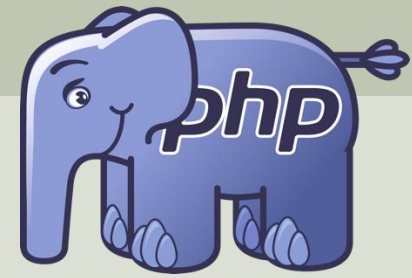


<https://www.halvorsen.blog>



# PHP Tutorial

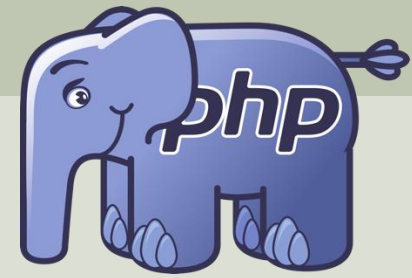
Hans-Petter Halvorsen



# Contents

- [Introduction to Web Development](#)
- [Introduction to PHP](#)
- [Getting Started with PHP](#)
- [PHP Programming](#)
- [HTML Forms in PHP](#)
- [Session Variables in PHP](#)
- [PHP and MySQL Database](#)

<https://www.halvorsen.blog>



PHP Tutorial

# Introduction



Hans-Petter Halvorsen

[Table of Contents](#)

# Web Development Frameworks

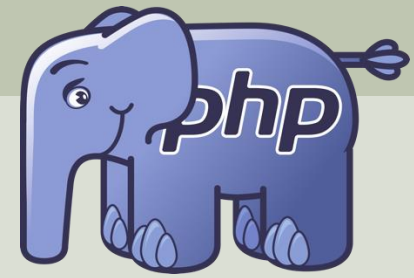
Some of the most used server-side (backend) Web Development Frameworks:

- **PHP** (Scripting language for Web development)
- **ASP.NET Core** (Microsoft, Visual Studio, C#)
- **Ruby on Rails** (Ruby)
- **Django** (Python)
- ++++

# Tools

- **PHP** - a server scripting language for making dynamic web pages/applications, typically communicating with a Database.
- We will host our PHP files on an existing **Web Server** that supports PHP and MySQL.
- We will use **Visual Studio Code** (you can use another IDE if you prefer).
- We will transfer the local files to the Web Server using **FTP** (File Transfer Protocol). We will use **WinSCP** (you can use another FTP tool if you prefer).
- **MySQL** - a widely used relational database system. MySQL is free and open-source.
- **phpMyAdmin** - a free and open-source administration tool for MySQL (and MariaDB).

<https://www.halvorsen.blog>



PHP Tutorial

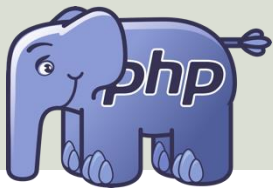
# Introduction to PHP

Hans-Petter Halvorsen



# PHP

- PHP is a **server scripting language** for making dynamic and interactive web pages.
- PHP scripts are executed on the server/webserver.
- PHP files have extension ".php" and are typically a mix of PHP, HTML, CSS and JavaScript
- PHP is free and open-source.
- With PHP you can easily communicate with a Database, and especially MySQL.
- LAMP: Linux, Apache, MySQL and PHP.
- PHP is widely used and still by far the most used/popular language for web development.
- PHP is easy to learn (but still very powerful) – which cannot be said on many other web technologies and programming languages.
- Homepage: <https://www.php.net>



# PHP + MySQL



- You need to have a **PHP** + **MySQL** Environment on your local computer on get access to it from a server/Internet.
- For local installation you need to download and install **Apache** (or another web server that supports PHP), **PHP** and **MySQL**.
- You can get server access from many providers (free or paid). Just search for, e.g., “PHP web hosting”.
- (I will use an internal **LAMP** server available for employees and students at my University.)



# LAMP

- LAMP = **L**inux, **A**pache, **M**ySQL, **P**HP
  - PHP is the Programming Language
  - MySQL is the Database System
  - Apache is the Web Server software
  - Linux is the operating system where the Web Server is running

Each part in LAMP is free and open-source, so it is a popular web hosting environment. You find also lots of online documentation and a large community.

