
echo "Full Name is $fullName";
```

```
?>
```

Full Name is John Hamilton

Concatenation Assignment (.=)

Appends the right string to the left string and assigns the result to the left variable.
දකුණුති තන්තුව වම් තන්තුවට විකතු කර වම් විවෘතයට ප්‍රතිඵලය ආද්‍යෝ කරයි.

```
<?php
```

```
$fullName = "John";
$fullName .= " Hamilton";
echo "Full Name is $fullName";
```

```
?>
```

Full Name is John Hamilton

var_dump() in PHP

The `var_dump()` function in PHP is used to display structured information (type and value) about one or more variables.

PHP හි `var_dump()` ශ්‍රීතය විවෘත විකක් හේ කිහිපයක් තෙන ව්‍යුහගත තොරතුරු (දැන්ත වර්ගය සහ අගය) පෙන්වීමට හාවතා කරයි.

It provides a detailed output, making it especially useful for debugging purposes. විය සවිස්තරාත්මක ප්‍රතිඵානයක් සපයයි, විය ලේඛන නිර්මාණය කිරීමේ අරමුණු සඳහා විශේෂයෙන් පෙශේෂනවත් වේ.

Syntax

```
<?php
var_dump($variable1, $variable2, ...);
?>
```

Ex.

```
<?php
$number = 42;
$string = "Hello, World!";
$float = 3.14;
$boolean = true;

var_dump($number);
var_dump($string);
var_dump($float);
var_dump($boolean);
?>

int(42)
string "Hello, world!" (length=13)
float 3.14
boolean true
```

Data types in PHP

PHP වල දත්ත වර්ග

String

A sequence of characters enclosed in single or double quotes.

තනි හේ ද්වීතීව උද්ධිභවලින් වට කර ඇති සලකුණු අනුලිපිවෙළකි.

```
<?php
$name = "John";
Var_dump($name);
?>
```

string "John" (length=4)

Integer

A non-decimal number between

-2,147,483,648 and 2,147,483,647.

-2,147,483,648) සහ 2,147,483,647 අතර දැනුම නොවන සංඛ්‍යාවක්.

```
<?php
$number = 42;
Var_dump($number);
?>
```

int 42

Float (Double)

A number with a decimal point.

දැනුම ස්ථානයක් සහිත සංඛ්‍යාවක්.

```
<?php
$floatNumber = 3.14;
Var_dump($floatNumber);
?>
```

float 3.14

Boolean

Represents two possible states: TRUE or FALSE.

පැවතිය හැකි අවස්ථා දෙකක් නියෝජනය කරයි: සත්‍ය හෝ අසත්‍ය.

```
<?php  
$bool1 = true;  
$bool2 = false;  
Var_dump($bool1, $bool2);  
?>
```

boolean true
boolean false

Array

A collection of values. Each value can be accessed using an index or key.

අගයන් කිහිපයක විකතුවකි. සෑම අගයකටම දුර්කායක් හෝ key විකක් භාවිතයෙන් ප්‍රවේශ විය හැක.

There are three main types of arrays in php.
php හී ප්‍රධාන ආරා වර්ග තනක් ඇත.

1. Indexed arrays
2. Multi-dimensional arrays
3. Associative arrays

1. Indexed arrays

These are arrays with numeric indices.
මේවා සංඛ්‍යාංකමය දුර්කක සහිත ආරාවන් වේ.

```
<?php  
  
$indexedArray = array(10, "apple",  
3.14, true);  
  
var_dump($indexedArray);  
  
?>  
  
array (size=4)  
  0 => int 10  
  1 => string 'apple' (length=5)  
  2 => float 3.14  
  3 => boolean true
```

2. Multi-dimensional arrays

These are Arrays containing one or more arrays.

මෙවා ආරච්ඡන් විකක් හේ කිහිපයක් අධිංග
Arrays වේ.

```
<?php  
  
$multiArray = array(  
    "nums" => array(10, 20, 30),  
    "letters" => array("a", "b", "c")  
);  
  
Var_dump($multiArray);  
  
?>  
  
array (size=2)  
  'nums' =>  
    array (size=3)  
      0 => int 10  
      1 => int 20  
      2 => int 30  
  'letters' =>  
    array (size=3)  
      0 => string 'a' (length=1)  
      1 => string 'b' (length=1)  
      2 => string 'c' (length=1)2.
```

3. Associative arrays

Arrays that use named keys instead of indices.

දැරුණක වෙනුවට keys හාවතා කරන ආරච්ඡන් වේ.

```
<?php  
  
$assocArray = array("a" => 10,  
"b" => "apple", "c" => 3.14,  
"d" => true);  
  
Var_dump($assocArray);  
  
?>  
  
array (size=4)  
  'a' => int 10  
  'b' => string 'apple' (length=5)  
  'c' => float 3.14  
  'd' => boolean true
```

Data types summary

| Data Type | Description | Example |
|-----------|-----------------------------|---|
| String | Sequence of characters | \$string = "Hello, World!"; |
| Integer | Non-decimal number | \$integer = 42; |
| Float | Number with a decimal point | \$float = 3.14; |
| Boolean | TRUE or FALSE | \$boolean = true; |
| Array | Collection of values | \$array = array("apple", "banana", "cherry"); |

if-else conditions

If conditions allow you to execute specific blocks of code based on whether a given condition evaluates to true or false.

දී ඇති කොන්දේසියක් සත්‍ය හෝ අසත්‍ය ලෙස ඇගයීමට ලක් කරන්නේද යන්න මත පදනම්ව නිශ්චිත කේත කොටස් ත්‍රියාත්මක කිරීමට ඔබට ඉඩ ලබා දෙයි.

This control structure is fundamental for creating dynamic and responsive applications.

ගතික සහ ප්‍රතිචාරාත්මක යෙදුම් නිර්මාණය කිරීම සඳහා මෙම පාලන ව්‍යුහය මූලික වේ.

Syntax

```
<?php
if (condition) {
    /* code to be executed if
       condition is true */
} else {
    /* code to be executed if
       condition is false */
}
?>
```

Ex.

```
<?php
$isRaining = true;

if ($isRaining) {
    echo "Bring an umbrella";
} else {
    echo "Bring a cap";
}
?>
```

Bring an umbrella

if-elseif-else Statement

You can test multiple conditions using elseif.

elseif භාවිතයෙන් ඔබට කොන්දේසි කිහිපයක් පරීක්ෂා කළ හැක.

Ex.

```
<?php
$score = 85;

if ($score >= 90) {
    echo "Grade: A";
} elseif ($score >= 80) {
    echo "Grade: B";
} elseif ($score >= 70) {
    echo "Grade: C";
} else {
    echo "Grade: F";
}
?>
Grade: B
```

Nested If Statements

If statements inside another if statement are known as nested if statements.

වෙනත් If ප්‍රකාශනයක් තුළ ඇති if ප්‍රකාශයෙන් nested if ප්‍රකාශ ලෙස හැඳුන්වේ.

```
<?php
$age = 20;
$hasID = true;

if ($age >= 18) {
    if ($hasID) {
        echo "You can enter.";
    } else {
        echo "You need an ID to
              enter.";
    }
} else {
    echo "You are not old enough
          to enter.";
}
?>
```

Shorthand If

The shorthand if statement in PHP is known as the **ternary operator**.

PHP හි ඇති කෙටි if ප්‍රකාශය මෙසේ හැඳුන්වේ.

It allows you to write a compact form of the if-else statement in a single line.

විය ඔබට if-else ප්‍රකාශයේ සංයුත්ත ආකෘතියක් තහි ජෝලියකින් ලිවීමට ඉඩ සමස්සේ.

Syntax

```
<?php  
condition ? true_value : false_value;  
?>
```

condition: The expression to be evaluated.

අගයීමට ඔක් කළ යුතු ප්‍රකාශනය.

true_value: The value to return if the condition is true.

කොහොදේසිය සත්ත්‍ය නම් ආපසු බඟා දිය යුතු අගය.

false_value: The value to return if the condition is false.

කොහොදේසිය අසත්ත්‍ය නම් ආපසු බඟා දිය යුතු අගය.

```
<?php
```

```
$age = 20;
```

```
$message = ($age >= 18) ? "You are  
an adult." : "You are a minor.;"
```

```
echo $message;
```

```
?>
```

You are an adult.

Nested Ternary Operators

You can nest ternary operators, but this can reduce readability.

Ternary operators, nest කළ හැකිය, නමුත් මෙය කියවීමේ හැකියාව අඩු කළ හැක.

```
<?php  
$score = 85;  
  
$grade = ($score >= 90) ? "A" :  
    (($score >= 80) ? "B" :  
        (($score >= 70) ? "C" : "F"));  
  
echo $grade;  
?>
```

B

With echo Statement

```
<?php  
$loggedIn = true;  
  
echo $loggedIn ? "Welcome back!" :  
    "Please log in.";  
  
?>
```

Welcome back!

Switch Statement

The switch statement in PHP is used to perform different actions based on different conditions.

PHP හි Switch ප්‍රකාශය විවිධ කොන්දේසි මත පදනම්ව විවිධ ක්‍රියා සිදු කිරීමට හාවිතා කරයි.

It is a cleaner and more readable way to compare a variable against multiple values than using multiple if statements.

විය if ප්‍රකාශන කිහිපයක් හාවිතා කරනවාට වඩා බඩු අගයන් සමඟ විවලුණයක් සංසන්දුනය කිරීමට වඩාත් තොදු සහ තියෙන් හැකි ක්‍රමයකි.

Syntax

```
<?php  
  
switch (variable) {  
    case value1:  
        /* code to be executed if  
         variable equals value1 */  
        break;  
    case value2:  
        /* code to be executed if  
         variable equals value2 */  
        break;  
    default:  
        /* code to be executed if  
         variable does not match any  
         case */  
}  
  
?>
```

How It Works

විය ක්‍රියා කරන ආකාරය

switch: The keyword that initiates the switch statement.
switch ප්‍රකාශය ආරම්භ කරන මූල පදය.

variable: The variable whose value you want to compare against different cases.
විවිධ අවස්ථා සමඟ විවලුණය සංසන්දුනය කිරීමට අවශ්‍ය විවෘත සම්බන්ධය.

case: Each possible value that the variable can have. If the variable matches a case value, the corresponding block of code is executed.
විවලුණයට තිබිය හැකි සෑම අගයක්ම. විවලුණය case නි අගයකට ගැළපේ නම්, අභාළ කේත කොටස ක්‍රියාත්මක වේ.

break: This keyword terminates the case once it is executed. Without break, PHP will continue to execute the following cases, a behavior known as "fall-through."
මෙම මූල පදය විය ක්‍රියාත්මක කළ පසු case වික අවසන් කරයි. break ප්‍රකාශයන් තොරව, PHP පහත සඳහන් අවස්ථා ක්‍රියාත්මක කිරීම දීගෙටම කරගෙන යනු ඇත, විය "fall-through" ලෙස හැඳුන්වේ.

default: An optional case that will be executed if the variable does not match any of the specified cases.
විවලුණය තිශ්චිත අවස්ථා කිසිවක් සමඟ තොගැලපේ නම් ක්‍රියාත්මක වන විකල්ප අවස්ථාවකි

Ex.

```
<?php  
$day = "Wednesday";  
  
switch ($day) {  
    case "Monday":  
        echo "Today is Monday.";  
        break;  
    case "Tuesday":  
        echo "Today is Tuesday.";  
        break;  
    case "Wednesday":  
        echo "Today is Wednesday.";  
        break;  
    case "Thursday":  
        echo "Today is Thursday.";  
        break;  
    case "Friday":  
        echo "Today is Friday.";  
        break;  
    default:  
        echo "Invalid day.";  
}  
?>
```

Today is Wednesday.

Example with numbers.

```
<?php  
$number = 2;  
  
switch ($number) {  
    case 1:  
        echo "Number is 1";  
        break;  
    case 2:  
        echo "Number is 2";  
        break;  
    case 3:  
        echo "Number is 3";  
        break;  
    default:  
        echo "Number is not 1, 2,  
        or 3";  
}  
?>
```

Number is 2

Example with Default Case

```
<?php  
$color = "purple";  
  
switch ($color) {  
    case "red":  
        echo "Color is red.";  
        break;  
    case "blue":  
        echo "Color is blue.";  
        break;  
    case "green":  
        echo "Color is green.";  
        break;  
    default:  
        echo "Color is not red,  
        blue, or green.";  
}  
?>
```

Color is not red, blue, or green.

Loops in PHP

Loops are used to execute the same block of code repeatedly as long as a certain condition is met.

යම් කොන්දේසියක් සපුරා අඟි තාක් විකම කේතය නැවත නැවතත් ක්‍රියාත්මක කිරීමට ලුප භාවිතා වේ.

There are four main types of loops in PHP
PHP හි ප්‍රධාන ලුප වර්ග හතරක් ඇත

1. for
2. foreach
3. while
4. do-while

For loop

The for loop is used when you know in advance how many times you want to execute a statement or a block of statements.

ප්‍රකාශයක් හෝ ප්‍රකාශ සමූහයක් ක්‍රියාත්මක කිරීමට කොපමණා වාර ගණනක් අවශ්‍ය දැයි කළීන් දුන්නා විට for loop භාවිතා වේ.

Syntax

```
<?php  
  
for (initialization; condition;  
increment) {  
    // code to be executed  
}  
  
?>
```

Ex1.

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10

Ex2.

```
<?php  
for ($x = 1; $x <= 10; $x++) {  
    echo "5 x $x = " , (5*$x);  
    echo "<br>";  
}  
?>
```

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

While loop

The while loop executes a block of code as long as the specified condition is true.

ලබා දී ඇති කොන්දේසිය සහය වන තාක් while ලුපය මගින් කේත කොටසක් ක්‍රියාත්මක කරයි.

Syntax

```
<?php  
  
while (condition) {  
    // code to be executed  
}  
?>
```

We can control the number of times that a while loop executes in three ways.

while ලුපයක් ක්‍රියාත්මක වන වාර ගණන ආකාර තුනකින් පාලනය කළ නැතිය.

1. By changing the control variable.
පාලන විවෘතය වෙනස් කිරීමෙන්.

2. By changing the condition.
කොන්දේසිය වෙනස් කිරීමෙන්.

3. By changing the increment/decrement of the control variable.

පාලන විවෘතයේ වැඩිකිරීම්/අඩුකිරීම් වෙනස් කිරීමෙන්.

```
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5
```

Ex2.

```
<?php  
  
$x = 1;  
  
while ($x <= 5) {  
  
    echo "5 x $x = ", 5 * $x;  
    echo "<br>";  
    $x++;  
}  
?>
```

```
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25
```

Ex1.

Control
variable

```
<?php  
  
$x = 1;  
  
while ($x <= 5) {  
  
    echo "The number is: $x <br>";  
  
    $x++;  
}  
?>
```

Increment/Decrement

Do-While Loop

The do-while loop will always execute the block of code inside it once, then it will check the condition and repeat the loop as long as the specified condition is true.

do-while ලුපය සමඟ විටම ඒ තුළ ඇති කේත කොටස වික් වරක් ක්‍රියාත්මක කරයි, පසුව විය කොන්දේසිය පරික්ෂා කර නිශ්චිත කොන්දේසිය අත් වන තාක් ලුපය නැවත ක්‍රියාත්මක කරයි.

Syntax

```
<?php  
do {  
    // code to be executed  
} while (condition);  
  
?>
```

Ex.

```
<?php  
$x = 1;  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
  
?>
```

```
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5
```

Let's consider a scenario where the condition is false at the very beginning.
අරමිහයෝදුම කොන්දේසිය අසත් වන අවස්ථාවක් සලකා බලමු.