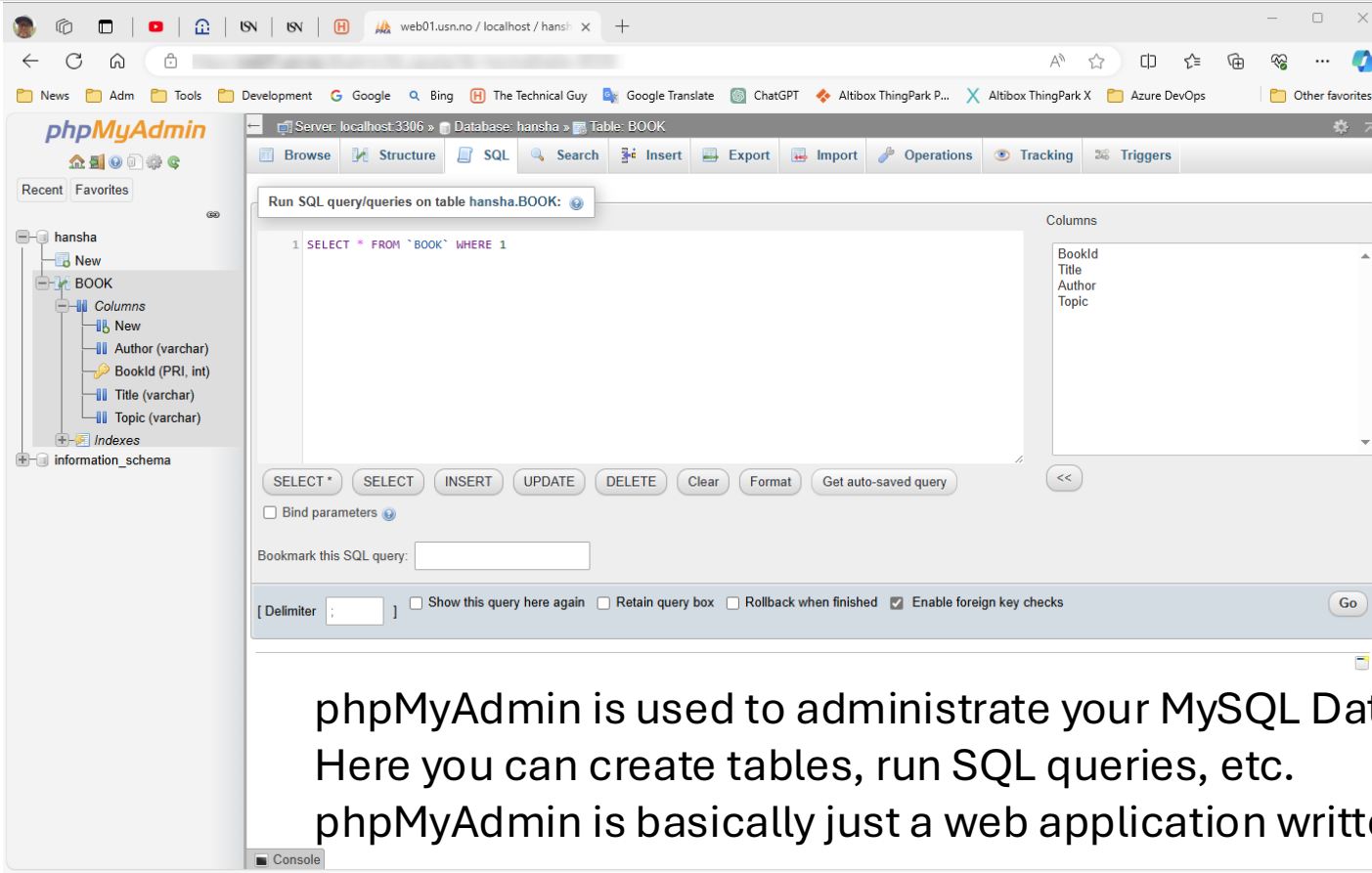
# LAMP/PHP Web Hosting

- There exists hundreds/thousands of different LAMP/PHP/MySQL Hosting Providers, some free but mostly paid options.
- Hostinger - <https://www.hostinger.no>
- InfinityFree - <https://www.infinityfree.com>
- PRO ISP - <https://www.proisp.no>
- +++ (Just Google)

# XAMPP

- XAMPP is a popular PHP development environment
- XAMPP is an acronym that stands for Cross-Platform, Apache, MySQL, PHP, and Perl.
- It works on Windows, macOS and Linux
- It installs Apache, MariaDB, PHP and Perl.
  - Apache is a Web Server.
  - MariaDB is almost identical to MySQL.
  - PHP and Perl are Web Programming languages.
- <https://www.apachefriends.org>

# phpMyAdmin



The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL `web01.usn.no / localhost / hansha`. The interface includes a sidebar on the left with a tree view of the database structure: `hansha` (database) containing `New` (table), `BOOK` (table), `Columns` (view), `Indexes` (view), and `information_schema` (database). The main panel shows the `BOOK` table structure with columns: `BookId` (PRIMARY, INT), `Author` (VARCHAR), `Title` (VARCHAR), and `Topic` (VARCHAR). A SQL query is entered in the query box: `1 SELECT * FROM `BOOK` WHERE 1`. Below the query box are buttons for `SELECT *`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `Clear`, `Format`, and `Get auto-saved query`. There is also a checkbox for `Bind parameters`. At the bottom, there is a `Bookmark this SQL query:` field, a `Delimiter` dropdown set to `:`, and checkboxes for `Show this query here again`, `Retain query box`, `Rollback when finished`, and `Enable foreign key checks`. A `Go` button is located at the bottom right of the main panel.

phpMyAdmin is used to administrate your MySQL Database. Here you can create tables, run SQL queries, etc. phpMyAdmin is basically just a web application written in PHP.

# PHP Code Editors

You can use any kind of tool/IDE for Python(Django development). Here are some recommendations:

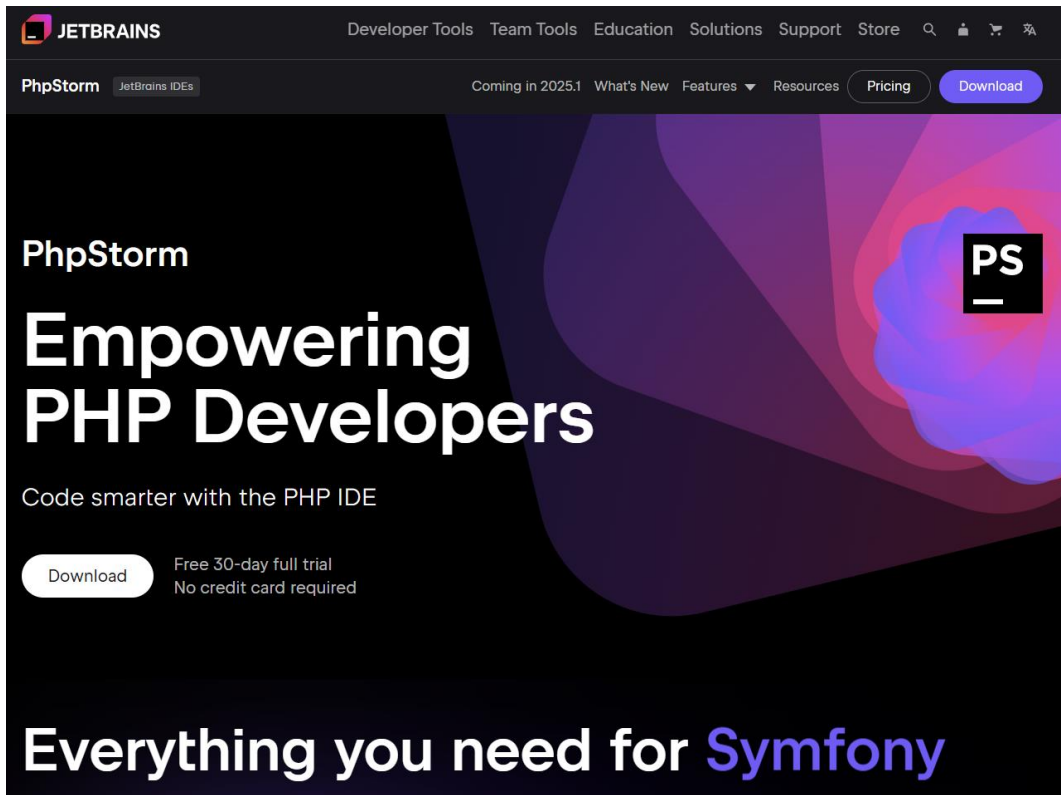
- **Visual Studio Code.**
  - Multiplatform and Free.
  - Homepage: <https://code.visualstudio.com>
- **PhpStorm.**
  - Free 30-day full trial.
  - Free access to all JetBrains IDEs for students and educators.
  - Homepage: <https://www.jetbrains.com>
- +++

# Visual Studio Code

- Visual Studio Code is a free and open-source code editor developed by Microsoft.
- It supports many programming languages, including PHP, through an extension.
- Visual Studio Code is a general-purpose Code Editor for almost any kind of programming language or framework through so-called extensions that you can install on top of the IDE.

# PhpStorm

IDE for PHP development by JetBrains



The screenshot shows the PhpStorm website landing page. At the top is the JetBrains logo and a navigation bar with links: Developer Tools, Team Tools, Education, Solutions, Support, and Store. Below the navigation bar are tabs for PhpStorm and JetBrains IDEs. The main content area features the PhpStorm logo, the tagline 'Empowering PHP Developers', and the text 'Code smarter with the PHP IDE'. A 'Download' button is prominently displayed, with the text 'Free 30-day full trial No credit card required' below it. At the bottom, it says 'Everything you need for Symfony'. The background has a dark theme with abstract purple and blue shapes.

JETBRAINS Developer Tools Team Tools Education Solutions Support Store

PhpStorm JetBrains IDEs Coming in 2025.1 What's New Features Resources Pricing Download

PhpStorm

**Empowering  
PHP Developers**

Code smarter with the PHP IDE

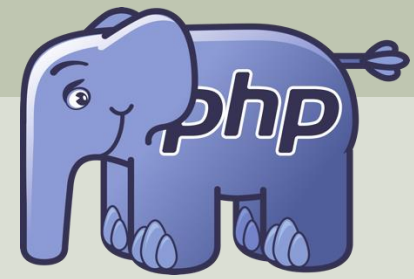
Download Free 30-day full trial  
No credit card required

Everything you need for **Symfony**

## Free Educational Use:

- JetBrains offers free PhpStorm licenses and special deals for educational purposes.
- Students and teachers are eligible to use the JetBrains All Products Pack (which includes PhpStorm, as well as other JetBrains IDEs and tools) free of charge.

<https://www.halvorsen.blog>



PHP Tutorial

# Getting Started with PHP



Hans-Petter Halvorsen

[Table of Contents](#)



# Getting Started

- We need a **Web Server with PHP installed**
  - You can setup your own server with PHP, but I will just use an existing webserver with PHP in this tutorial.
- We need a Code Editor like **Visual Studio Code**.
- We need an **FTP** program like **WinSCP** to transfer files from local PC to the webserver.
- You will also need to have a **MySQL** Database for some parts in this tutorial.

# My first PHP page

Visual Studio Code

```
<?php
$name = "Hans-Petter Halvorsen";
echo "Hello, my name is $name";
?>
```

index.php

PHP Files have the file extension **.php**

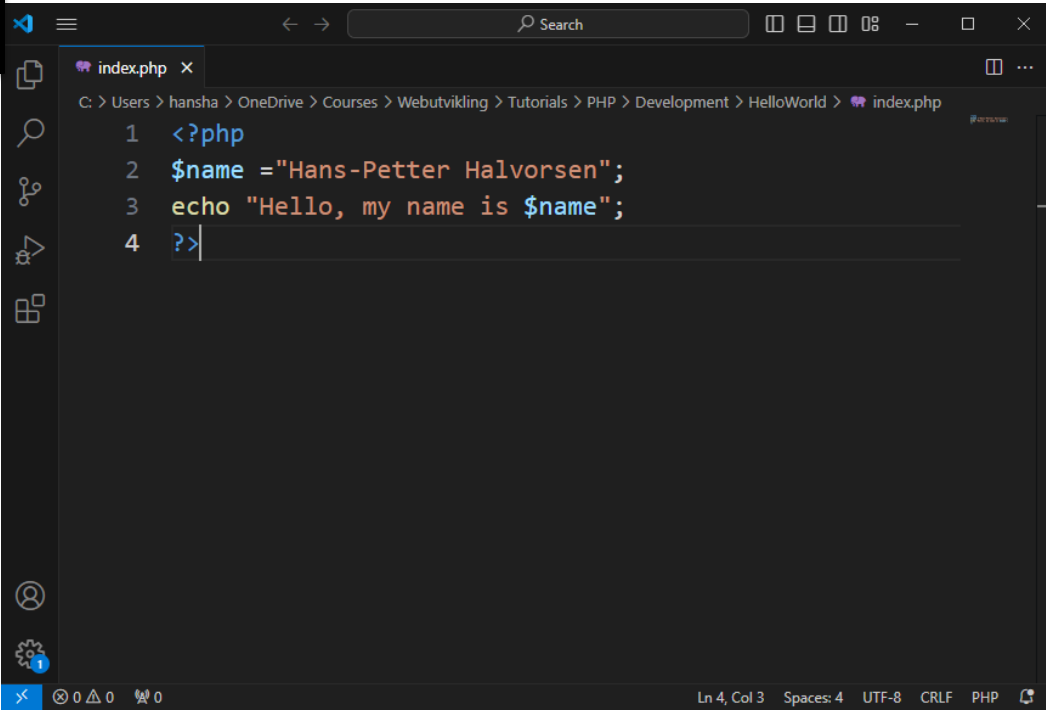
The startup file is typically names **index.php**