```
<?php  
  
$weight = 52;  
$maxWeight = 45;  
do {  
    echo "Weight($weight) is less  
than max weight($maxWeight).";  
    echo "<br>";  
    $weight++;  
} while ($weight <= $maxWeight);  
  
echo "Weight($weight) exceeded max  
weight($maxWeight).";  
  
?>
```

```
Weight(52) is less than max weight(45).  
Weight(53) exceeded max weight(45).
```

In the above example, normal weight is already higher than the max weight when the program starts.

ඉහත උදාහරණයේ දී, වැඩසටහන ආරමිහ වන විටදීම බර උපරිම බරට වඩා වැඩි ය.

But the do-while loop was executed once, outputting that the weight is less than max weight.

නමුත් බර උපරිම බරට වඩා අඩු බව ප්‍රතිඵ්‍යා කරමින් do-while ලුපයක් වික වරක් ක්‍රියාත්මක විය.

So, a do-while loop executes the instruction inside it at least one time.

එබඳවීන්, do-while ලුපයක් අවම වශයෙන් වික් වරක්වන් විත ඇති උපදෙස් ක්‍රියාත්මක කරයි.

Foreach Loop

The foreach loop is used to loop through arrays.

foreach ලුපය ආරාවන් හරහා පුනරාවර්තනය කිරීමට හාඛිනා කරයි.

For every iteration, the value of the current array element is assigned to a variable, and the array pointer is moved by one, until it reaches the last array element.

සම්පූර්ණ සඳහාම, වත්මන් අරා මුලාවයටේ ඇගය විවෘතයකට පවරනු බෙන අතර, අවසාන අරා මුලාවයට වෙත පෙන තෙක් අරා දුර්ගකය විකැක්න් වැඩි කරනු ලැබේ.

Syntax

```
<?php  
foreach ($array as $value) {  
    // code to be executed  
}  
?  
>
```

Here, each element in the \$array variable is assigned to the \$value variable in each iteration until the loop reaches the last element.

මෙහිදී, \$array විවෘතයේ ඇති සම්පූර්ණයක්ම, ලුපය අවසාන මුලාවයට වෙත පෙන තුරු වික් වික් පුනරාවර්තනයේදී \$array විවෘතයට පවරනු ලැබේ.

Ex.

```
<?php  
$colors = array("red", "green",  
"blue", "yellow");  
  
foreach ($colors as $color) {  
    echo "$color <br>";  
}  
?  
>
```

```
red  
green  
blue  
yellow
```

Associative Array Example:

```
<?php  
  
$ages = array("Peter"=>"35",  
"Ben"=> "37", "Joe"=> "43");  
  
foreach ($ages as $name => $age){  
    echo "$name is $age years  
old.<br>";  
}  
  
?  
>
```

Peter is 35 years old.
Ben is 37 years old.
Joe is 43 years old.

Nested Loops

You can nest loops inside each other.
ඔබට ඉප තළ ඉප කැඳුම් කළ හැක.

This is useful when working with multidimensional arrays or when you need to perform complex iterations.

Multidimensional arrays සමඟ ක්‍රියා කරන විට හෝ සකිරීතු ප්‍රහරුවර්තන සිදු කිරීමට අවශ්‍ය විට මෙය ප්‍රයෝගනවත් වේ.

Ex.

```
<?php  
  
for ($i = 1; $i <= 3; $i++) {  
    echo "Outer Loop iteration  
$i<br>;  
  
    for ($j = 1; $j <= 3; $j++) {  
        echo "- Inner iteration  
$j<br>;  
    }  
    echo "<br>;  
}  
  
?>
```

Outer Loop iteration 1

- Inner iteration 1
- Inner iteration 2
- Inner iteration 3

Outer Loop iteration 2

- Inner iteration 1
- Inner iteration 2
- Inner iteration 3

Outer Loop iteration 3

- Inner iteration 1
- Inner iteration 2
- Inner iteration 3

Break and Continue

break

Used to exit the loop immediately.
ඉපයෙන් වහාම ඉවත් වීමට හාවිතා කරයි.

continue

Used to skip the current iteration and proceed to the next iteration of the loop.
වත්මන් ප්‍රහරුවර්තනය මග හැර ඉපයේ ඊලග ප්‍රහරුවර්තනය වෙත ගාමට හාවිතා කරයි.

Ex. - **break**

```
<?php  
for ($x = 0; $x < 10; $x++) {  
    if ($x == 5) {  
        break;  
    }  
    echo "The number is: $x <br>;  
}  
?>
```

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4

Ex. - **continue**

```
<?php  
for ($x = 0; $x < 7; $x++) {  
    if ($x == 5) {  
        continue;  
    }  
    echo "The number is: $x <br>;  
}  
?>
```

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 6

PHP Functions

PHP හිත

Functions are blocks of code that perform specific tasks and can be called upon whenever needed.

හිත යනු නිශ්චිත කාර්යයන් ඉටු කරන කේත කොටස් වන අතර අවශ්‍ය විටෙක ව්‍යා ඇමතීමට හැක.

They help in organizing code, making it reusable, and improving readability and maintainability.

ව්‍යා කේතය සංවිධානය කිරීමට, නැවත හාවිතා කළ හැකි බවට පත් කිරීමට සහ කියවීමේ හැකියාව සහ නඩත්තු කිරීමේ හැකියාව වැඩි දියුණු කිරීමට උපකාර වේ.

Defining and Calling Functions

හිත නිර්වචනය කිරීම සහ ඇමතීම

Basic Function Definition

To define a function in PHP, use the `function` keyword, followed by the function name, parentheses `()`, and a block of code within `{}`.

PHP හි තිතයක් නිර්වචනය කිරීම සඳහා, `function` මූල පදය හාවිතා කරන්න, ඉන්පසු තිත නාමය, වර්ගන් `()` සහ `{}` තළ කේත කොටසක් හාවිතා කරන්න.

Syntax

```
<?php  
  
function functionName() {  
    // code to be executed  
}  
  
?>
```

Ex.

```
<?php  
function greet() {  
    echo "Hello, world!";  
}
```

```
// Calling the function  
greet();
```

```
?>
```

Hello, world!

Functions can be categorized into four types based on parameters and the return value.

පරාමිති සහ `return` අයය මත පදනම්ව තිත වර්ග හතරකට කාණ්ඩ කළ හැකිය.

1. Functions with Only Parameters

These functions accept input values but do not return a value.

මෙම තිතයන් ආදාළ අයයන් පිළිගන්නා නමුත් අයයක් ආපසු ලබා නොදේ.

They perform operations based on the provided parameters.

සපයන ලද පරාමිතින් මත පදනම්ව ව්‍යා මෙහෙයුම් සිදු කරයි.

Ex.

```
<?php  
function greet($name) {  
    echo "Hello, $name!";  
}
```

```
greet("Alice");
```

```
?>
```

Hello, Alice!

2. Functions with Only Return Values

These functions do not accept any parameters but return a value after execution.

මෙම ශ්‍රීතයන් කිසිදු පරාමිතියක් නොපිළිගන්නා නමුත් වියාත්මක කිරීමෙන් පසු අගයක් ආපසු ලබා දෙයි.

Ex.

```
<?php  
  
function getGreeting() {  
    return "Hello, world!";  
}  
  
echo getGreeting();  
  
?>
```

Hello, world!

3. Functions with Parameters and Return Values

Functions with both parameters and return values accept input values, perform operations, and return a result.

පරාමිති සහ ආපසු අගයන් දෙකම සහිත ශ්‍රීත ආදාන අගයන් පිළිගනීවත් ප්‍රතිචලයක් ආපසු ලබා දෙන්නේවත් නැත.

Ex.

```
<?php  
  
function add($a, $b) {  
    return $a + $b;  
}  
  
$result = add(3, 4);  
echo "Result is $result.";  
  
?>
```

Result is 7.

4. Functions without Parameters and Return Values

Functions without parameters and return values neither accept input values nor return a result.

පරාමිති සහ ආපසු අගයන් නොමැති ශ්‍රීත ආදාන අගයන් පිළිගන්නේවත් ප්‍රතිචලයක් ආපසු ලබා දෙන්නේවත් නැත.

They perform predefined tasks.

එවා පෙර නිර්වචනය කරන මද කාර්යයන් ඉටු කරයි.

Ex.

```
<?php  
  
function greet() {  
    echo "Hello, world!";  
}  
  
greet();  
  
?>
```

Hello, world!

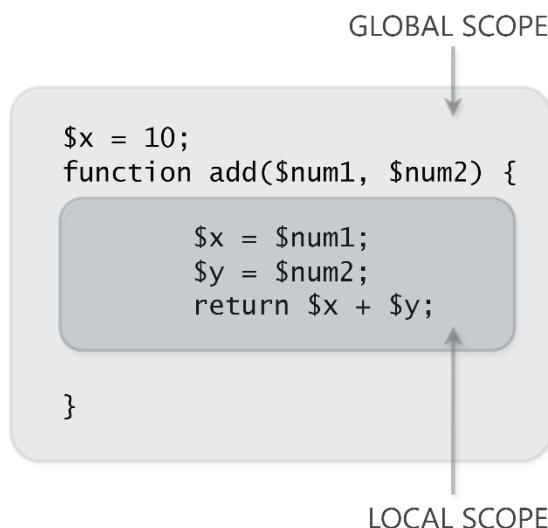
Variable Scope

Variables inside a function have local scope, meaning they can only be accessed within that function. They are known as **local variables**.

ශ්‍රීතයක් තුළ ඇති විවලුණයන්ට ස්ථානිය පරියක් ඇත, විනම් එවාට ප්‍රවේශ විය හැක්කේ විම ක්‍රිතය තුළදී පමණි. එවා ස්ථානිය විවලුණයන් නම් වේ.

Variables outside of functions have global scope and cannot be accessed inside functions unless explicitly specified. They are called **global variables**.

ශ්‍රීතයෙන් පිටත විවලුණයන්ට ගෝලීය පරියක් ඇති අතර පැහැදිලිව සඳහන් කර ඇත්තැම් මිස ක්‍රිත තුළට ප්‍රවේශ විය නොහැක. එවා ගෝලීය විවලුණයන් නම් වේ.



Ex.

```
<?php
$globalVar = "I'm a global variable";

function testScope() {

    $localVar = "I'm a local variable";
    echo $localVar; // Accessible
    echo "<br>";

    // echo $globalVar;
    // Error: Undefined variable
}

testScope();
echo $globalVar; // Accessible

// echo $localVar;
// Error: Undefined variable

?>
```

I'm a local variable
I'm a global variable

\$x in the global scope and \$x in the local scope are two different variables.

ගෝලීය පරියේ \$x සහ ස්ථානිය පරියේ \$x යනු වෙනස් විවලුණ දෙකකි.

The global Keyword

To access global variables within a function, the `global` keyword can be used.

මෙයක් තුළදී ගෝලිය විවෘතයන් වෙත ප්‍රවේණ වීමට `global` මුළු පදය භාවිත කළ හැක.

```
<?php  
$globalVar = "I'm a global variable";  
  
function testScope() {  
  
    global $globalVar;  
    echo $globalVar; // Accessible  
}  
  
testScope();  
  
?>
```

I'm a global variable

The \$GLOBALS Array

The `$GLOBALS` array is a superglobal associative array in PHP that contains references to all variables available in the global scope of the script.

`$GLOBALS` අර්ථ යනු කේතයේ ගෝලිය පරිදේ ඇති සියලුම විවෘතයන් සඳහා යොමු ඇති ප්‍රාග්ධනය සියලුම ප්‍රාග්ධනය විකති.

It provides a way to access global variables from anywhere in the script, including within functions and methods where global variables are not normally accessible.

මිය සාමාන්‍යයෙන් ගෝලිය විවෘතයන් වෙත ප්‍රවේණ විය නොහැකි නිත සහ methods ඇතුළුව, කේතයේ සිහ්‍යම තැනක සිට ගෝලිය විවෘත වෙත ප්‍රවේණ වීමට මාරුගයක් සපයයි.

NOTE: The keys in the `$GLOBALS` array are the names of the global variables, and the values are the contents of those variables.

`$GLOBALS` අර්ථවේ keys යනු ගෝලිය විවෘතවල නම් වන අතර විම keys වල අයන් යනු විවෘතවල අන්තර්ගතය වේ.

It allows functions and methods to access and modify global variables without explicitly using the `global` keyword.

`global` මුළු පදය භාවිත නොකර ගෝලිය විවෘතයන් වෙත ප්‍රවේණ වීමට සහ වෙනස් කිරීමට විය නිත සහ methods වලට ඉඩ දෙයි.

```
<?php  
  
$globalVar = "I'm a global variable";  
  
function testGlobalAccess() {  
  
    echo $GLOBALS['globalVar'];  
}  
  
testGlobalAccess();  
  
?>
```

I'm a global variable

In this example, the function `testGlobalAccess` accesses the global variable `$globalVar` using the `$GLOBALS` array.

මෙම උදාහරණයේදී, `testGlobalAccess` නිතය `$GLOBALS` අර්ථ භාවිතයෙන් `$globalVar` ගෝලිය විවෘතයට ප්‍රවේණ වේ.

Modifying Global Variables

```
<?php  
  
$globalVar = "I'm a global variable";  
  
function modifyGlobal() {  
    echo $GLOBALS['globalVar'];  
    echo "<br>";  
    $GLOBALS['globalVar'] = "I've  
    been modified!";  
}  
  
modifyGlobal();  
echo $globalVar;  
?>
```

I'm a global variable
I've been modified!

Using \$GLOBALS with Multiple Global Variables

ගෝලීය විවෘතයන් කිහිපයක් සමඟ \$GLOBALS හාඩිතා දිරීම

```
<?php  
  
$var1 = "First variable";  
$var2 = "Second variable";  
  
function showGlobals() {  
  
    echo $GLOBALS['var1'];  
    echo "<br>";  
    echo $GLOBALS['var2'];  
}  
  
showGlobals();  
  
?>
```

First variable
Second variable

In this example, the function `showGlobals` accesses multiple global variables using the `$GLOBALS` array.

මෙම උදාහරණයේදී, `showGlobals` ලිඛිතය `$GLOBALS` අරාව හාඩිතයෙන් බහු ගෝලීය විවෘතයන් වෙත ප්‍රවේශ වේ.

Avoiding Conflicts with Local Variables

ස්ථානික විවෘතයන් සමඟ ඇති වන ගැටලු වළක්වා ගැනීම

The `$GLOBALS` array helps prevent conflicts between local and global variables by providing a clear distinction and access mechanism.

`$GLOBALS` අරාව පැහැදිලි වෙනසක් සහ ප්‍රවේශ යාන්ත්‍රණයක් ලබා දීමෙන් ස්ථානික සහ ගෝලීය විවෘතයන් අතර ගැටලු වළක්වා ගැනීමට උපකාරී වේ.

```
<?php  
  
$var = "Global variable";  
  
function testConflict() {  
    $var = "Local variable";  
    echo $var; echo "<br>";  
    echo $GLOBALS['var'];  
}  
  
testConflict();  
  
?>
```

Local variable
Global variable

In this example, the function `testConflict` demonstrates the distinction between a local variable and a global variable with the same name, using the `$GLOBALS` array to access the global variable.

මෙම උදාහරණයේ දී, `testConflict` ලිඛිතය මගින් ගෝලීය විවෘතයට ප්‍රවේශ වීමට `$GLOBALS` අරාව හාඩිත කරමින් ස්ථානික විකම නම සහිත විවෘතයක් සහ ගෝලීය විවෘතයක් අතර වෙනස පෙන්වුම් කරයි.

Default Parameters

You can specify default values for function parameters.

මෙය සිතු පරාමිති සඳහා පෙරහිම් අගයන් නියම කළ හැක.

If a new argument for a parameter with a default value is not provided during the function call, that default value will be used.
මෙහි ඇමතුම අතරතුර පෙරහිම් අගයක් සහිත පරාමිතියකට විකළ සපයා තොමැති නම්, විම පෙරහිම් අගය භාවිතා වේ.

Ex.