PHP code should be put inside

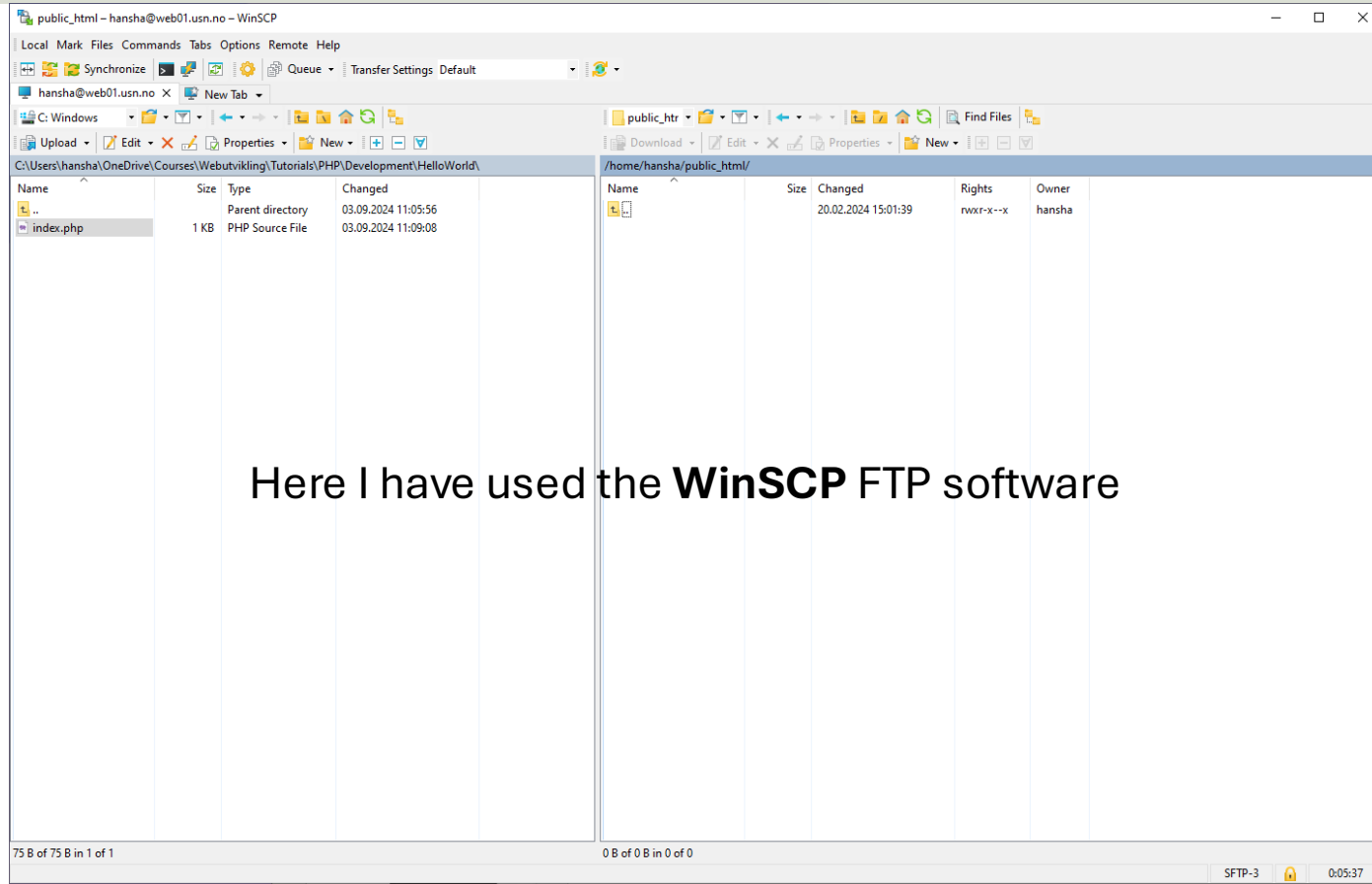
```
<?php
```

```
..
?>
```

Variables starts with **\$**

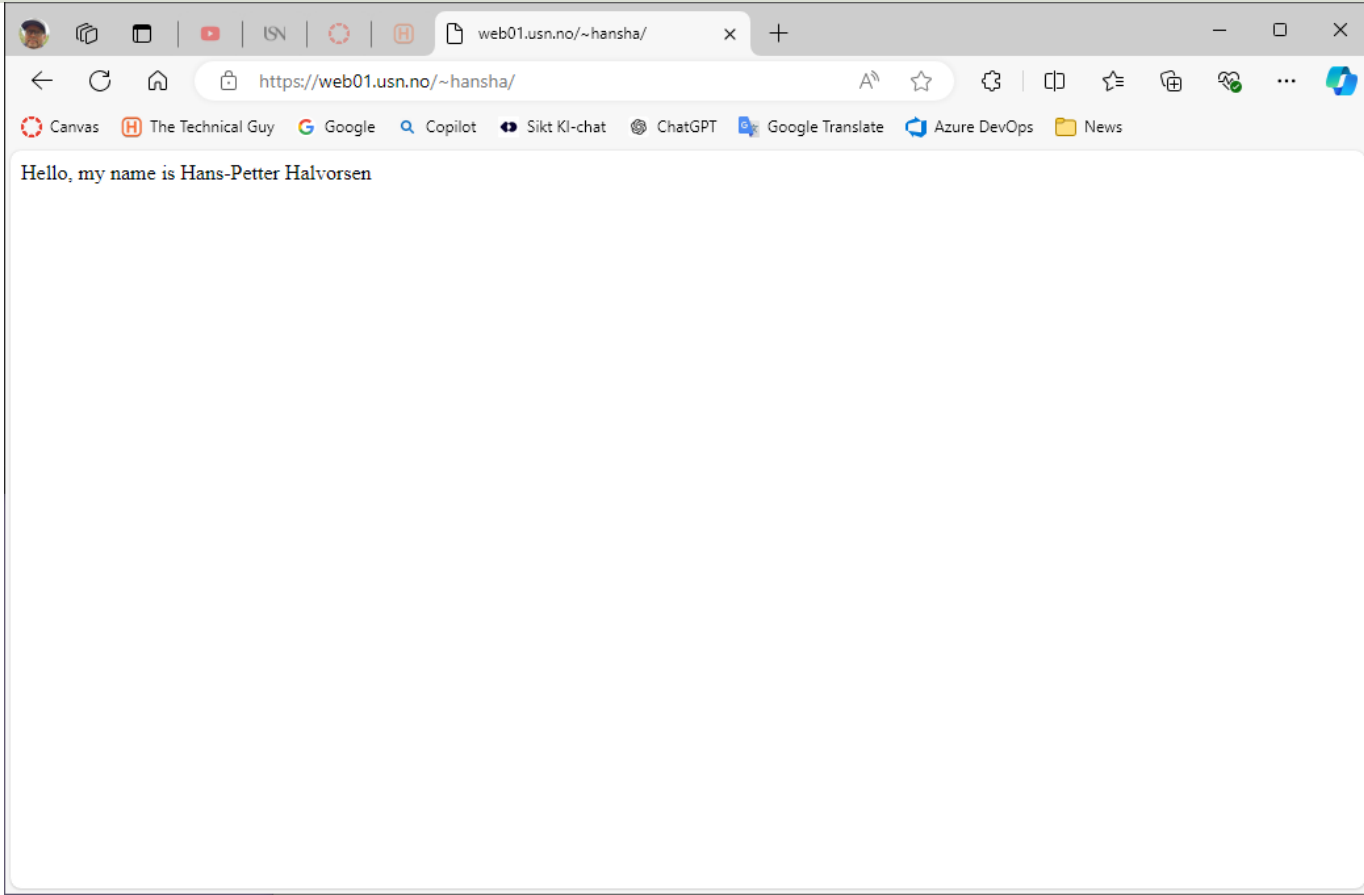


# Upload File using FTP - WinSCP



Here I have used the **WinSCP** FTP software

# Test in your Web Browser



# Code Explained

```
<?php
$name = "Hans-Petter Halvorsen";
echo "Hello, my name is $name";
?>
```

PHP code should be put inside

```
<?php
```

```
..
```

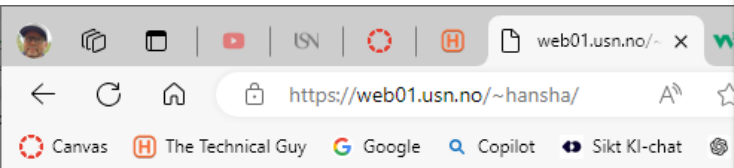
```
?>
```

Variables starts with **\$**

“**echo**” is a built-in function in PHP that is much used to output text or contents of a variable to the web browser.

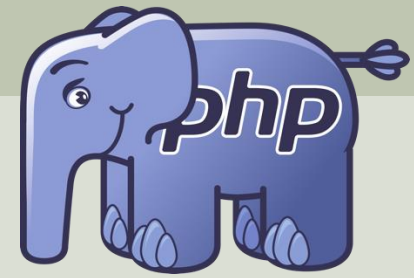
# HTML + PHP

Typically, you include PHP code in between your HTML code. Here is a basic example:



```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello World</h1>
    <?php
      $name = "Hans-Petter Halvorsen";
      echo "Hello, my name is $name";
    ?>
  </body>
</html>
```

<https://www.halvorsen.blog>



PHP Tutorial

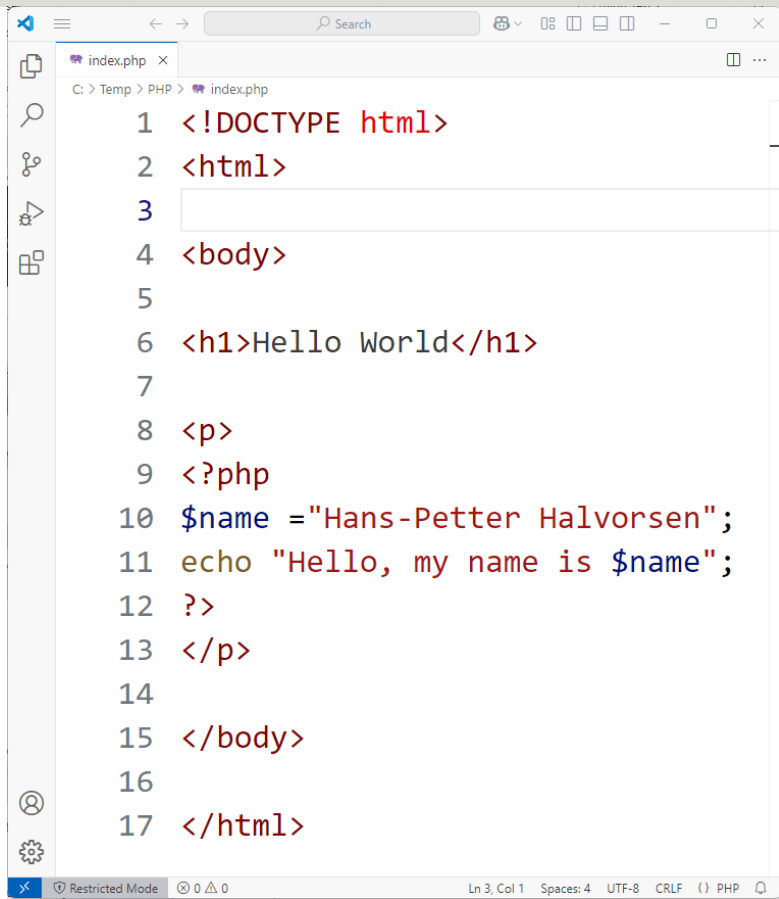
# PHP Programming



Hans-Petter Halvorsen

[Table of Contents](#)