```
<?php
function greet($name = "Guest") {
    echo "Hello, $name!";
}

greet();
greet("Alice");
?>
```

Hello, Guest!
Hello, Alice!

Recursive Functions

A recursive function is a function that calls itself.

ප්‍රතිචාර විසින් මෙය තමන් විසින්ම කැදුව ගෙන්නා සිතුයකි.

Recursion is useful for tasks that can be divided into similar subtasks.

සමාන උප කාර්යයන් වලට බෙදිය හැකි කාර්යයන් සඳහා ප්‍රතිචාර විසින් ප්‍රශ්නවත් වේ.

Ex. Factorial Calculation

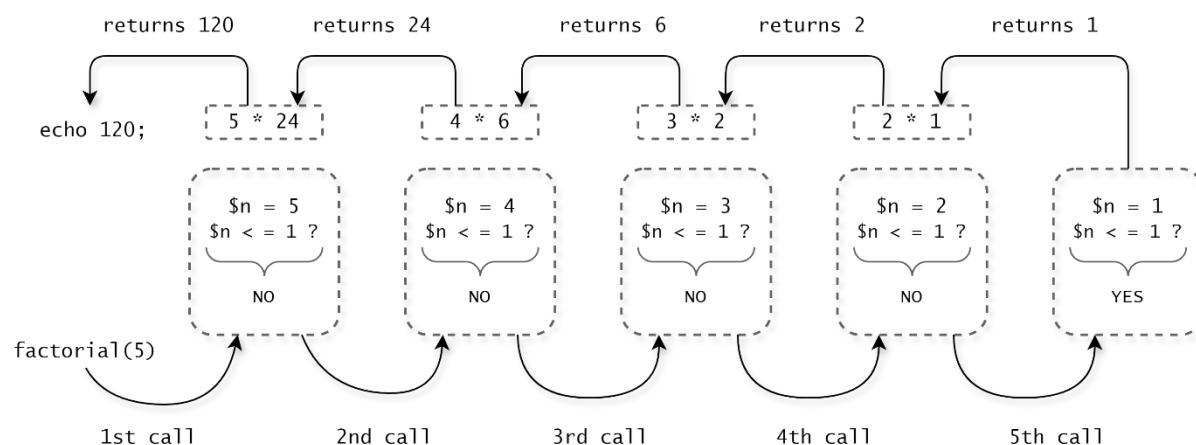
```
<?php

function factorial($n) {
    if ($n <= 1) {
        return 1;
    } else {
        return $n * factorial($n - 1);
    }
}

echo factorial(5);
?>
```

120

How Recursive functions work



Consider each box as a function call

Commonly used built-in functions

1. String Functions

strlen()

Returns the length of a string.

தன்னுவக டிட லூ டேடி.

```
<?php  
echo strlen("Hello");  
?>
```

5

str_replace()

Replaces all occurrences of a search string with a replacement string.

செவும் தன்னுவக கீயலும் கீடுவீம் பூதிச்சீபந
தன்னுவகின் பூதிச்சீபநங் கரடி.

```
<?php  
$txt = "Hello world!";  
echo str_replace("Hello", "Hi",  
"$txt");  
?>
```

Hi world!

strpos()

Finds the position of the first occurrence of a substring in a string.

தன்னுவக அதிக ஏப் தன்னுவக பலம் கீடுவீமே
பீதிரீம் சொய வெ.

```
<?php  
echo strpos("Hello world!", "world");  
?>
```

6

substr()

Returns a portion of a string.

தன்னுவக கோவக்க லூ டேடி.

```
<?php  
echo substr("Hello world!", 6, 5);  
?>
```

world

2. Array Functions

count()

Counts all elements in an array or something in an object.

அருவக ஹே வகீஷுவக அதிக கீயலும் மூலவயவ
ஞாங் கரடி.

```
<?php  
$arr = array(1, 2, 3);  
echo count($arr);  
?>
```

3

array_merge()

Merges one or more arrays.

அரு விகக் ஹே கிதிபயக் வீகாவிட்டீ கரடி.

```
<?php  
  
$array1 = array(1,2,3);  
$array2 = array(4,5,6);  
$result = array_merge($array1,  
$array2);  
var_dump($result);  
  
?>
```

array (size=6)

| |
|------------|
| 0 => int 1 |
| 1 => int 2 |
| 2 => int 3 |
| 3 => int 4 |
| 4 => int 5 |
| 5 => int 6 |

array_push()

Pushes one or more elements onto the end of an array.

මුළුවයට විනක් හෝ තිහිපයක් අරාවක අවසානයට තල්ල කරයි.

```
<?php
```

```
$arr = array("apple", "banana");
array_push($arr, "orange", "grape");
var_dump($arr);
```

```
?>
```

```
array (size=4)
0 => string 'apple' (length=5)
1 => string 'banana' (length=6)
2 => string 'orange' (length=6)
3 => string 'grape' (length=5)
```

array_keys()

Returns all the keys or a subset of the keys of an array.

අරාවක සියලුම යතුරු හෝ යතුරුවම උප කුලකයක් ආපසු ලබා දෙයි.

```
<?php
$arr = array("a" => "apple",
"b" => "banana");
var_dump(array_keys($arr));
?>
```

```
array (size=2)
0 => string 'a' (length=1)
1 => string 'b' (length=1)
```

3. Mathematical Functions

abs()

Returns the absolute value of a number.
සංඛ්‍යාවක නිරපේක්ෂ අගය ලබා දෙයි.

```
<?php
```

```
echo abs(-4.2);
```

```
?>
```

4.2

round()

Rounds a floating point number.
ඉපිලෙන ලක්ෂණ සංඛ්‍යාවක් වටයයි.

```
<?php
```

```
echo round(3.6);
```

```
?>
```

4

rand()

Generates a random integer.

සයළුම් පූර්ණ සංඛ්‍යාවක් ජනනය කරයි.

```
<?php
```

```
echo rand(1, 10);
```

```
// Outputs: a random number
```

```
// between 1 and 10
```

```
?>
```

sqrt()

Returns the square root of a number.

සංඛ්‍යාවක වර්ගමුලය ලබා දෙයි.

```
<?php
```

```
echo sqrt(16);
```

```
?>
```

4

Other commonly used functions

අනෙකුත් බහුලව හාවතා ශ්‍රීත

isset()

The isset() function in PHP is used to determine if a variable is set and is not NULL.

(ඡ්‍යාප) හි ඇති isset() ශ්‍රීතය විවෘතයක් සකසා තිබේද සහ NULL නොවේද යන්න තීරණය කිරීමට හාවතා කරයි.

Returns true if the variable exists and is not NULL.

විවෘතය පවතින්නේ නම් සහ NULL නොවේ නම් සහ වේ.

Returns false if the variable does not exist or is NULL.

විවෘතය නොමැති නම් හෝ NULL නම් අසත්‍ය වේ.

```
<?php
$variable = "Hello, World!";
if (isset($variable)) {
    echo "Variable is set";
} else {
    echo "Variable is not set";
}
?>
```

Variable is set

empty()

Determines whether a variable is empty.
විවෘතයක් හිස් ද යන්න තීරණය කරයි.

```
<?php
$var = "";
echo empty($var);
?>
```

1

print_r()

Prints human-readable information about a variable.

විවෘතයක් ගැන මිතිසුන්ට කියවීය හැකි තොරතුරු මුද්‍රණය කරයි.

```
<?php
$arr = array("apple", "banana");
print_r($arr);
?>
```

Array ([0] => apple [1] => banana)

var_dump()

Dumps information about a variable.
විවෘතයක් පිළුබඳ තොරතුරු බිම්පේ කරයි.

```
<?php
$var = "Hello";
var_dump($var);
?>
```

string 'Hello' (length=5)

die() or exit()

Outputs a message and terminates the current script.

පණිවේඩයක් ප්‍රතිදානය කර වත්මන් කේතය වික අවසන් කරයි.

```
<?php
die("Script terminated");
?>
```

Script terminated

implode()

The `implode()` function (also known as `join()`) is used to join elements of an array into a single string.

අරාවක මුලාවයට තහි තන්තුවකට සම්බන්ධ කිරීමට `implode()` ශ්‍රීතය (`join()`) ලෙසද හැඳින්වේ) හාවිතා වේ.

Syntax

```
string implode(string $separator,  
array $array)
```

Example

```
<?php  
$array = ['apple', 'banana'];  
$string = implode(", ", $array);  
echo $string;  
?>
```

explode()

The `explode()` function is used to split a string by a specified delimiter into an array.

නිශ්චීවෙන පරීක්ෂෙමකයක් මගින් තන්තුවක් අරාවකට බෙදීමට `explode()` ශ්‍රීතය හාවිතා කරයි.

Syntax

```
array explode(string $separator,  
string $string)
```

Example

```
<?php  
$string = "apple, banana, cherry";  
$array = explode(", ", $string);  
print_r($array);  
?>
```

Array ([0] => apple [1] => banana
[2] => cherry)

static keyword

The `static` keyword in PHP allows a local variable to retain its value across multiple calls to that function.

PHP හි මූල පදය ස්ථානීය විවෘතයකට වීම ශ්‍රීතයට ආමතුම් කිහිපයක් හරහා වහි අගය රුද්‍රවා ගැනීමට ඉඩ සලසයි.

This means that the value of the variable is memorized between function calls.

මෙයින් අදහස් කරන්නේ විවෘතයේ අගය ශ්‍රීත ඇමතුම් අතර මතක තබා ගෙන ඇති බවයි.

```
<?php  
function counter() {  
    static $count = 0;  
    $count++;  
    echo $count;  
}
```

```
counter();  
counter();  
counter();  
?>
```

1
2
3

Handling HTML forms with PHP

PHP සමඟීන් HTML පෝරම හැකිරවීම

HTML forms are used to collect user input and submit it to a server for processing.

HTML පෝරම හාවිතා කරනුයේ පරිශීලක ආදාන විකතු කර විය සැකසීම සඳහා සේවාදායකයෙක පරිගණකයකට යැවීමටයි.

The form consists of five input fields for personal information: First Name, Last Name, Date of Birth, Email id, and Mobile Number. Each field has a corresponding grey input box. At the bottom of the form are two buttons: 'SUBMIT' on the left and 'RESET' on the right.

They are essential for creating interactive web applications, such as login forms, search boxes, and data entry forms.

පිවිසුම් පෝරම, සේවුම් පෝරිටි සහ දත්ත ඇතුළත් කිරීමේ පෝරම වැනි අන්තර්ත්‍යාකාර වෙබ් යොදුම් නිර්මාණය කිරීම සඳහා ව්‍යාපෘතිවාසික වේ.

Benefits of using HTML forms

HTML පෝරම හාවිතා කිරීමේ වාසි

• **Access Control**

ප්‍රවේශ පාලනය

Forms act as an intermediary, preventing users from directly interacting with the database.

පෝරම අතරමැදියෙකු ලෙස ක්‍රියා කරයි, පරිශීලකයින් දත්ත සමුළුය සමඟ සංප්‍රව්‍ය අන්තර් ක්‍රියා කිරීම වළක්වයි.

This reduces the risk of unauthorized access and malicious activities such as SQL injection.

මෙමගින් SQL injection වැනි අනවසර ප්‍රවේශයන් සහ අනිශ්චිත ක්‍රියාකාරකම් වල අවදානම ඇතුළත් ඇති කරයි.

• **Validation**

වලංගු කිරීම

HTML forms and server-side scripts can validate and sanitize user input to ensure that it meets specific criteria before it is entered into the database, preventing harmful data from being stored.

HTML පෝරාම සහ සේවාදායක පාර්ශ්ව කේත වලට පරිශීලක ආදානය වලංගු කිරීමට සහ විය දත්ත සමුළුයට ඇතුළු කිරීමට පෙර විය නිශ්චිත නිර්ණායක සපුරාලන බවට සහතික කිරීමට සහ හානිකර දත්ත ගබඩා කිරීම වළක්වයි.

• **Consistent Data Entry**

ස්ථාවර දත්ත ඇතුළත් කිරීම

Forms can enforce specific formats and data types, ensuring that the data entered by users is consistent and structured correctly.

පරිශීලකයන් විසින් ඇතුළත් කරන ලද දත්ත ස්ථාවර සහ නිවැරදිව විශුහගගත බව සහතික කරමින්, පෝරමවලට නිශ්චිත ආකෘති සහ දත්ත වර්ග බලාත්මක කළ හැක.

- **Error Checking**
දේශ පරික්ෂා කිරීම

Forms can provide immediate feedback to users about incorrect or incomplete data, reducing the likelihood of errors in the database.

දත්ත සමුදායේ දේශ ඇතිවිමේ සම්භාවිතාව අඩු කරමින්, වැරදි හෝ අසම්පූර්ණ දත්ත පිළිබඳව පරිශීලකයින්ට ක්ෂේත්‍රීක ප්‍රතිචාර පෝරම මගින් ලබා දිය හැක.

- **User Experience**
පරිශීලක අන්දැකිම්

Forms can be designed to be user-friendly, making it easier for users to enter data correctly.

පෝරම පරිශීලක-නිරකාම් ලෙස නිර්මාණය කළ හැකි අතර, පරිශීලකයින්ට දත්ත නිවැරදිව ඇතුළත් කිරීම පහසු කරයි.

They can include dropdowns, checkboxes, and other input types to simplify data entry. දත්ත ඇතුළත් කිරීම සරල කිරීම සඳහා එක්ව dropdows, checkboxes සහ වෙනත් ආභාහ වර්ග ඇතුළත් කළ හැක.

Forms can include instructions, tooltips, and validation messages to guide users through the data entry process.

දත්ත ඇතුළත් කිරීමේ ක්‍රියාවලිය හරහා පරිශීලකයින්ට මග පෙන්වීම සඳහා උපදෙස්, මෙවලම් ඉග් සහ වලංගු කිරීමේ පණිව්‍ය පෝරමවලට ඇතුළත් කළ හැක.

- **Controlled Data Flow**
ආලිත දත්ත ප්‍රවාහය

Forms allow developers to control the flow of data, ensuring that it is processed and stored correctly.

පෝරම මගින් සංවර්ධකයින්ට දත්ත ගොනාම පාලනය කිරීමට ඉඩ සලසයි, විය නිවැරදිව සකසන ලද සහ ගධ්‍ය කර ඇති බව සහතික කරයි.

This includes logging submissions, sending confirmation emails, or triggering other processes.

Submit කිරීම් log කිරීම, තහවුරු කිරීමේ රුමේල් යැවීම හෝ වෙනත් ක්‍රියාවලි සිදුකිරීම මෙයට ඇතුළත් වේ.

- **Data Preprocessing**
දත්ත පෙර සැකසුම්

Before data is entered into the database, it can be preprocessed in various ways, such as formatting dates, normalizing text, or aggregating data.

දත්ත සමුදායට දත්ත ඇතුළු කිරීමට පෙර, විය දින ආකෘතිකරණය කිරීම, පෙළ සාමාන්‍යකරණය කිරීම හෝ දත්ත විකුත් කිරීම වැනි විවිධ ආකෘතික් පෙර සැකසීය හැක.

Basic Structure of an HTML Form

HTML පෝරමයක මූලික ව්‍යුහය

<form> tag

An HTML form is defined using the <form> element, which can contain various types of input elements.

විවිධ ආකාරයේ ආදාන මූලුවන අඩංගු විය හැකි <form> මූලාවයට හාවතයෙන් HTML පෝරමයක් අර්ථ උක්වනු ලබයි.

A form can contain various types of input elements, buttons, and other elements that enable users to submit data to a web server.

පෝරමයක විවිධ ආකාරයේ ආදාන මූලාවයට, බොත්තම් සහ වෙබ් සේවාදායකයකට දත්ත ඉදිරිපත් කිරීමට පරිශ්ලකයින්ට හැකි වෙනත් මූලාවයට අඩංගු විය හැක.



Write an excerpt (optional)

Example:

```
<body>
```

```
<form action="submit.php" method="post" autocomplete="on">
    <!-- Form elements go here --&gt;
&lt;/form&gt;</pre>
```