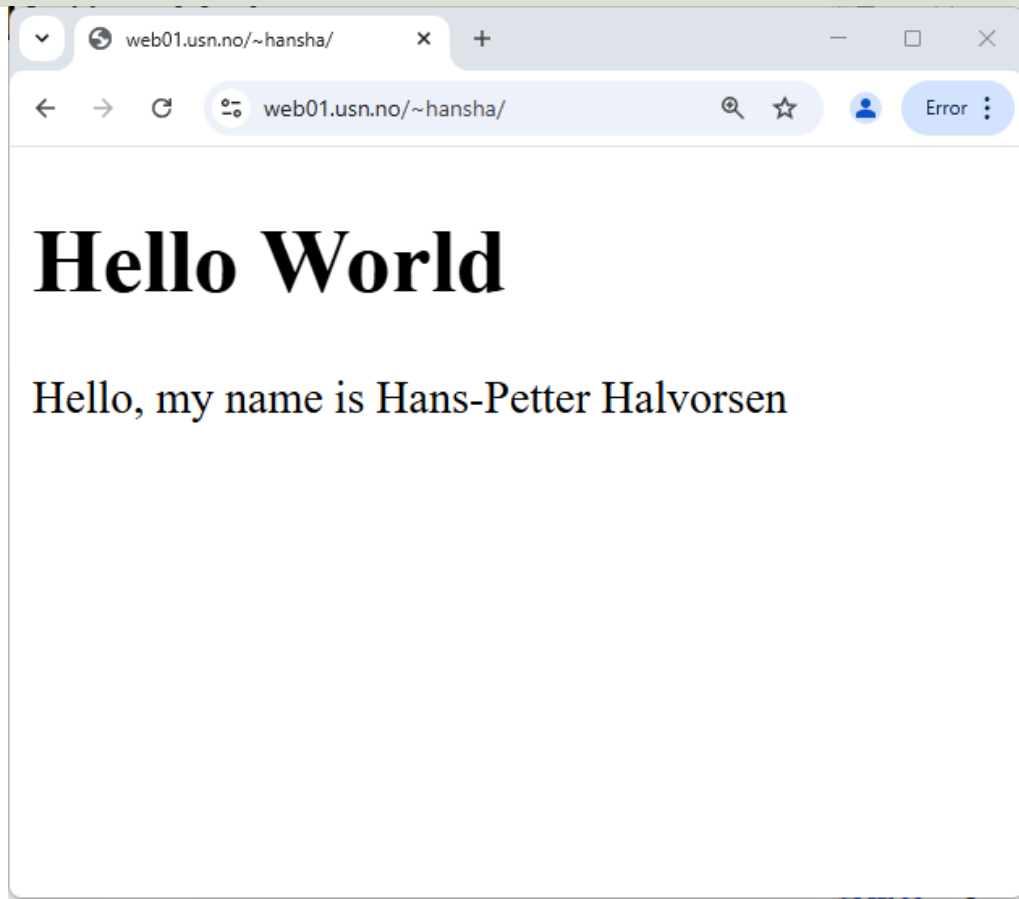
# Hello World App



A screenshot of a code editor window. The title bar shows 'index.php' and a search bar. The file path is 'C:\Temp\PHP > index.php'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 
4 <body>
5 
6 <h1>Hello World</h1>
7 
8 <p>
9 <?php
10 $name ="Hans-Petter Halvorsen";
11 echo "Hello, my name is $name";
12 ?>
13 </p>
14 
15 </body>
16 
17 </html>
```

The status bar at the bottom shows 'Ln 3, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'PHP'.





# PHP

PHP is a programming language with all the functionalities that a standard programming language has, like:

- Variables, Data Types, Arrays, etc.
- If.. Else.., While Loops, For Loops, etc.
- Functions, Classes and OOP.
- In addition, PHP has lots of web specific functionality.
- PHP has also builtin support for Databases, i.e., inserting and retrieving data from different database systems.

Here we will focus on the specific web features and database features, and not plain programming, since this is like C++/C# and any other programming language.

# Variables in PHP

- Variables in PHP starts with **\$**.
- **echo** is a built-in function in PHP that is much used to output contents of a variable to the web browser. You can also use the **print()** function.
- Note! Variables in PHP is case-sensitive!
- You don't need to define the datatype of the variable in PHP (PHP is a so-called loosely typed language)

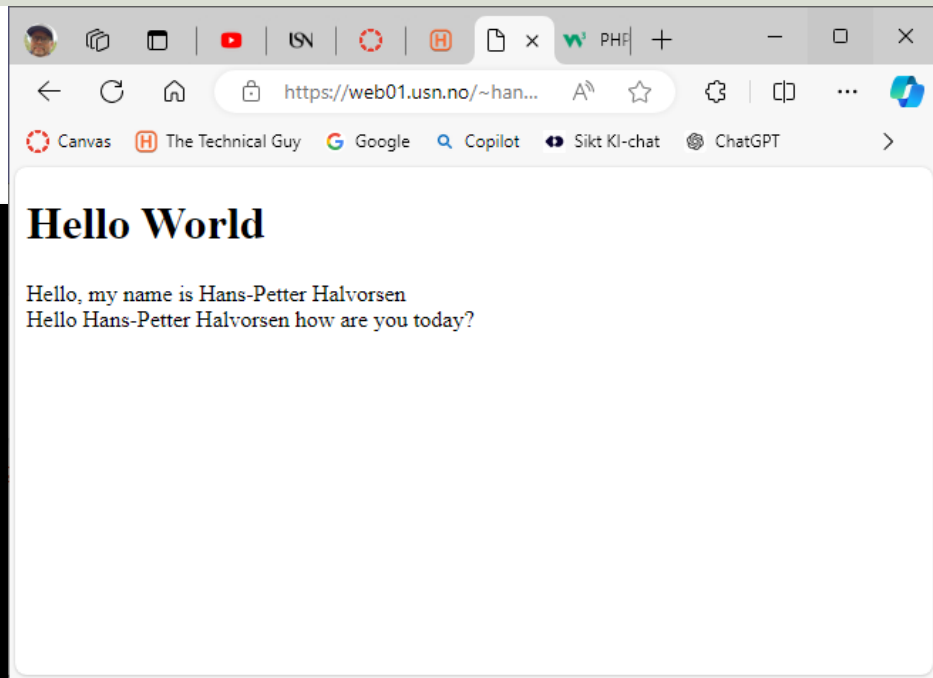
# Combining Text and Variables

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello World</h1>
    <?php
      $name ="Hans-Petter Halvorsen";

      echo "Hello, my name is $name";

      echo "<br>";

      echo 'Hello ' . $name . ' how are you today?';
    ?>
  </body>
</html>
```



# Combining Text and Variables

There is a huge difference between **double quotes (")** and **single quotes (')** in PHP:

Here you see some examples:

```
$name ="Hans-Petter Halvorsen";  
echo "Hello, my name is $name";
```

**Double quotes (") :**

Here will \$name be treated as a variable


```
$name ="Hans-Petter Halvorsen";  
echo 'Hello, my name is $name';
```

**Single quotes ('):**

Here will \$name just be part of the string

```
$name ="Hans-Petter Halvorsen";  
  
echo 'Hello ' . $name . ' how are you today?';
```

If using single quotes ('), you can use **. \$name .** to treat \$name as a variable.



```
1 <?DOCTYPE html>
2 <html>
3   <body>
4     <h1>Hello World</h1>
5     <?php
6       $name = "Hans-Petter Halvorsen";
7
8       echo "Hello, my name is $name";
9
10      echo "<br>";
11
12      echo 'Hello ' . $name . ' how are you today?';
13    ?>
14  </body>
15 </html>
```

1 PHP code server-side

Final Result shown  
in the Web Browser

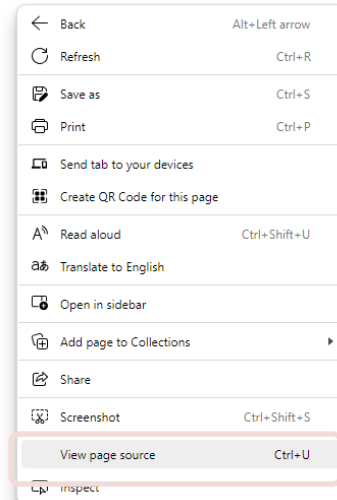
3  
**Hello World**

Hello, my name is Hans-Petter Halvorsen  
Hello Hans-Petter Halvorsen how are you today?

Line wrap ☐

2  
Generated HTML File that is  
sent to the Client (Web Browser)

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Hello World</h1>
5     Hello, my name is Hans-Petter Halvorsen<br>Hello Hans-Petter Halvorsen how are you today?
6   </body>
7 </html>
```



# String Function in PHP

PHP has many useful built-in functions for string manipulation, e.g.:

- `strlen()`
- `str_word_count()`
- `str_replace()`
- `strpos()`
- `substr()`
- `strrev()`
- `trim()`
- `strtoupper()`
- `strtolower()`
- `+++`

You will probably need to use many of these String function in your PHP Web Applications daily.

<https://www.php.net/manual/en/ref.strings.php>

# Comments in PHP

```
..  
// Single line comment  
/* Multiline comment  
..  
..  
..  
*/
```

It is good practice to add and use comments inside your code

Using comments are also a good “Debugging technique” by commenting out one or more code lines and make those are not executed.

# If .. Else, Loops, etc.

- PHP has built-in functionality for Conditions and Loops as other programming languages.
- The syntax is very similar to C/C#.
- PHP supports many different types
- Here you see a basic example:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Getting Started with PHP</h1>
    <?php
      $number = 18;

      if ($number > 10)
      {
        echo "The number is larger than 10";
      }
      else
      {
        echo "The number is smaller than 10";
      }
    ?>
  </body>
</html>
```



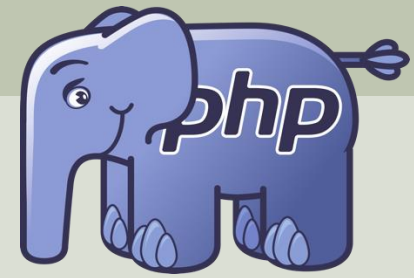
# Functions

PHP has many useful built-in function, but you can of course also make your own functions

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Getting Started with PHP</h1>
    <?php
      function FindAverage($number1, $number2)
      {
        $average = ($number1 + $number2)/2;
        return $average;
      }

      $x = 6;
      $y = 12;
      $mean = FindAverage($x, $y);
      echo $mean
    ?>
  </body>
</html>
```

<https://www.halvorsen.blog>



PHP Tutorial

# HTML Forms in PHP



Hans-Petter Halvorsen

[Table of Contents](#)

# HTML Forms

Example of a Web Page that contains an HTML Form:

## New Book

Please enter book information:

Title:

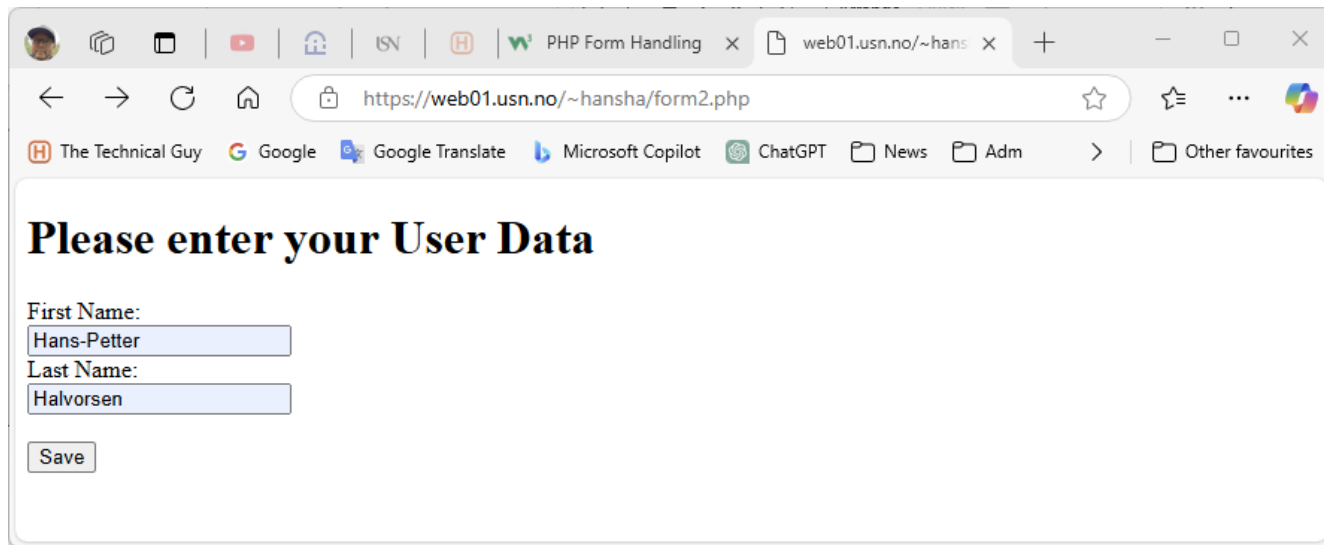
Author:

Topic:

When the user clicks on the Save button, the page will be sent back to the server. Then it is possible to store the data in a database, etc.

# HTML Forms

We start by creating a basic HTML Form like this:



The screenshot shows a web browser window with the address bar displaying `https://web01.usn.no/~hansha/form2.php`. The browser's tab bar shows two tabs: 'PHP Form Handling' and 'web01.usn.no/~hans...'. The browser's address bar includes navigation buttons (back, forward, refresh, home) and a search bar. Below the address bar, there is a row of search engines and services: 'The Technical Guy', 'Google', 'Google Translate', 'Microsoft Copilot', 'ChatGPT', 'News', 'Adm', and 'Other favourites'. The main content area of the browser displays a form titled 'Please enter your User Data'. The form contains two text input fields: 'First Name:' with the value 'Hans-Petter' and 'Last Name:' with the value 'Halvorsen'. Below these fields is a 'Save' button.

**Please enter your User Data**

First Name:

Last Name:

# HTML Forms

- An HTML form is typically used to collect data from the user.
- Then typically the data will then be sent to the server for processing and storage, e.g., in a Database like MySQL.

```
<!DOCTYPE html>
<html>
<body>

<h1>Please enter your User Data</h1>

<form action="showdata.php" method="POST">
  <label for="firstName">First Name:</label><br>
  <input type="text" id="firstName" name="firstName">
  <br>

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

  <input type="submit" value="Save">
</form>

</body>
</html>
```

# Forms and POST Example

```
<!DOCTYPE html>
<html>
<body>

<h1>Please enter your User Data</h1>

<form action="showdata.php" method="POST">
  <label for="firstName">First Name:</label><br>
  <input type="text" id="firstName" name="firstName">
  <br>

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

  <input type="submit" value="Save">
</form>

</body>
</html>
```

Please enter your User Data

First Name:

Hans Petter

Last Name:

Halvorsen

Save

angular Snip

```
<!DOCTYPE html>
<html>
<body>

<h1>Information about your User Data</h1>

Your First Name is <?php echo $_POST["firstName"]; ?>
<br>

Your Last Name is: <?php echo $_POST["lastName"]; ?>

</body>
</html>
```

showdata.php

**Information about your User Data**

Your First Name is Hans Petter

Your Last Name is: Halvorsen

We use **`$_POST`** to get the Form Data

# GET Example

We can also **`$_GET`**

```
form2.php x showdata2.php
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > form2.php
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Please enter your User Data</h1>
6
7 <form action="showdata2.php" method="GET">
8   <label for="firstName">First Name:</label><br>
9   <input type="text" id="firstName" name="firstName">
10  <br>
11
12   <label for="lastName">Last Name:</label><br>
13   <input type="text" id="lastName" name="lastName">
14   <br><br>
15
16   <input type="submit" value="Save">
17 </form>
18
19
20 </body>
21 </html>
```