```
</body>
```

<form> Attributes

action

Specifies the URL to which the form data will be submitted.

පෝරම දත්ත ඉදිරිපත් කරන URL වික සඳහන් කරයි.

method

Specifies the HTTP method to use when submitting the form (get or post).

පෝරමය ඉදිරිපත් කිරීමේදී හාවතා කළ යුතු HTTP තුළය සඳහන් කරයි (get හෝ post).

(Action & method attributes will be further discussed later.)

(Action සහ method යන උපලක්ෂණ පසුව තවදුරටත් සාකච්ඡා කෙරේ.)

autocomplete

Specifies whether the form should have autocomplete on or off.

පෝරමය ස්ට්‍රෑට් සම්පූර්ණ කිරීම සඳහා හෝ අක්‍රිය කළ යුතු යන්න සඳහන් කරයි.

<input> tag

The <input> element is the most widely used form element, supporting various input types such as text, password, email, and more.

<input> මුළුවයට යනු පෙළ, මුරපද, විශ්‍යත් තැපෑල සහ තවත් බොහෝ ආදාන ව්‍යුහ සඳහා සහය දක්වන ව්‍යාපෘති නිස්ථාව වන පෝරම මුළුවයට වේ.

<input> Attributes

type

Specifies the type of input (Ex. text, password, email, radio, checkbox, submit, etc.). (will be discussed later)

ආදාන ව්‍යුහ සඳහන් කරයි (ලභ. පෙළ, මුරපද, විශ්‍යත් තැපෑල, රේඛියා බොහෝතම්, පීරික්සුම් කොටු, submit කිරීම ආදිය).

name

Specifies the name of the input, which is used to identify the form data after submission.

Submit කිරීමෙන් පසු පෝරම දත්ත හඳුනා ගැනීමට නාවිතා කරන ආදානයේ නම සඳහන් කරයි.

value

Specifies the initial value of the input.

ආදානයේ ආරම්භක අගය තියම කරයි.

placeholder

Provides a hint to the user about what to enter in the input.

ආදානය තුළ ඇතුළත් කළ යුතු දේ ගැන පරිශීලකයාට ඉටියක් සහයයි.

Email

name@example.com



Placeholder

required

Specifies that the input must be filled out before submitting the form.

පෝරමය කිරීමට පෙර ආදානය පීරිවීම අනිවාර්ය බව සඳහන් කරයි.

Name:*

Submit

Please fill out this field

Please enter the following information to create your account.

| | |
|------------------|---|
| First name | * |
| Last name | * |
| Email address | * |
| Password | * |
| Confirm password | * |

Required

readonly

Specifies that the input field is read-only. ආදාන ක්ෂේත්‍රය කියවීමට පමණක් බව සඳහන් කරයි.

| | |
|-------------|--------------------|
| First Name* | Paul |
| Last Name* | Allen |
| Email* | pallen@example.com |
| Phone | 1234567890 |

disabled

Disables the input field. ආදාන ක්ෂේත්‍රය අක්‍රිය කරයි.

Value (disabled)

id

Specifies a unique identifier for the input.
අදාළතය සඳහා අනන්‍ය හඳුනාගැනීමක් නියම කරයි.

The name attribute in detail

ලුප්ලක්ෂණය විසින් රුමක්ව

The name attribute in HTML form elements is a fundamental component for web forms, playing a crucial role in the form submission process, server-side data handling, and client-side scripting.

HTML පෝරම මූලාධාරවල name ලුප්ලක්ෂණය වෙති පෝරම සඳහා මූලික අංශයක් වන අතර, පෝරමය submit කිරීමේ ක්‍රියාවලියේදී, සේවාදායක අන්තයේ දත්ත හැසිරවීමේදී, සහ සේවාලාභී පාර්ශ්වයේ ස්ක්‍රීඩ්‍රේ කිරීමේදී තීරණාත්මක කාර්යනාරයක් ඉටු කරයි

What is the name attribute?

name ලුප්ලක්ෂණය යනු කුමක්ද?

The name attribute is used to specify a unique identifier for form elements.

පෝරම මූලාධාරය සඳහා අනන්‍ය හඳුනාගැනීමක් නියම කිරීමට name ලුප්ලක්ෂණය හාවිතා කරයි.

This identifier is used when data is submitted to the server, allowing the server-side script to retrieve and process the form data.

මෙම හැඳුනුම්කාරකය පෝරම දත්ත බඟ ගැනීමට සහ සැකසීමට සේවාදායක අන්තයේ කේතයට ඉඩ සලසුම්හි සේවාදායකය වෙත දත්ත submit කරන විට හාවිතා කරනු ලැබේ.

Syntax

```
<input type="text" name="username">
```

Importance and Functions of the name Attribute

When a form is submitted, the browser sends the form data as **key-value** pairs to the server.

පෝරමයක් submit කළ විට, අනර්ස්පුව මගින් පෝරම දත්ත යතුරු-අගය යුගල මෙස සේවාදායකයට යවයි.

The name attribute of each form element acts as the key in these pairs. The value entered by the user is the value in the key-value pair.

එක් එක් පෝරම මූලාධාරයේ name ලුප්ලක්ෂණය මෙම යුගලට යතුර මෙස ක්‍රියා කරයි. පරිශ්ලකා විසින් ඇතුළත් කරන ලද අගය යතුරු-අගය යුගලයේ අගය වේ.

Example:

```
<body>
<form action="submit.php" method="post">
  <input type="text" name="username" placeholder="Username">
  <input type="password" name="password" placeholder="Password">
  <button type="submit">Login</button>
</form>
</body>
```

When this form is submitted, the data sent to the server might look like this:

මෙම පෝරමය submit කළ විට, සේවාදායකය වෙත යවන ලද දත්ත මෙමෙස පෙනීය හැකිය:

username=johndoe&password=securepassword

<label> tag

The <label> element defines a label for an <input> element.

<label> මූලවයට <input> මූලවයටක් සඳහා ලේඛුයක් අර්ථ දක්වයි.

Clicking on the label will focus/select the corresponding input.

ලේඛුය මත ක්මික් කිරීමෙන් අදාළ ආදානය අවධානය යොමු කරයි/තෝරා ඇත.

for Attribute

Specifies which form element a label is bound to by referring to the id of that particular element.

ලේඛුයක් බැඳී ඇත්තේ කුමන පෝරම මූලවයටද යන්න විම මූලවයටයේ අනන් හඳුනාගැනීම වෙත යොමු කරමින් සඳහන් කරයි.

Example:

```
<body>
<form> <pre>

    <label for="username">Username:</label>
    <input type="text" id="username" name="u_name">

    <label for="password">Password:</label>
    <input type="password" id="password" name="pwd">

</pre> </form>
</body>
```



<fieldset> and <legend> tags

<fieldset>

The <fieldset> tag is used to create a grouping of form elements.

පෝරම මූලාවයට සම්බන්ධක් සඳහා <fieldset> උපයනය හාවතා කරයි.

It helps in organizing the form elements into logical sections, making the form easier to understand and use.

විය පෝරමය මූලාවය තාර්කික කොටස් වලට සංවිධානය කිරීමට උපකාරී වන අතර, පෝරමය තේරේම් ගැනීමට සහ හාවතා කිරීමට පහසු කරයි.

Example: (<fieldset> & <legend>)

```
<!DOCTYPE html>
<html>
<head>
    <title>Fieldset and Legend Example</title>
</head>
<body>
<form> <pre>
<fieldset>
    <legend>Personal Information</legend>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
</fieldset>

<fieldset>
    <legend>Account Details</legend>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
</fieldset>
<input type="submit" value="Submit">

</pre> </form>
</body>
</html>
```

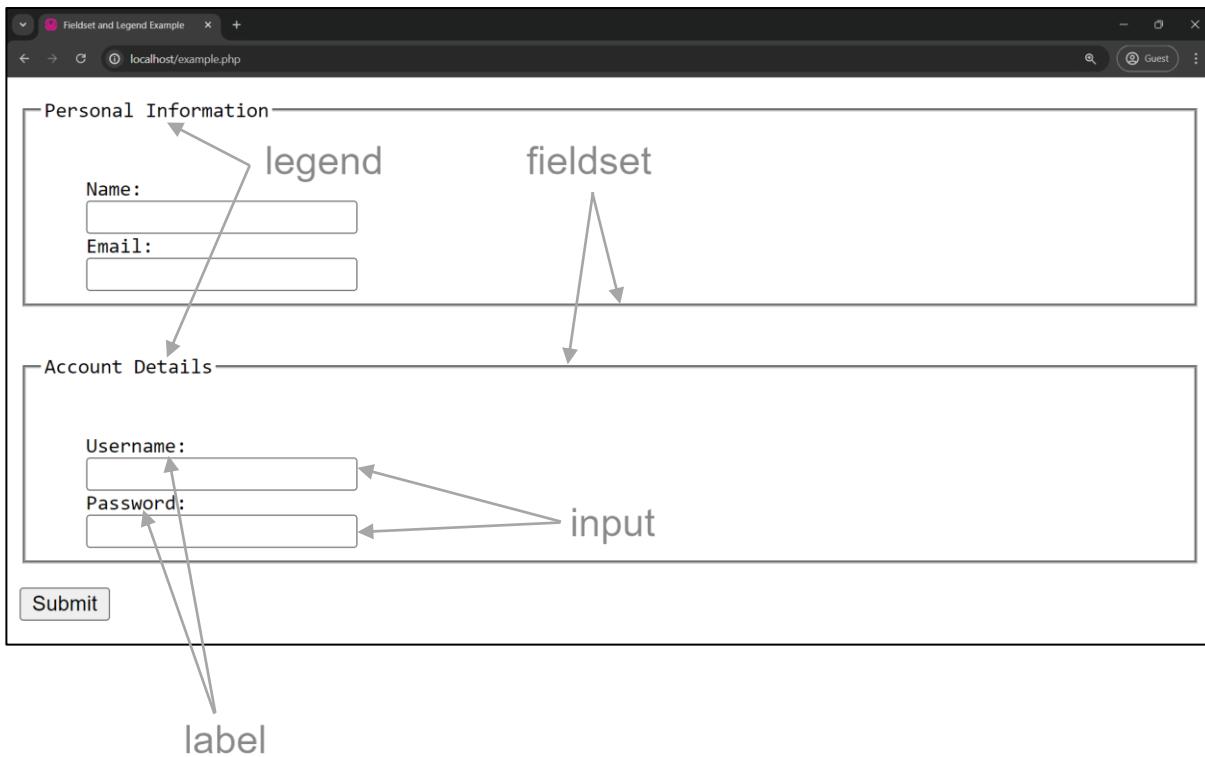
<legend>

The <legend> tag is used to provide a caption or title for the <fieldset>.

<fieldset> සඳහා එකක් හෝ මාතෘකාවක් සැපයීමට <legend> උපයනය හාවතා වේ.

It should be the first element inside the <fieldset> and gives a brief description of the group of form elements.

විය <fieldset> තුළ අති පළමු මූලාවය විය යුතු අතර පෝරම මූලාවය පිළිබඳ කෙටි විස්තරයක් ලබා දෙය



Input Types

අභ්‍යන්තර වර්ග

HTML forms use various input types to collect different kinds of user data.

HTML පෝරම විවිධ ආකාරයේ පරීක්ෂක දත්ත යෙහි කිරීමට විවිධ අභ්‍යන්තර වර්ග හාවිතා කරයි.

1. Text (`<input type="text">`)

Used for single-line text input.

තනි පෝලයේ පෙළ අභ්‍යන්තර සඳහා හාවිතා වේ.

Username:

2. Password (`<input type="password">`)

Used for password input, hides the characters.

මුදලස් අභ්‍යන්තර සඳහා හාවිතා කරයි, අනුවක්ෂණ සැගවයි.

Password:

3. Email (`<input type="email">`)

Used for email addresses, includes validation.

විද්‍යුත් තැපෑලේ ලිපින සඳහා හාවිත කෙරේ, වෘත්තිකරණය ඇතුළත් වේ.

Email: submit

Email: submit

 Please include an '@' in the email address. 'abgmail.com' is missing an '@'.

4. Number (`<input type="number">`)

Used for numeric input, includes validation for numbers.

සංඛ්‍යාත්මක අභ්‍යන්තර සඳහා හාවිත කෙරේ, සංඛ්‍යා සඳහා වෘත්තිකරණය ඇතුළත් වේ

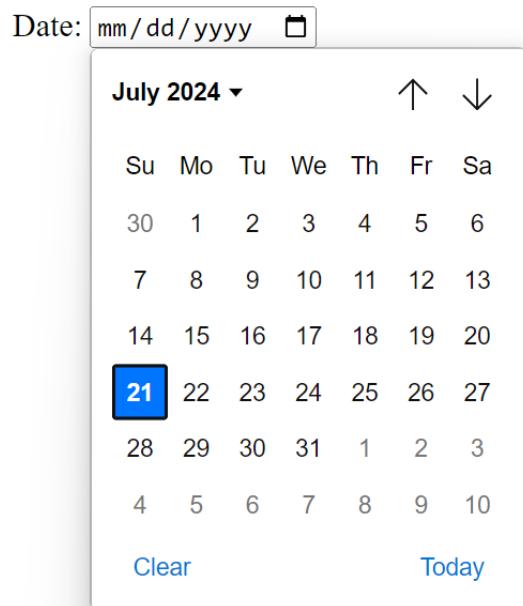
Number:

5. Date (<input type="date">)

Used for date input, provides a date picker.

දින ආදානය සඳහා භාවිතා කරයි, දින තෝර්නයක් සපයයි.

Date: 



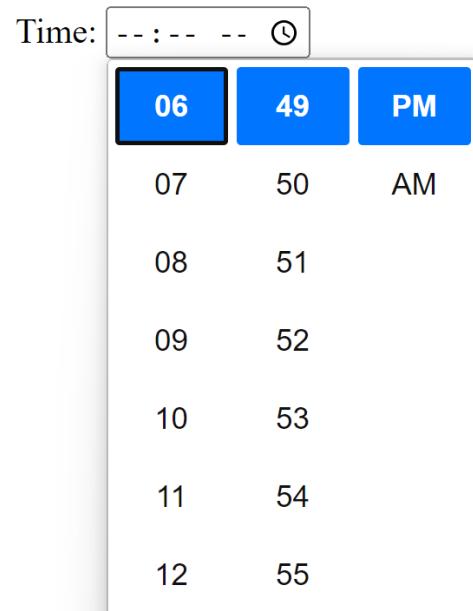
Date: 

6. Time (<input type="time">)

Used for time input, provides a time picker.

කාල ආදානය සඳහා භාවිතා කරයි, කාල තෝර්නයක් සපයයි.

Time: 



Time: 

7. Checkbox

(`<input type="checkbox">`)

Checkboxes allow users to select multiple options from a set.

Checkboxes මගින් පරිශීලකයින්ට කට්ටවුයකින් විකල්ප කිහිපයක් තෝරා ගැනීමට ඉඩ සමසයි.

Each checkbox operates independently, even if they share the same name attribute සෑම checkbox විකක්ම විකම name උපලක්ෂණය බෙදා ගෙන්තද විවා ස්වාධීනව ක්‍රියා කරයි

Subscribe to newsletter:

Receive updates:

Get special offers:

Submit

HTML code for the above page (`<html>`, `<body>` are not mentioned here)

ඉහත පිටුව සඳහා **HTML** කේතය

```
<form action="submit.php" method="post">
    <label for="newsletter">Subscribe to newsletter:</label>
    <input type="checkbox" id="newsletter" name="preferences[]"
           value="newsletter">

    <label for="updates">Receive updates:</label>
    <input type="checkbox" id="updates" name="preferences[]"
           value="updates">

    <label for="offers">Get special offers:</label>
    <input type="checkbox" id="offers" name="preferences[]"
           value="offers">

    <button type="submit">Submit</button>
</form>
```

The name attribute is set to preferences[], creating an array of selected values when the form is submitted.

name උපලක්ෂණය preferences[] වෙත සකසා ඇත, පෝරමය submit කරන විට තෝරාගත් අයෙන්ගෙන් සමන්විත අරාවක් තිබුමාණය කරයි.

Server-side processing (PHP)

```
<?php
$preferences = $_POST['preferences']; // $preferences is an array
foreach ($preferences as $preference) {
    echo htmlspecialchars($preference) . "<br>";
}
?>
```

8. Radio Buttons

(<input type="radio">)

Radio buttons allow users to select only one option from a group.

Radio buttons මගින් පරිශීලකයින්ට කණ්ඩායමකින් විස් විකල්පයක් පමණක් තෝරා ගැනීමට ඉඩ සමස්යි.

Radio buttons with the same name attribute are grouped together, meaning only one button in the group can be selected at a time.

විකම name උපලක්ෂණය සහිත Radio buttons විකට කාණ්ඩාගත කර ඇත, විනම් විම කාණ්ඩායෙන් විස් බොත්තමක් පමණක් වරුකට තෝරා ගත හැකිය.

Male:

Female:

Other:

Submit

HTML code for the above page

ඉහත පිටුව සඳහා HTML කේතය

```
<form action="submit.php" method="post">
    <label for="male">Male:</label>
    <input type="radio" id="male" name="gender" value="male">

    <label for="female">Female:</label>
    <input type="radio" id="female" name="gender" value="female">

    <label for="other">Other:</label>
    <input type="radio" id="other" name="gender" value="other">

    <button type="submit">Submit</button>
</form>
```

The name attribute is set to gender, grouping the radio buttons together.

Radio buttons විකට සමුහගත කරමින් name උපලක්ෂණය gender ලෙසට සකසා ඇත.

Server-Side processing (PHP)

```
<?php
$gender = $_POST['gender']; // $gender contains the selected value
echo "Selected gender: " . htmlspecialchars($gender);
?>
```

Only the selected value will be submitted and accessible.

තෝරාගත් අයය පමණක් submit වන අතර වියට පමණක් ප්‍රවේශ විය හැකිය.

9. File (<input type="file">)

Used for file uploads.

ගොනු උපිගත කිරීම් සඳහා භාවිතා වේ.

File: Choose File No file chosen

10. Hidden (<input type="hidden">)

Used for hidden data, not visible to the user but sent with the form.

සැගවුණු දැන්ත සඳහා භාවිතා වේ, පරිශීලකයට නොපෙනෙන නමුත් පෝරමය සමඟ යවනු ලැබේ.

11. Submit (<input type="submit">)

Used for submitting the form data.

පෝරම දැන්ත submit කිරීම සඳහා භාවිතා වේ.

First Name:

Last Name:

12. Reset (<input type="reset">)

Used for resetting the form to its initial values.

පෝරමය විහි ආරම්භක අගයන් වෙත නැවත සැකසීමට භාවිතා කරයි.

First Name:

Last Name:

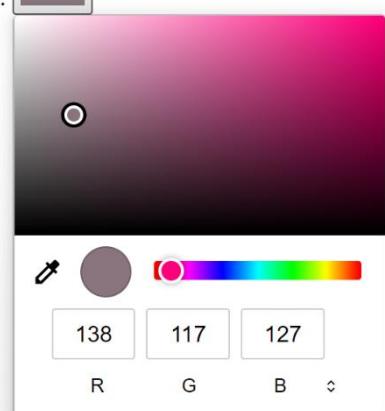
13. Color (<input type="color">)

Used for color input, provides a color picker.

වර්ණ ආදානය සඳහා භාවිතා කරයි, වර්ණ තෝරා ගැනීමක් සපයයි.

Pick a color:

Pick a color:



14. Range (<input type="range">)

Used for numeric input within a range, provides a slider.

පරාසයක් තුළ සංඛ්‍යාත්මක ආදානය සඳහා භාවිතා වේ, ස්ලයිඩරයක් සපයයි.

Range:

15. URL (<input type="url">)

Used for URL input, includes validation.

URL ආදානය සඳහා භාවිතා කරයි, වෘත්ත කිරීම ඇතුළත් වේ.

<select> and <option> tags

The `<select>` and `<option>` HTML tags are used to create drop-down lists in forms. පෝරමවට පහලට වැවෙන ලැයිස්තු සඳහාමට `<select>` සහ `<option>` යන උපුලන නාවිතා කරයි.

These elements allow users to select one or more options from a predefined list. මෙම මූලවයට පරිශීලකයින්ට පූර්ව නිශ්චිත ලැයිස්තුවකින් විකල්ප විකල්ප හෝ කිහිපයක් තෝරා ගැනීමට ඉඩ සළසයි.

<select> Tag

The `<select>` tag creates a drop-down list. It is used to define the list and can contain multiple `<option>` tags inside it.

`<select>` උපුලනය පහළට වැවෙන ලැයිස්තුවක් සාදයි. විය ලැයිස්තුව නිරවචනය කිරීමට නාවිතා කරන අතර විය තුළ `<option>` උපුලන කිහිපයක් අඩංගු විය නැක.

Attributes of <select>

<select> හි උපාක්ෂණ

name

Specifies the name of the drop-down list, which is sent to the server when the form is submitted.

පෝරමය submit කරන විට සේවාදායකය වෙත යවතු ලබන පහළට වැවෙන ලැයිස්තුවේ නම සඳහන් කරයි.

id

Unique identifier for the `<select>` element, useful for linking labels.

`<select>` මූලවය සඳහා අනන්‍ය හඳුනාගැනීම, ලේඛිල සම්බන්ධ කිරීම සඳහා ප්‍රයෝගනවත් වේ.

size

Specifies the number of visible options in the drop-down list.

පහළට වැවෙන ලැයිස්තුවේ පෙනෙන විකල්ප ගණන නියම කරයි.

multiple

Allows multiple selections if set. By default, only one option can be selected.

සකසා ඇත්තේ බහු තෝරීම් වලට ඉඩ දෙයි. පෝරනිමියෙන්, තෝරා ගත භැක්කේ විකල්පයක් පමණි.

disabled

Disables the drop-down list, preventing user interaction

පරිශීලක අන්තර්ඩ්‍රිය වළක්වම්හි පතන ලැයිස්තුව අවුරුදු කරයි

<option> Tag

The `<option>` tag defines an option in the drop-down list.

`<option>` උපුලනය පහළට වැවෙන ලැයිස්තුවේ විකල්පයක් නිරවචනය කරයි.

It is placed inside the `<select>` tag.

විය `<select>` උපුලනය තුළ තබා ඇත.

Attributes of <option>

value

Specifies the value to be sent to the server when the option is selected. If omitted, the text content of the `<option>` is sent.

විකල්පය තෝරාගත් විට සේවාදායකය වෙත යැවිය යුතු අයය සඳහන් කරයි. මෙය ඉවත් කර ඇත්තේ, `<option>` හි පෙළ අන්තර්ගතය යවතු ලැබේ.

selected

Pre-selects the option when the page loads.

පිටුව පූර්ණය වන විට විකල්පය පෝර-තෝරා ගනී.

disabled

Disables the option, preventing user selection.

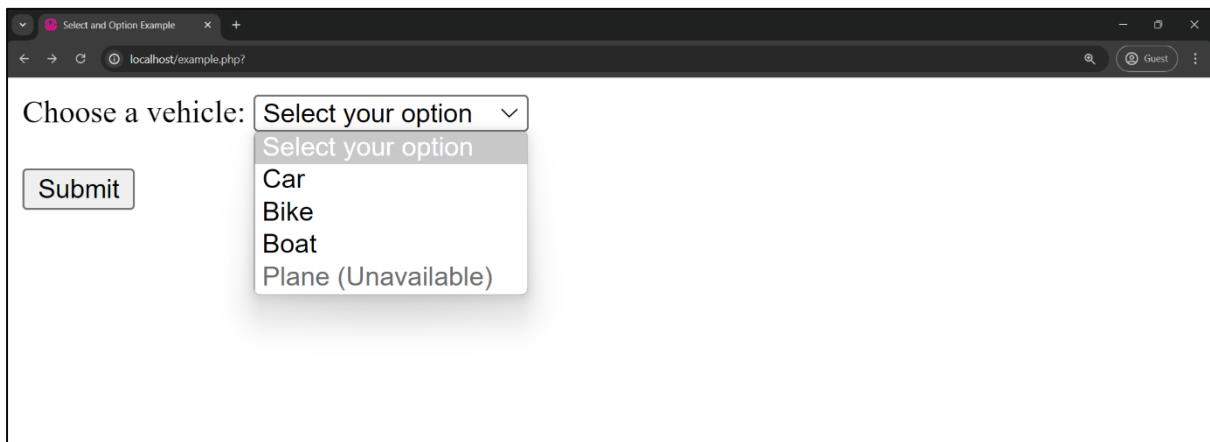
විකල්පය අවුරුදු කරයි, පරිශීලක තෝරීම වළක්වයි.

Example.

```
<html>
<head>
    <title> Select and Option example </title>
</head>
<body>
<form action="/submit_form" method="post">

    <label for="vehicle-select">Choose a vehicle:</label>
    <select name="vehicle" id="vehicle-select" required>
        <option value="" disabled selected>Select your option</option>
        <option value="car">Car</option>
        <option value="bike">Bike</option>
        <option value="boat">Boat</option>
        <option value="plane" disabled>Plane (Unavailable)</option>
    </select>

</form>
</body>
</html>
```



Special Note on the name Attribute

The name attribute in the `<select>` tag specifies the key used to reference the selected value when the form data is submitted.

`<select>` උපනය තුළ ඇති උපලක්ෂණය පෝරම දත්ත submit කරන විට තෝරාගත් අගය යොමු කිරීමට හාවතා කරන යතුරු සඳහන් කරයි.

This attribute is crucial for server-side processing.

සේවාදායක අන්තරේ සැකසුම් සඳහා මෙම උපලක්ෂණය ඉතා වැදගත් වේ.

Action attribute

Action උපලක්ෂණය

The action attribute of the <form> tag specifies the URL where the form data should be sent when the form is submitted. <form> හුවනයේ action උපලක්ෂණය පෝරමය submit කරන විට පෝරම දත්ත යෙවිය යුතු URL වික සඳහන් කරයි.

It defines the location of the server-side script or application that will process the form data.

විය පෝරම දත්ත සකසන, සේවාදායක අන්තයේ කෙතය හෝ යෙදුමේ පිහිටීම තීර්වවනය කරයි.

This attribute is essential for determining how the form data is handled after submission.

Submit කිරීමෙන් පසු පෝරම දත්ත හඳුරුවන්නේ කෙසේද යන්න තීරණය කිරීම සඳහා මෙම උපලක්ෂණය අත්‍යවශ්‍ය වේ.

Method attribute

Method උපලක්ෂණය

When an HTML form is submitted, the data can be sent to the server using either the GET or POST method.

HTML පෝරමයක් submit කළ විට, දත්ත GET හෝ POST කුමය නාවිතයෙන් සේවාදායකයට යෙවිය හැක.

Each method has its own characteristics and use cases.

මෙම කුම දෙකටම වෙන් වූ ලක්ෂණ සහ නාවිත අවස්ථා ඇත.

GET Method

The GET method appends form data to the URL specified in the action attribute of the form.

GET කුමය පෝරමයේ action ගණාගයේ දක්වා ඇති URL හි පෝරමයට ආලාතය කළ දත්ත විකතු කරයි.

This means the data is visible in the browser's address bar.

මෙයෙන් අදහස් කරන්නේ විම දත්ත අතරක්ෂුවේ ලිපින තීරුවේ දිස්වන බවයි

Characteristics of the GET Method

GET කුමයේ ලක්ෂණ

Data is appended to the URL as query parameters, making it visible and shareable.

දැනුමාන සහ බෙදාගත හැකි ලෙස දත්ත විමසුම් පරාමිති ලෙස URL වෙත විකතු කරයි.

The length of the URL is limited, so GET requests are suitable for sending small amounts of data.

URL හි එග සිමිත වේ, විබැවේන් GET ඉල්ලීම් කඩා දත්ත ප්‍රමාණයක් යැවීම සඳහා සුදුසු වේ.

GET requests can be cached by browsers and intermediary devices.

GET ඉල්ලීම් අතරක්ෂු සහ අතරමැදි උපාංග මගින් තිහිත කළ හැක.

Repeated requests (Ex. refreshing the page) with the same URL will have the same effect.

විකම URL සමග හැවත හැවත ඉල්ලීම් (ලභ.

පිටුව refresh කිරීම) මගින් විකම බලපෑමක් ඇති කරයි.

Example:

```
<!DOCTYPE html>
<html>
<head></head>
<body>

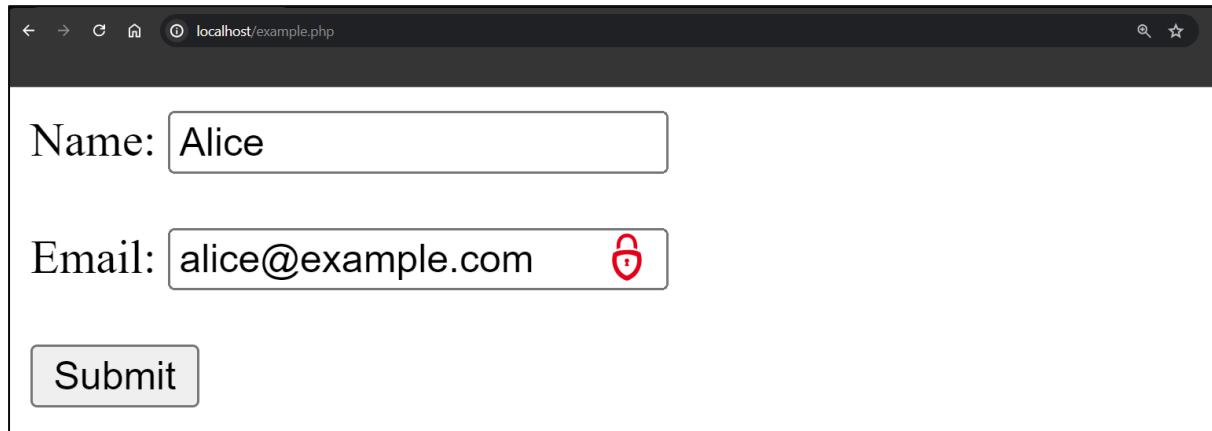
    <form action="process.php" method="get">

        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>

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

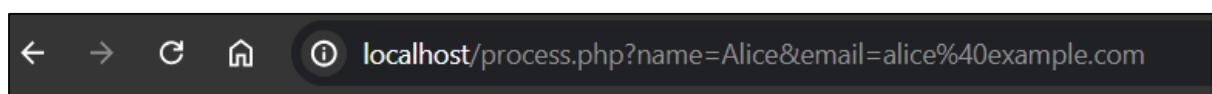
        <input type="submit" value="Submit">

    </form>
</body>
</html>
```



If the user enters "Alice" for the name and "alice@example.com" for the email, the URL will look like this upon submission:

පරිශීලකයා නම සඳහා "Alice" සහ විද්‍යුත් තැපෑල සඳහා "alice@example.com" ඇතුළු කරන්නේ නම්, Submit කිරීමේදී URL වික මෙලෙස දිස් වේ:



Use Cases for GET

GET වල හාවත

Searching for information (Ex. search queries).

නොරතුරු සේවීම (ලභ. සේවුම් විමසුම්).

Retrieving data that does not change the state of the server.

සේවාලායකයේ තත්ත්වය වෙනස් නොවන දත්ත ලබා ගැනීම.

Bookmarking and sharing URLs with specific parameters.

නිශ්චිත පරාමිතීන් සහිත URL පිටු සලකුණු කිරීම සහ බෙදාගැනීම සඳහා.

Used to retrieve non-sensitive data (not suitable for things like passwords)
සංවේදී නොවන දත්ත ලබා ගැනීමට හාවතා කරයි (මුරපද වැනි දේ සඳහා සුදුසු නොවේ)

POST Method

The POST method sends form data in the body of the HTTP request, not in the URL.
POST කුමය මගින් පෝරම දත්ත යවනුයේ HTTP ඉල්ලීමේ අන්තර්ගතය තුළය. URL තුළ නොවේ.

This makes the data invisible in the browser's address bar.

මෙමගින් මූල්‍යය මේ ලිපින සීරුවේ දත්ත නොපෙනි යයි.

Characteristics of the POST Method

POST කුමයේ ලක්ෂණ

Data is sent in the body of the HTTP request, not visible in the URL.

දත්ත HTTP ඉල්ලීමේ අන්තර්ගතය තුළ යවන ලද්දේ, URL හි නොපෙන්.

POST requests can handle larger amounts of data compared to GET requests.

GET ඉල්ලීම්වලට සාපේක්ෂව POST ඉල්ලීම්වලට විශාල දත්ත ප්‍රමාණයක් හැසිරවීය හැක.

POST requests are not cached by default.
POST ඉල්ලීම් පෙරනිමයෙන් නිශ්චිත නොවේ.

POST requests cannot be bookmarked because the data is not in the URL.

දත්ත URL හි නොමැති නිසා POST ඉල්ලීම් පිටු සලකුණු කළ නොහැක.

Repeated requests can have different effects (e.g., submitting a form multiple times).

නැවත නැවත කරන ලද ඉල්ලීම්වලට විවිධ බලපෑම් ඇති විය හැක (ලභ: පෝරමයක් කිහිප වත්‍යාවක් කිරීම).

Example:

```
<!DOCTYPE html>
<html>
<head></head>
<body>

    <form action="process.php" method="post">

        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>

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

        <input type="submit" value="Submit">

    </form>
</body>
</html>
```

Use Cases for POST

POST වල භාවිත

Submitting form data that changes the state of the server (Ex. creating a new user, processing payments).

සේවාදායකයේ තත්ත්වය වෙනස් කරන පෝරම දැනු ඉදිරිපත් කිරීම (ලඛ: නව පරීක්ෂණයක තිර්මාණය කිරීම, ගෙවීම් සැකසීම).

Sending large amounts of data.
විශාල දැනු ප්‍රමාණයක් යැවීම.

Uploading files.
ගොනු උපිගත කිරීම.

Sending sensitive information (though HTTPS is recommended for security)
සංවේදී තොරතුරු යැවීම (අරක්ෂාව සඳහා
HTTPS නිර්දේශ කර තිබුණුද)

Security Considerations

Validation

Always validate and sanitize form data to prevent security issues like SQL injection and XSS attacks.

SQL injection සහ XSS ප්‍රකාර වැනි ආරක්ෂක ගැටුම වැළැක්වීමට පෝරම දැනු සැම විටම වෙළඳ කරන්න.

HTTPS

Use HTTPS to encrypt data in transit, especially when using GET for sensitive data.

හුවමාරුවේදී දැනු සංකේතනය කිරීමට HTTPS භාවිත කරන්න, විශේෂයෙන් සංවේදී දැනු සඳහා GET භාවිත කරන විට.

Data Exposure

Avoid using GET for sensitive data, as it can be exposed in browser history, server logs, and referral headers.

සංවේදී දැනු සඳහා GET භාවිත කිරීමෙන් වළකින්න, විය ඔවුසර ඉතිහාසය, සේවාදායක ලොග සහ ගොමු ශීර්ෂයන් තුළ නිරාවරණය විය හැක.

Superglobal variables

සුපර් ගෝලීය විවෘතය

Superglobal variables in PHP are built-in global arrays that are always accessible, regardless of scope, meaning you can access them from any function, class, or file without needing to do anything special.

PHP හි ඇති සුපර් ගෝලීය විවෘතයන් යනු පටය නොසලකා සෑම විවෘත ප්‍රවේශ විය හැකි ගෝලීය අරාවන් වේ, වහාම ඔබට විශේෂ කිසිවක් සිදු කිරීමට අවශ්‍ය නොවේ ඕනෑම ශ්‍රීතයකින්, පහ්තියකින් හෝ ගොනුවකින් ඒවාට ප්‍රවේශ විය හැකි.

These superglobal variables are predefined by PHP and provide access to various kinds of data.

මෙම සුපර් ගෝලීය විවෘතයන් PHP මගින් සුද්ධ තිරේච්‍යා කර ඇති අතර විවිධ ආකාරයේ දත්ත වෙත ප්‍රවේශය සපයයි.

List of Superglobal Variables

සුපර් ගෝලීය විවෘතය ඉඩිස්ත්බු

- `$_GLOBALS`
- `$_SERVER`
- `$_GET`
- `$_POST`
- `$_FILES`
- `$_COOKIE`
- `$_SESSION`
- `$_REQUEST`
- `$_ENV`

We will only study about those are in the syllabus.

අප විසින් විෂය නිර්දේශයේ අඩංගු ඒවා පමණක් ඉගෙන ගනිමු.

`$_GLOBALS` (discussed earlier)

Contains a reference to every variable that is currently available in the global scope of the script.

කේතයේ ගෝලීය පටය තුළ දැනැව පවතින සෑම විවෘතයක් සඳහාම ගොමුවක් අධිංගු වේ.

Useful for accessing global variables from any point in the script.

කේතයේ ඕනෑම ස්ථානයක සිට ගෝලීය විවෘතයන් වෙත ප්‍රවේශ වීම සඳහා ප්‍රයෝගනවත් වේ.

```
<?php  
$x = 10;  
function test() {  
    $GLOBALS['x']++;  
}  
test();  
echo $x;  
?>
```

11

`$_SERVER`

Contains information about headers, paths, and script locations. It is filled by the web server.

ශීර්ෂක, මාර්ග සහ කේත ස්ථාන පිළිබඳ තොරතුරු අධිංගු වේ. විය වෙති සේවාදායකය විසින් පුරවනු ඇත.

Common Elements

`$_SERVER['PHP_SELF']`

The filename of the currently executing script.

දැනැව ක්‍රියාත්මක වන කේතයේ ගොනු නාමය

`$_SERVER['REQUEST_METHOD']`

The request method used to access the page (Ex. GET, POST).

පිටුවට ප්‍රවේශ වීමට සාම්බා කරන ඉල්ලීම් ක්‍රමය (Ex., GET, POST)

`$_SERVER['SERVER_NAME']`

The name of the server host.
සේවාලායක සත්කාරකයේ නම.

`$_SERVER['HTTP_HOST']`

The HTTP Host header.
HTTP සත්කාරක හිරුණකය.

`$_SERVER['HTTP_USER_AGENT']`

The user agent string of the browser.
බ්‍රවූසරයේ පරිශීලක නියෝජිත පෙළ.

We can use the `var_dump()` to look
what's inside `$_SERVER`

`$_SERVER` තුළ ඇති දේ බැඳුමට අපට
`var_dump()` හාවිතා කළ නැත

```
<?php  
var_dump($_SERVER);  
?>
```

```
D:\wamp64\www\example.php:3:  
array (size=43)  
  'HTTP_HOST' => string 'localhost' (length=9)  
  'HTTP_CONNECTION' => string 'keep-alive' (length=10)  
  'HTTP_CACHE_CONTROL' => string 'max-age=0' (length=9)  
  'HTTP_SEC_CH_UA' => string '"Not/A)Brand";v="8", "Chromium";v="126", "Google Chrome";v="126"' (length=64)  
  'HTTP_SEC_CH_UA_MOBILE' => string '?0' (length=2)  
  'HTTP_SEC_CH_UA_PLATFORM' => string '"Windows"' (length=9)  
  'HTTP_UPGRADE_INSECURE_REQUESTS' => string '1' (length=1)  
  'HTTP_USER_AGENT' => string 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)  
  'HTTP_ACCEPT' => string 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/  
  'HTTP_SEC_FETCH_SITE' => string 'same-origin' (length=11)  
  'HTTP_SEC_FETCH_MODE' => string 'navigate' (length=8)  
  'HTTP_SEC_FETCH_USER' => string '?1' (length=2)  
  'HTTP_SEC_FETCH_DEST' => string 'document' (length=8)  
  'HTTP_REFERER' => string 'http://localhost/example.php' (length=28)  
  'HTTP_ACCEPT_ENCODING' => string 'gzip, deflate, br, zstd' (length=23)  
  'HTTP_ACCEPT_LANGUAGE' => string 'en-US,en;q=0.9' (length=14)  
  'PATH' => string 'C:\ProgramData\Oracle\Java\javapath;C:\Program Files\Common Files\Oracle\Java\javapath;C:  
  'SystemRoot' => string 'C:\WINDOWS' (length=10)  
  'COMSPEC' => string 'C:\WINDOWS\system32\cmd.exe' (length=27)  
  'PATHEXT' => string '.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC' (length=53)  
  'WINDIR' => string 'C:\WINDOWS' (length=10)  
  'SERVER_SIGNATURE' => string '<address>Apache/2.4.54 (Win64) PHP/8.0.26 mod_fcgid/2.3.10-dev Server at loc  
(length=101)  
  'SERVER_SOFTWARE' => string 'Apache/2.4.54 (Win64) PHP/8.0.26 mod_fcgid/2.3.10-dev' (length=53)  
  'SERVER_NAME' => string 'localhost' (length=9)  
  'SERVER_ADDR' => string '::1' (length=3)  
  'SERVER_PORT' => string '80' (length=2)  
  'REMOTE_ADDR' => string '::1' (length=3)  
  'DOCUMENT_ROOT' => string 'D:/wamp64/www' (length=13)  
  'REQUEST_SCHEME' => string 'http' (length=4)  
  'CONTEXT_PREFIX' => string '' (length=0)  
  'CONTEXT_DOCUMENT_ROOT' => string 'D:/wamp64/www' (length=13)  
  'SERVER_ADMIN' => string 'wampserver@wampserver.invalid' (length=29)  
  'SCRIPT_FILENAME' => string 'D:/wamp64/www/example.php' (length=25)  
  'REMOTE_PORT' => string '51139' (length=5)  
  'GATEWAY_INTERFACE' => string 'CGI/1.1' (length=7)  
  'SERVER_PROTOCOL' => string 'HTTP/1.1' (length=8)  
  'REQUEST_METHOD' => string 'GET' (length=3)  
  'QUERY_STRING' => string '' (length=0)  
  'REQUEST_URI' => string '/example.php?' (length=13)  
  'SCRIPT_NAME' => string '/example.php' (length=12)  
  'PHP_SELF' => string '/example.php' (length=12)  
  'REQUEST_TIME_FLOAT' => float 1721588066.2193  
  'REQUEST_TIME' => int 1721588066
```

\$ _GET

An associative array of variables passed to the current script via the URL parameters. URL පරාමිති හරහා වත්මන් කේතය වෙත යවන ලද ව්‍යවස්ථාපිත සමන්වීත Associative array විකති.

Used to collect form data after submitting an HTML form with the `method="get"`. `method="get"` පෝරමයක් submit කිරීමෙන් පසු පෝරම දැන්ත රැස් කිරීමට හාවතා කරයි.

A real-life example:

Consider we have an HTML form in a file called `example.php`.

`example.php` නම් ගොනුවක `HTML` පෝරමයක් ඇති බව සලකන්න.

Imagine we need to collect the data that user entered into the form and display that data back to the user.

පරිශීලකයා පෝරමයට ඇතුළු කළ දත්ත විකත කර වීම දත්ත නැවත පරිශීලකයාට පෙන්විය යුතු යැයි සිතන්න.

We can use another php file to collect this data. Let's call that file `process.php`.

මෙම දත්ත විකත කිරීම සඳහා අපට වෙනත් `php` ගොනුවක් හාවතා කළ හැකිය. අපි විම ගොනුව `process.php` ලෙස හඳුන්වමු.

example.php file

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Personal details</title>
</head>
<body>
<form method="get" action="process.php" style = "margin-left:100px;"> <pre>
<fieldset style = "width: 50%;">
    <legend>Data entry form</legend>
    <label for="userName">Username: </label>
    <input type="text" name="uname" id = "userName">

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

    <label for="age">Age: </label>
    <input type="number" name="age" id = "age" min = "14" max = "65">

    <input type="radio" name="gender"> Male

    <input type="radio" name="gender"> Female

    <input type="checkbox" name="agree" value = "yes"> I agree!

    <input type="submit" value = "Show data">
</fieldset>
</pre></form> </body> </html>
```

The screenshot shows a web browser window with a title bar 'Personal details' and a URL 'localhost/example.php?'. The main content is a 'Data entry form' with the following fields:

- Username:** JohnWilliam100
- Email:** JohnWillion100@gmail.com
- Age:** 17
- Gender:** Male (radio button selected)
- Agreement:** I agree! (checkbox not checked)
- Show data** (button)

When the user hits that show data button, we should be able to show the data that was entered by the user.

පරිශීලකයා show data බොත්තම විඩු විට, පරිශීලකයා විසින් අභ්‍යුත් කළ දත්ත පෙන්වීමට අපට හැකි විය යුතුය.

We have another file called process.php that is responsible for collecting and displaying this data back to the user.

අප සතුව process.php නම් තවත් ගොනුවක් ඇත, විය මෙම දත්ත විකුණ කිරීම සහ පරිශීලකයාට නැවත ප්‍රදේශනය කිරීම සඳහා වගකිව යුතුය.

Because we have used the GET method to send the data to the server, we can use the superglobal variable `$_GET` to collect that data.

සේවාලායකය වෙත දත්ත යැවීමට අප GET කුමය භාවිතා කර ඇති තිසු, විම දත්ත රස් කිරීමට අපට `$_GET` යන සුපිරි ගෝලීය විවෘතය භාවිතා කළ හැකිය.

Since `$_GET` is an associative array, the keys of the key-value pairs are the values of the name attribute, while the values of the key-value pairs are the data entered by the user. `$_GET` යන associative array විකක් බැවින්, යතුරු-අගය යුගලවල යතුරු name උපලක්ෂණය අගයන් වන අතර යතුරු-අගය යුගලවල අගයන් පරිශීලකයා විසින් අභ්‍යුත් කළ දත්ත වේ.

The PHP code to collect and display data back to the user will look like this.

දත්ත රස් කිරීමට සහ ව්‍යාපරික විෂය පරිශීලකය වෙත නැවත ප්‍රදේශනය කිරීමට අලාම PHP කේතය මේ ආකාරයෙන් පෙනෙනු ඇත.

process.php file

```
<?php

// Collecting the data using $_GET
$username = $_GET['uname'];
$email = $_GET['email'];
$age = $_GET['age'];
$gender = $_GET['gender'];

echo "<pre>"; // Using <pre> tags to make the web page look better:)

echo "Your username is $username. <br>";
echo "Your email is $email. <br>";

if($age > 18) {
    echo "You are older than 18 years.";
} elseif ($age == 18) {
    echo "You are exactly 18 years old.";
} else {
    echo "You are younger than 18 years.";
}
echo "<br>";
echo "Your gender is $gender. <br>";

// Checking if the user has checked the checkbox.
if (isset($_GET['agree'])) {
    echo "You have checked the agreed button";
} else {
    echo "You have not checked the agreed button";
}
echo "<pre>";

?>
```

When the show data button is pressed, the user will be redirected to the process.php file and displayed as follows:

දැන්ත show data බොත්තම් විඩු විට, පරිගිලකයා process.php ගොනුව වෙත හරවා යවා පහත පරිදි පෙන්වනු ඇත:



\$_POST

Used to collect form data after submitting an HTML form with the `method="post"`.

`method="post"`. සමඟ HTML පෝරමයක් කිරීමෙන් පසු පෝරම දැන්ත රැස් කිරීමට නාවතා කරයි.

In previous example, if we set the `method` attribute to `post`, we have to change the `$_GET` to `$_POST` (in `process.php` file) in order to collect the data entered by the user. There is no change in everything else. පෝර උදාහරණයේදී, අප `method` උපලක්ෂණය `post` මෙස සකසන්නේ නම්, පරිශ්ලකය විසින් ඇතුළත් කළ දැන්ත විකත කිරීම සඳහා අපට `$_GET` යන්න `$_POST` (`process.php` ගොනුවේ) මෙස වෙනස් කළ යුතුය. අනිකුත් සියල්ලේ කිසිදු වෙනසක් නොමැත.

Working with databases

Storing form data

පෝරම දැන්ත ගබඩා කිරීම

After processing the form data, the next step is to store it in a database.

පෝරම දැන්ත සකසීමෙන් පසු, ඊළග පියවර වන්නේ විය දැන්ත ගබඩාවක ගබඩා කිරීමයි.

PHP provides several extensions for database interactions, with PDO (PHP Data Objects) and MySQLi (MySQL Improved) being the most commonly used.

PDO (PHP Data Objects) සහ MySQLi (MySQL Improved) බහුලව නාවතා වන දැන්ත සමූල අන්තර්ක්‍රියා සඳහා PHP දිග කිහිපයක් සපයයි.

These extensions allow PHP to connect to various database management systems (DBMS), such as MySQL, PostgreSQL, and SQLite.

මෙම දිගුවන් MySQL, PostgreSQL, සහ SQLite වැනි විවිධ දැන්ත සමූල කළමනාකරණ පද්ධති (DBMS) වෙත සම්බන්ධ වීමට PHP හට ඉඩ සලසයි.

To connect to a database, the PHP script needs the database credentials, which typically include the hostname (or server address), database name, username, and password.

දැන්ත සමූලායකට සම්බන්ධ වීමට, PHP කේතයට දැන්ත සමූලයේ credentials අවශ්‍ය වේ, සාමාන්‍යයෙන් වියට සත්කාරක නාමය (හෝ සේවාදායක ලිපිනය), දැන්ත සමූල නාමය, පරිශ්ලක නාමය සහ මුරපදය ඇතුළත් වේ.

Using these credentials, PHP establishes a connection to the database server.

මෙම credentials නාවතා කරමින්, PHP දැන්ත සමූල සේවාදායකයට සම්බන්ධනාවයක් ස්ථාපිත කරයි.

Once connected to the database, the PHP script prepares an SQL query to insert the processed form data into the appropriate table within the database.

දත්ත සමුදායට සම්බන්ධ වූ පසු, PHP කේතය විසින් සැකසු පෝරම දත්ත දත්ත සමුදාය තුළ ඇති සුදුසු වගුවට ඇතුළු කිරීමට SQL විමසුමක් සකස් කරයි.

This involves specifying the table name and the columns where the data should be stored.

මෙහි වගුවේ නම සහ දත්ත ගබඩා කළ යුතු තීරු තියම කිරීම ඇතුළත් වේ.

Prepared statements are often used in this step to enhance security and prevent SQL injection attacks.

මෙම පියවරේදී ආරක්ෂාව වැඩි දියුණු කිරීමට සහ SQL injection attacks වැළැක්වීමට පෙර සුදුනම් කළ ප්‍රකාශ බොහෝ විට භාවිත වේ.

The PHP script then executes the SQL query.

PHP කේතය පසුව SQL විමසුම ක්‍රියාත්මක කරයි.

If the execution is successful, the data is stored in the database, and the script can provide feedback to the user, such as a confirmation message or redirection to another page.

ක්‍රියාත්මක කිරීම සාර්ථක නම්, අදාළ දත්ත, දත්ත සමුදායේ ගබඩා කර ඇති අතර, තහවුරු කිරීමේ පණ්ඩිඩියක් හෝ වෙනත් පිටුවකට හරවා ගැවීමක් වැනි ප්‍රතිපෝෂණයක් කේතය මගින් පරිශ්ලකයාට ලබා දිය හැක.

If there is an error during the execution, the script should handle it gracefully by logging the error and informing the user without exposing sensitive information.

ක්‍රියාත්මක කිරීමේදී දේශයක් තිබේ නම්, කේතය විසින් දේශය ලෙස කිරීමෙන් සහ සම්බැඳී තොරතුරු හෙපුදුරුවී නොකර පරිශ්ලකයාට දත්තා විය ඇමෙන් ලෙස හැසිරවිය යුතුය.

Retrieving and Displaying Data

දත්ත ලබා ගැනීම සහ ප්‍රදාර්ශනය කිරීම

In addition to storing data, PHP can also retrieve data from the database for various purposes.

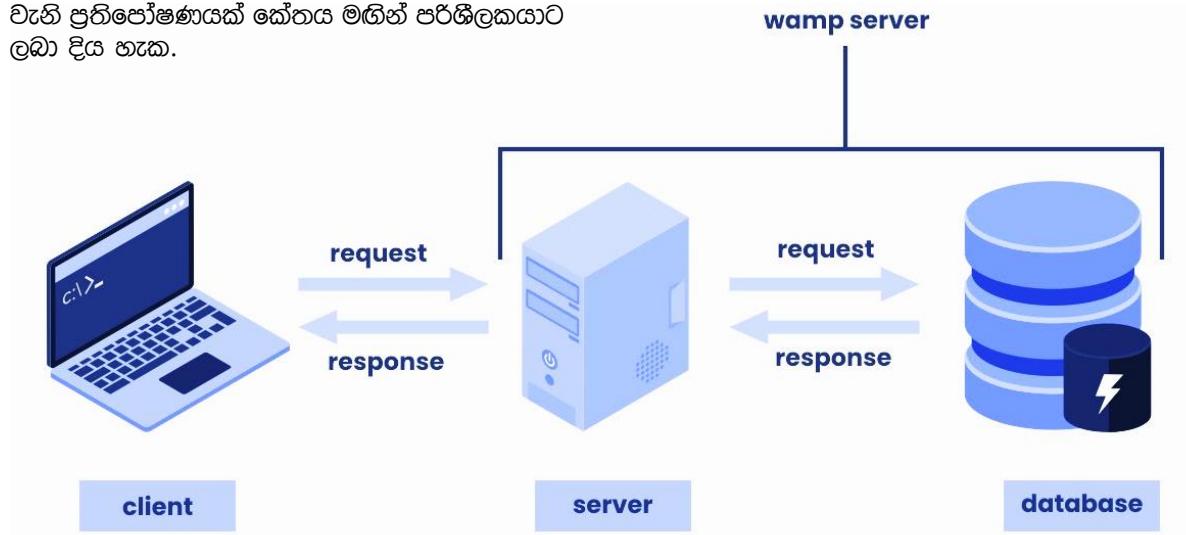
දත්ත ගබඩා කිරීමට අමතරව, PHP හට විවිධ අරමුණු සඳහා දත්ත ගබඩාවෙන් දත්ත ලබා ගැනීමටද හැක.

This involves executing SELECT queries and fetching the results.

SELECT විමසුම් ක්‍රියාත්මක කිරීම සහ ප්‍රතිච්ච ලබා ගැනීම මෙයට ඇතුළත් වේ.

The retrieved data can then be displayed on web pages, providing dynamic content based on the information stored in the database.

දත්ත සමුදායේ ගබඩා කර ඇති තොරතුරු මත පදනම්ව ගතික අන්තර්ගතයන් ලබා දෙමීන් ලබාගත් දත්ත පසුව වෙබ් පිටු මත පෙන්විය හැක.



Establishing database connection

දැන්ත සමුදා සම්බන්ධතාවය ස්ථාපිත කිරීම

There are two primary approaches to establish a database connection in PHP.

PHP ති දැන්ත සමුදා සම්බන්ධතාවයක් ස්ථාපිත කිරීම සඳහා මූලික ප්‍රශ්නයන් දෙකක් තිබේ.

1. Structured approach (procedural)
2. Object-Oriented Programming (OOP) approach.

We will use the MySQLi extension for these examples.

මෙම උදාහරණ සඳහා අප MySQLi දිගුව හාවතා කරමු

1. Structured approach (procedural)

```
<?php  
  
// Database credentials  
$servername = "localhost"; // 127.0.0.0 can also be used  
$username = "root";  
$password = "";  
$dbname = "test_database";  
  
// Create connection  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
echo "Connected successfully";  
  
//Close the connection  
mysqli_close($conn);  
?>
```

Explanation

1. Database Credentials

\$servername

This variable stores the hostname or IP address of the database server. In most local development environments, this is localhost.

මෙම විවිධය දැන්ත සමුදා සේවාදායකයේ ධරක නාමය හෝ IP ලිපිනය ගබඩා කරයි. බොහෝ ස්ථානීය සංවර්ධන පරිසරයන් තුළ, මෙය localhost වේ.

\$username

This is the username used to authenticate with the database. For WAMP servers, the default is usually root.

දැන්ත සමුදාය සමග සත්‍යාපනය කිරීමට හාවතා කරන පරිග්‍රෑක නාමය මෙයයි. WAMP සේවාදායකයන් සඳහා, පෙරනීමිය සාමාන්‍යයෙන් root වේ.

\$password

This is the password for the specified user. By default, for WAMP, it is usually an empty string ("").

මෙය යම් නිශ්චිත පරිශීලකයෙකුගේ මුද්‍රායයයි. පෙරතිමියෙන්, WAMP සඳහා, විය සාමාන්‍යයෙන් නිස් තන්තුවක් ("") .

\$dbname

This variable stores the name of the database you want to connect to. It must be created in MySQL beforehand.

මෙම විවෘත ඔබට සම්බන්ධ වීමට අවශ්‍ය දත්ත සම්බන්ධයේ නම ගබඩා කරයි. විය කළුන් MySQL නිස් නිර්මාණය කළ යුතුය.

2. Create Connection

mysqli_connect()

This function attempts to open a new connection to the MySQL server. It takes four parameters.

මෙම ලියාපෑම MySQL සේවාදායකයට නව සම්බන්ධතාවයක් විවෘත කිරීමට උත්සාහ කරයි. විය පරාමිති හතරක් ලබා ගැනී.

The function returns a MySQLi link identifier on success (connection object), or `false` on failure.

මිතය මින් සම්බන්ධතාව සාර්ථක වූ විට MySQLi සංඝැලු හැඳුනුගැනීමක් හෝ අසාර්ථක වූ විට ලෙස `false` ලබා දෙයි.

\$conn

This variable is used to store the result of the `mysqli_connect()` function which returns a connection object (if the connection is successful) or `false` (if the connection is unsuccessful).

සම්බන්ධතා වස්තුවක් (සම්බන්ධතාවය සාර්ථක නම්) හෝ `false` (සම්බන්ධතාවය අසාර්ථක නම්) ලෙස අයයක් ලබාදෙන `mysqli_connect()` ලියාපෑම් ප්‍රතිඵලය ගබඩා කිරීමට මෙම විවෘත හාවතා කරයි.

3. Check Connection

if (!\$conn)

This condition checks if the connection failed. If `$conn` is false, it means the connection was not successful.

මෙම කොන්දේසිය මින් සම්බන්ධතාවය අසාර්ථක වී ඇත්දැයි පරික්ෂා කරයි. `$conn` අසත්ත නම්, වියින් අදහස් වන්නේ සම්බන්ධතාවය සාර්ථක නොවූ බවයි.

die()

This function prints a message and terminates the current script. If the connection fails, it prints "Connection failed: " followed by the error message.

මෙම ලියාපෑම මින් පන්වීයයක් මුද්‍රණය කර වන්මත් කේතය අවසන් කරයි. සම්බන්ධතාවය අසාර්ථක වුවනාත්, විය ("Connection failed: ") යන්න දේශීල පන්වීයයක් දේශයෙද සමග මුද්‍රණය කරයි.

mysqli_connect_error()

This function returns a string description of the last connection error. It provides information about why the connection failed.

මෙම ලියාපෑම අවසාන සම්බන්ධතා දේශයේ විස්තරයක් ලබා දෙයි. සම්බන්ධතාවය අසාර්ථක වූයේ මන්දැයි යන්න විමහින් තොරතුරු සපයයි.

4. Closing the Connection

After completing the database operations, it is a good practice to close the connection to free up resources.

දත්ත සම්බන්ධතා මෙහෙයුම් සම්පූර්ණ කිරීමෙන් පසුව, සම්පත් නිදහස් කිරීම සඳහා සම්බන්ධතාවය වසා දැමීම හොඳ පුරුද්දකි.

2. Object-Oriented Programming (OOP) approach.

```
<?php  
  
// Database credentials  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "test_database";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
echo "Connected successfully";  
$conn->close();  
?>
```

Explanation

1. Create Connection

new mysqli

The new mysqli constructor in PHP is used to create an object-instance of the MySQLi class, which establishes a connection to a MySQL database.

PHP හි new mysqli යන්න MySQL දත්ත ගබඩකට සම්බන්ධයෙන් ඇති MySQLi පහේතියේ වස්තුවක්/අවස්ථාවක් නිර්මාණය කිරීමට භාවිත කරයි.

The parameters passed to the constructor are the server name, username, password, and database name.

Constructor වික වෙත ලබා දී ඇති පරාමිති වන්නේ කේවාදායකයේ නම, පරිශීලක නාමය, මුදලය සහ දත්ත සමූහ නාමයයි.

How new mysqli works මියා කරන ආකාරය

When the new mysqli constructor is called with valid parameters, it attempts to establish a connection to the MySQL database.

new mysqli constructor වික වෙත පරාමිති සම්බන්ධයෙන් ඇමතු විව, විය MySQL දත්ත සමූහයට සම්බන්ධයෙන් ඇති කර ගැනීමට උත්සාහ කරයි.

The \$conn variable holds an object of the mysqli class.

\$conn විවෘතය යුතු වූය පහේතියට අයිති වස්තුවක් දරයි.

If the connection is successful, this object contains properties and methods that can be used to interact with the database.
සම්බන්ධතාවය කාර්යක නම්, මෙම වස්තුව තුළ දත්ත සමූහය සමග අන්තර්ඩ්‍රියා කිරීමට භාවිත කළ හැකි උපලක්ෂණ සහ methods අධිංශු වේ.

The object provides methods such as `query()`, `prepare()`, `close()`, etc., which are used to execute queries and manage the connection.

මෙම වස්තුව මගින් විමුදුම් ක්‍රියාත්මක කිරීමට සහ සම්බන්ධතාවය කළමනාකරණය කිරීමට නාවතා කරන `query()`, `prepare()`, `close()` වැනි methods සපයයි.

if there is an issue connecting to the MySQL server, it still creates an instance of the `mysqli` class.

MySQL සේවාලාංකයට සම්බන්ධ වීමේ ගෙවෙනුවක් නියෝ නම්, විය විසෙවුවද `mysqli` පන්තියේ වස්තුවක් නිර්මාණය කරයි.

The `$conn` variable still holds an object of the `mysqli` class, but this object represents a failed connection.

`$conn` විවෘතය තවමත් `mysqli` පන්තියේ වස්තුවක් දරයි, නමුත් දැන් මෙම වස්තුව අකාර්ථක සම්බන්ධතාවයක් නියෝජනය කරයි.

The `connect_error` property of this object contains the error message describing why the connection failed.

මෙම වස්තුවේ `connect_error` උපලක්ෂණය තුළ සම්බන්ධතාවය අකාර්ථක වීමට හේතුව විස්තර කරන දේප පනිච්චය අඩංගු වේ.

2. Check Connection

```
if ($conn->connect_error)
```

`$conn->connect_error` is a property of the `mysqli` object that contains the error message if the connection fails.

`$conn->connect_error` යනු `mysqli` වස්තුවේ උපලක්ෂණයක් වන අතර විහි, සම්බන්ධතාවය අසමත් වූවහොත් දේප පනිච්චය අඩංගු වේ.

The if statement checks if there is a connection error.

if පකාශය මගින් සම්බන්ධතා දේපයක් නියෝදුයේ පරීක්ෂා කරයි.

```
$conn->connect_error
```

This property provides the description of the last connection error.

මෙම උපලක්ෂණය අවසාන සම්බන්ධතා දේපය පිළිබඳ විස්තරය සපයයි.

```
$conn->close()
```

Closes the database connection.

දත්ත සම්බන්ධතාවය වසා දමයි.

Collecting form data and inserting that data into the database.

පෝරම දැන්ත විකණ කර විම දැන්ත දැන්ත සම්බායට අනුලත් කිරීම.

HTML form

```
<!DOCTYPE html>
<html>
<head>
    <title>User Registration</title>
</head>
<body>
<form action="process.php" method="post">
<fieldset>
    <legend> Register Here </legend>

    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

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

    <label for="phone">Phone Number:</label>
    <input type="tel" id="phone" name="phone" required><br><br>

    <label>Gender:</label>
    <input type="radio" id="male" name="gender" value="Male" required>
    <label for="male">Male</label>
    <input type="radio" id="female" name="gender" value="Female" required>
    <label for="female">Female</label><br><br>

    <label>Interests:</label>
    <input type="checkbox" id="sports" name="interests[]" value="Sports">
    <label for="sports">Sports</label>
    <input type="checkbox" id="music" name="interests[]" value="Music">
    <label for="music">Music</label>
    <input type="checkbox" id="reading" name="interests[]" value="Reading">
    <label for="reading">Reading</label><br><br>

    <input type="submit" value="Submit">

</fieldset>
</form>

</body>
</html>
```

Let's imagine the user fills the form as follows:
පරිශීලකයා විසින් පෝරමය පහත පරිදි පුරවන්නේ යැයි සිතමු:

—Register Here—

Name:

Email: 

Phone Number:

Gender: Male Female

Interests: Sports Music Reading

Suppose we have a database table as below and want to insert form data into this table. අපට පහත පරිදි දත්ත සමූහ වගුවක් ඇති අතර පෝරමය දත්ත මෙම වගුවට අනුල් කිරීමට අවශ්‍ය බව සිතමු.

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL,
    gender VARCHAR(10) NOT NULL,
    interests VARCHAR(255)
);
```

When we collect data from users, there could be special characters or malicious inputs that can compromise the integrity and security of our SQL queries.

අම් පරිශීලකයින්ගෙන් දත්ත රයේ කරන විට, අපගේ SQL විමසුම්වල අඛණ්ඩතාව සහ ආරක්ෂාව සම්මුතියට පත් කළ හැකි විශේෂ ප්‍රක්ෂේප හෝ අනිශ්චිත යොදුවුම් තිබිය නැක.

To prevent these issues, we need to sanitize and escape these inputs, ensuring they are treated as literal data rather than executable code.

මෙම ගෙටු වලක්වා ගැනීම සඳහා, ක්‍රියාත්මක කළ හැකි කේත වෙනුවට එ්වා වචනානුසාරයෙන් දත්ත ලෙස සළකන බව සහතික කරමින්, මෙම යොදුවුම් නාශනාය කර ගෙවා ගත යුතුය.

mysqli_real_escape_string()

This is a built-in PHP function that prepares a string for use in an SQL query by escaping special characters.

මෙය PHP built-in ක්‍රිතයක් වන අතර විය විශේෂ ප්‍රක්ෂේප වලින් ගෙවීමේ SQL විමසුම්ක හාවතා කිරීම සඳහා තහැකුවක් සූදානම් කරයි.

This ensures that the input data doesn't interfere with the query execution or compromise the database.

අදාළ දත්ත විමසුම් ක්‍රියාත්මක කිරීමට බාධා නොකරන බව හෝ දත්ත සමූහය සම්මුතියට පත් නොවන බව මෙමක් සහතික කරයි.

PHP Script to Process Form Data and Insert into Database (process.php file)

පෝරම දුත්ත සැකකීමට සහ දුත්ත සමුදායට ඇතුළු කිරීමට අදාළ PHP කේතය (process.php තොනුව)

Procedural approach

```
<?php

// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Collect and sanitize form data
$name = mysqli_real_escape_string($conn, $_POST['name']);
$email = mysqli_real_escape_string($conn, $_POST['email']);
$phone = mysqli_real_escape_string($conn, $_POST['phone']);
$gender = mysqli_real_escape_string($conn, $_POST['gender']);
$interests = isset($_POST['interests']) ? implode(", ", $_POST['interests']) : '';

// Prepare the SQL query
$sql = "INSERT INTO users (name, email, phone, gender, interests)
        VALUES ('$name', '$email', '$phone', '$gender', '$interests')";

// Execute the query
if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

// Close the connection
mysqli_close($conn);

?>
```

This is how the database table will look like after inserting the data entered by the user
පරිශීලකයා ආදානය කළ දැන්ත ඇතුළත් කිරීමෙන් පසු දැන්ත සමුදා වුවට මෙලෙස පෙනෙනු ඇත.

| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | id | first_name | last_name | email | phone | hire_date |
|--|--------------------------|-------------------------------|-------------------------------|---------------------------------|-----------|-------------------|------------------|---------------------------|--------------|------------------|
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 1 | John | Doe | john.doe@example.com | 555-1234 | 2021-05-15 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 2 | Jane | Smith | jane.smith@example.com | 555-5678 | 2022-03-10 |
| | <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 3 | Emily | Johnson | emily.johnson@example.com | 555-8765 | 2023-01-22 |

Explanation

HTML Form

This form collects the user's name, email, phone number, gender, and interests.
මෙම පේරුමය පරිශීලකයාගේ නම, රීමේල්, දුරකථන අංකය, ස්කේ පුරුෂ භාවය සහ රැවිකත්වයන් විකතු කරයි.

The `interests[]` checkboxes use an array notation to collect multiple values.
`interests[]` checkboxes බහු අගයන් විකතු කිරීම සඳහා අරා අංකනයක් භාවිත කරයි.

PHP Script (process.php)

User input is sanitized using `mysqli_real_escape_string()` to prevent SQL injection.
SQL injection වැළැක්වීමට `mysqli_real_escape_string()` භාවිතයෙන් පරිශීලක ආදානය භාෂණය කෙරේ.

Name, Email, Phone, and Gender:
These fields are straightforward text and radio inputs.
මෙම ක්ෂේත්‍ර සඡු පෙළ සහ රේඛියෝ ආදාන වේ.

Interests

If the user has selected any interests, the values are joined into a single string using `implode(", ", $_POST['interests'])`.
පරිශීලකයා ඩිසියම් රැවිකත්වයක් තොරාගෙන තිබේ නම්, විම අගයන් `implode(", ", $_POST['interests'])` භාවිතයෙන් තහි තන්තුවකට විකාඩ්ද කර ඇත.

SQL Query

The script prepares an SQL INSERT statement to add the collected data into the users table.
විකතු කරන ලද දැන්ත වුවට වික් කිරීමට කේතයේ විසින් SQL INSERT ප්‍රකාශයක් සකසයි.

Executing the Query

The script executes the query using `mysqli_query()`.
කේතය `mysqli_query()` භාවිතයෙන් විමසුම ක්‍රියාත්මක කරයි.

If the query is successful, it outputs a success message.
විමසුම සාර්ථක නම්, විය සාර්ථක පණිවිධියක් නිකුත් කරයි.

If there is an error, it outputs the error message.
දේශයක් තිබේ නම්, විය දේශ පණිවිධියක් ප්‍රතිඵානය කරයි.

OOP approach

```
<?php

// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection using OOP approach
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Collect and sanitize form data
$name = $conn->real_escape_string($_POST['name']);
$email = $conn->real_escape_string($_POST['email']);
$phone = $conn->real_escape_string($_POST['phone']);
$gender = $conn->real_escape_string($_POST['gender']);
$interests = isset($_POST['interests']) ? implode(", ", $_POST['interests']) : '';

// Prepare the SQL query
$sql = "INSERT INTO users (name, email, phone, gender, interests)
        VALUES ('$name', '$email', '$phone', '$gender', '$interests')";

// Execute the query
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

// Close the connection
$conn->close();
?>
```

OOP approach executes the SQL query using the `query()` method.
OOP ප්‍රධාන සේවකය `query()` යන method විසින් හාටිතයෙනා SQL විමසුම ක්‍රියාත්මක කරයි.

Retrieving Data from an Existing Table

ପାତିକିନ ଲାଗୁଳିକିନେ ଦୂର୍ବଳ ଲବା ଗେତୀମ

Let's assume we have a MySQL table named employees with the following data.

පහත දුරක්ත සහිත employees නම් MySQL වැවක් අප සතුව ඇතැයි උපකළුපනය කරමු

| id | first_name | last_name | phone | hire_date | email |
|-----------|-------------------|------------------|--------------|------------------|---------------------------|
| 1 | John | Doe | 555-1234 | 2021-05-15 | john.doe@example.com |
| 2 | Jane | Smith | 555-5678 | 2022-03-10 | jane.smith@example.com |
| 3 | Emily | Johnson | 555-8765 | 2023-01-22 | emily.johnson@example.com |

Connect to the Database

Establish a connection to the MySQL database using PHP. This involves providing the necessary credentials like server name, username, password, and database name.

PHP නාවිතයෙන් MySQL දැන්ත සමුදායට සම්බන්ධතාවයක් ස්ථාපිත කරන්න. මෙයට සේවාදායක නාමය, පරිශීලක නාමය, මුරපදය සහ දැන්ත සමුදා නාමය වැනි අවශ්‍ය අක්තපත්‍ර සැපයීම් ඇතැන් වේ.

Prepare an SQL Query

Write an SQL SELECT query to fetch the data you need. For instance, you might want to retrieve all columns from the employees table.

ଓଦିବ ଆପଣଙ୍କ ଦର୍ଶନ ଲବା ଗେତେମତ SQL SELECT ବିମୋହିତକୁ ଲିଯନ୍ତିନ. ଉଦ୍ଧବରତନୀଙ୍କୁ ଲେଖ, ଓଦିବ ଦେଖିବି ବିରୁଦ୍ଧରେ ଜୀବିତରେ ତିରିରି ଲବା ଗେତେମତ ଆପଣଙ୍କ ଵିଦ୍ୟା ହାତକିଯ.

Execute the Query

Run the SQL query using PHP. This will send the query to the MySQL database and retrieve the results.

(PHP) හාවිතයෙන් SQL විමසුම සියාත්මක කරන්න. මෙය MySQL දත්ත සම්ඳාය වෙත විමසුම යටා ප්‍රතිවශ්‍යා ලබා ගනී.

Process the Results

Check if the query returned any results. If so, fetch and process the data row by row. This usually involves iterating through the result set and displaying the data.

ବୀମାନ୍ତ କିମିଟିଲୁଗାକୁ ଲୋକେ ଆହେଦ୍ୱୟାଦି ପରିକଳ୍ପନା କରନ୍ତି. ବିଶେଷ ଲୋକେ ଆହେ ନାହିଁ, ଦୃତିର ପେଣ୍ଟିଯେନ୍ ପେଣ୍ଟିଯ ଗେନ ସକଳନ୍ତିର. ମେଘ କ୍ଷାମାନଙ୍କରେଣ୍ଟ ପ୍ରତିଲିପି କରିବିଲୁଗ ହରାହା ପ୍ରକାରର ପରିବହନ କିରୀମ କିମିଟିଲୁଗାକୁ ଲୋକେ ଆହେଦ୍ୱୟାଦି ପରିକଳ୍ପନା କରନ୍ତି.

Close the Connection

After processing the data, close the database connection to free up resources.

Retrieving Data

දුත්ත ලබා ගැනීම

Procedural Approach

```
<?php

// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// SQL query to fetch data
$sql = "SELECT * FROM employees";
$result = mysqli_query($conn, $sql);

// Check if there are any results
if (mysqli_num_rows($result) > 0) {

    // Output data of each row
    while($row = mysqli_fetch_assoc($result)) {

        echo "ID: " . $row["id"]. " - Name: " . $row["first_name"]. " " .
        $row["last_name"]. " - Email: " . $row["email"]. " - Phone: " .
        $row["phone"]. " - Hire Date: " . $row["hire_date"]. "<br>";
    }
} else {
    echo "0 results";
}

// Close connection
mysqli_close($conn);

?>
```

OOP approach

```
<?php

// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection using OOP approach
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// SQL query to fetch data
$sql = "SELECT * FROM employees";
$result = $conn->query($sql);

// Check if there are any results
if ($result->num_rows > 0) {

    // Output data of each row
    while($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"]. " - Name: " . $row["first_name"]. " " .
        $row["last_name"]. " - Email: " . $row["email"]. " - Phone: " .
        $row["phone"]. " - Hire Date: " . $row["hire_date"]. "<br>";
    }
} else {
    echo "0 results";
}

// Close connection
$conn->close();

?>
```

Explanation

Procedural

`mysqli_query($conn, $sql)`

In the procedural style, the `mysqli_query()` function is used to execute a query against the database.

Procedural කුමයේදී, දත්ත සම්බාධයට වෙත විමසුමක් කියාත්මක කිරීමට `mysqli_query()` ලිඛිත භාවිතා කරයි.

For successful SELECT, SHOW, DESCRIBE, or EXPLAIN queries, it returns a `mysqli_result` object representing the **result set**.

සාර්ථක SELECT, SHOW, DESCRIBE හෝ EXPLAIN විමසුම් සඳහා, විය ප්‍රතිචලනය තියෙන්නය කරන `mysqli_result` වස්තුවක් ලබා දෙයි.

For successful INSERT, UPDATE, or DELETE queries, it returns TRUE.

සාර්ථක INSERT, UPDATE හෝ DELETE විමසුම් සඳහා, විය TRUE ලබා දෙයි.

If the query fails, it returns FALSE.

විමසුම අසාර්ථක වූවහොත්, විය FALSE ලෙස ලබා දෙයි.

`mysqli_num_rows($result) > 0`

`mysqli_num_rows()` is used to determine the number of rows in a result set returned by a SELECT query.

`mysqli_num_rows()` යන ලිඛිත SELECT විමසුමක් ආපසු රැබෙන ප්‍රතිචලනය පෝළු ගණන තීරණය කිරීමට භාවිතා කරයි.

It takes the result set as an argument and returns an integer representing the number of rows.

විය ප්‍රතිචලනය ත්‍රේකයක් ලෙස ගෙන පෝළු ගණන නියෝජනය කරන පූර්ණ සංඛ්‍යාවක් ලබා දෙයි.

`mysqli_fetch_assoc()`

For fetching data, `mysqli_fetch_assoc()` is used to retrieve a row from the result set as an associative array.

දත්ත ලබා ගැනීම සඳහා, ප්‍රතිචලනය පෝළුයක් associative array විකක් ලෙස ලබා ගැනීමට `mysqli_fetch_assoc()` භාවිතා කරයි.

This function returns the next row as an associative array where the keys are the column names of the table, and it returns FALSE when there are no more rows.

මෙම ලිඛිත මගින් රිළුය පෝළුය associative array විකක් ලෙස ලබා දෙයි, විශිෂ්ට යනුරු යනු වගුවේ තීරු නාම වන අතර තවත් පෝළු නොමැති විට විය FALSE ලෙස ලබා දෙයි.

OOP

\$conn->query(\$sql)

In the object-oriented style, the query() method of the mysqli class is used to perform a query against the database.
Object-oriented තුමයේදී, mysqli පන්තියේ query() වික දත්ත සමූහය වෙත විමසුමක් සිදු කිරීමට භාවිතා කරයි.

For successful SELECT, SHOW, DESCRIBE, or EXPLAIN queries, it returns a mysqli_result object representing the result set.

සාර්ථක SELECT, SHOW, DESCRIBE හෝ EXPLAIN විමසුම් සඳහා, විය ප්‍රතිච්‍රිත කට්ටලය නියෝජනය කරන mysqli_result වස්තුවක් බඟ දෙයි.

For successful INSERT, UPDATE, or DELETE queries, it returns TRUE.

සාර්ථක INSERT, UPDATE හෝ DELETE විමසුම් සඳහා, විය TRUE බඟ දෙයි.

If the query fails, it returns FALSE.

විමසුම අසාර්ථක ව්‍යවහාරක්, විය FALSE ලෙස බඟ දෙයි.

\$result->num_rows > 0

The num_rows property of a mysqli_result object is used to determine the number of rows in a result set returned by a SELECT query.

mysqli_result වස්තුවක num_rows උපලක්ෂණය SELECT විමසුමකින් ආපසු ලබයෙන ප්‍රතිච්‍රිත කට්ටලයක ජේල් ගණන තීරණය කිරීමට භාවිතා කරයි.

It provides an integer representing the number of rows.

විය ජේල් ගණන නියෝජනය කරන පූර්ණ සංඛ්‍යාවක් සපයයි.

\$result->fetch_assoc()

For fetching data, the fetch_assoc() method of the mysqli_result class is used to retrieve a row from the result set as an associative array.

දත්ත බඟ ගැනීම සඳහා, ප්‍රතිච්‍රිත කට්ටලයෙන් ජේල්යක් associative array විකක් ලෙස බඟ ගැනීමට mysqli_result පන්තියේ fetch_assoc() method වික භාවිතා කරයි.

This method returns the next row as an associative array where the keys are the column names of the table, and it returns FALSE when there are no more rows.

මෙම method වික මගින් රේපර ජේල්ය associative array විකක් ලෙස බඟ දෙයි, විහිදු යතුරු යනු වශයෙන් තීරණ භාම වන අතර තවත් ජේල් නොමැති විට විය FALSE ලෙස බඟ දෙයි

Other SQL operations using PHP

PHP හාවතා කරන අනෙකුත් SQL මෙහෙයුම්

Creating a table

වගුවක් නිර්මාණය කිරීම

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// SQL query to create table
$sql = "CREATE TABLE employees (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    phone VARCHAR(20),
    hire_date DATE
)";

// Execute the query
if ($conn->query($sql) === TRUE) {
    echo "Table employees created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

// Close the connection
$conn->close();
?>
```

Updating table data

වග දත්ත යාවත්කාලීන කිරීම

```
<?php  
  
// Database credentials  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "test_database";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
// SQL query to update data  
$sql = "UPDATE employees SET email='john.newemail@example.com' WHERE id=1";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Record updated successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}  
  
// Close connection  
$conn->close();  
  
?>
```

Deleting table data

වග දත්ත මකා දැයුම

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test_database";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// SQL query to delete data
$sql = "DELETE FROM employees WHERE id=3";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

// Close connection
$conn->close();
?>
```