```
form2.php x showdata2.php x
C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > HelloWorld > showdata2.php
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Information about your User Data</h1>
6
7 Your First Name is <?php echo $_GET["firstName"]; ?>
8 <br>
9
10 Your Last Name is: <?php echo $_GET["lastName"]; ?>
11
12 </body>
13 </html>
```

# GET and Query String

**Please enter your User Data**

First Name:  
Hans-Petter

Last Name:  
Halvorsen

Save

https://web01.usn.no/~hansha/form2.php

Note! Here we can also manually change the Query String Data.

**Information about your User Data**

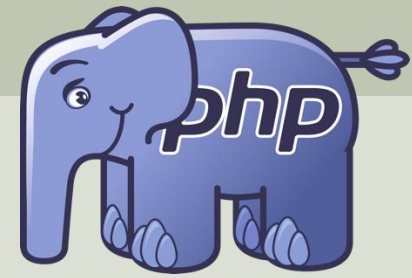
Your First Name is Hans-Petter  
Your Last Name is: Halvorsen

https://web01.usn.no/~hansha/showdata2.php?firstName=Hans-Petter&lastName=Halvorsen

When using **GET** the values will be sent as part of the URL/Query String. This alternative may not be advisable for secret information like passwords, etc.



<https://www.halvorsen.blog>



PHP Tutorial

# Session Variables in PHP



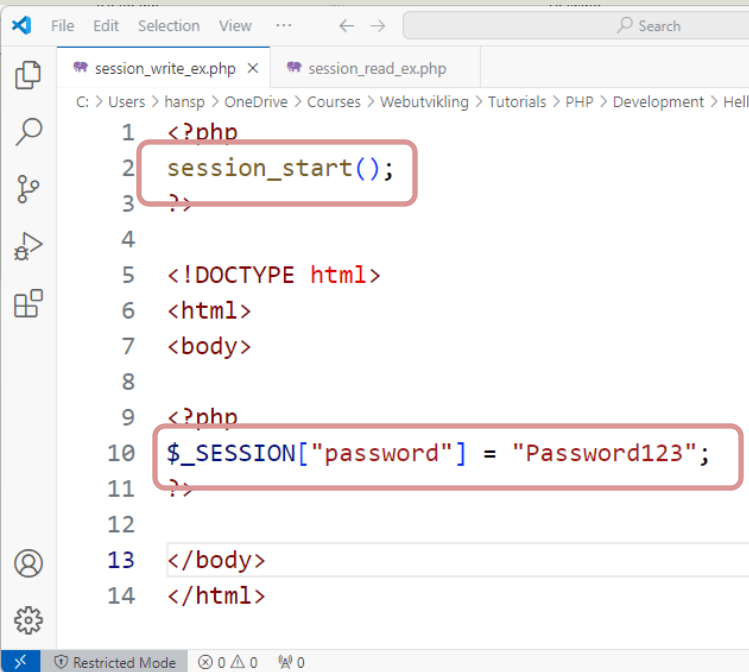
Hans-Petter Halvorsen

[Table of Contents](#)

# Session Variables

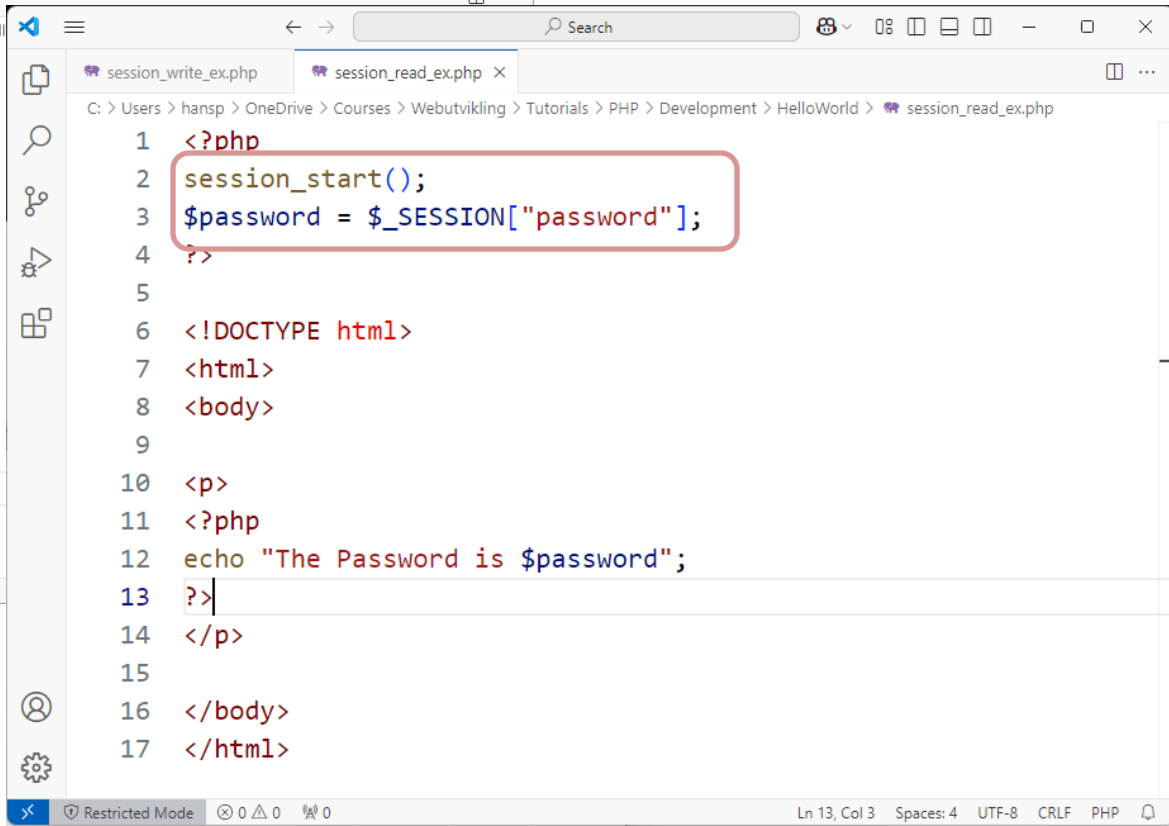
- We can use something called Session variables in order to **send data between 2 web pages**.
- Unlike a cookie, the information is not stored on the users PC.
- Session variables are very handy in web development.
- Session variables hold information only for one single user (you), so this means the information is only available for you and not for other users of the web page or web application.
- Sessions has a time limitation, typically 20 minutes.
- All web frameworks have these Sessions variables.

# Session Example



A screenshot of a code editor window titled 'session\_write\_ex.php'. The editor shows a PHP script that starts with a PHP opening tag, calls `session_start();` on line 2, and then sets a session variable `$_SESSION["password"] = "Password123";` on line 10. The script is enclosed in an HTML document structure with `<!DOCTYPE html>`, `<html>`, and `<body>` tags. The `session_start();` and the assignment line are highlighted with red boxes. The status bar at the bottom indicates 'Restricted Mode' and '0 0 0'.

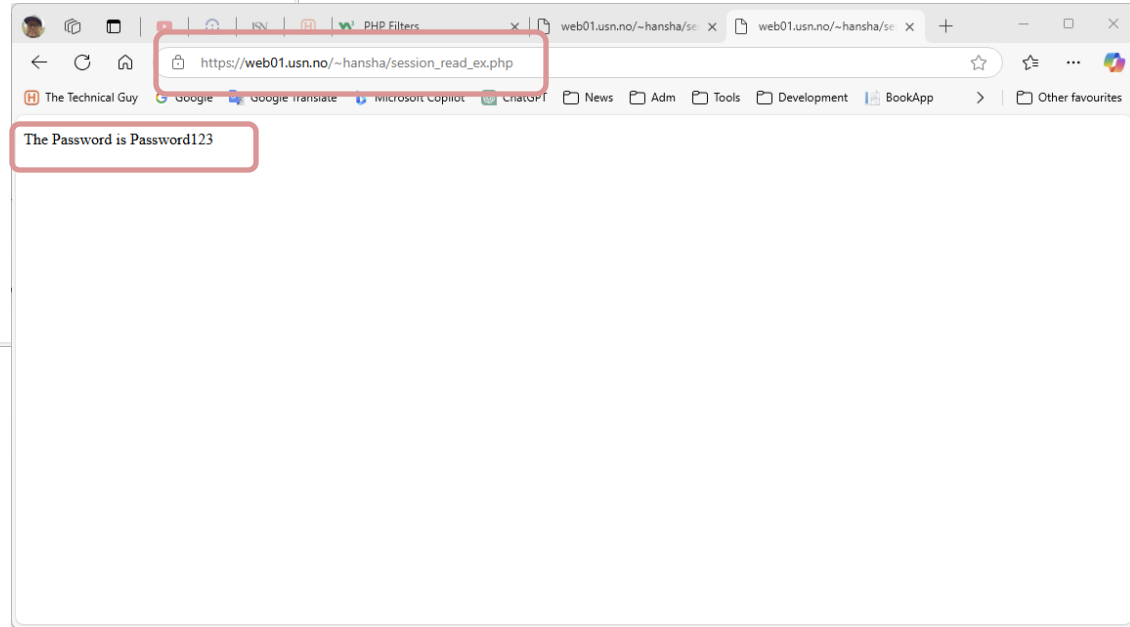
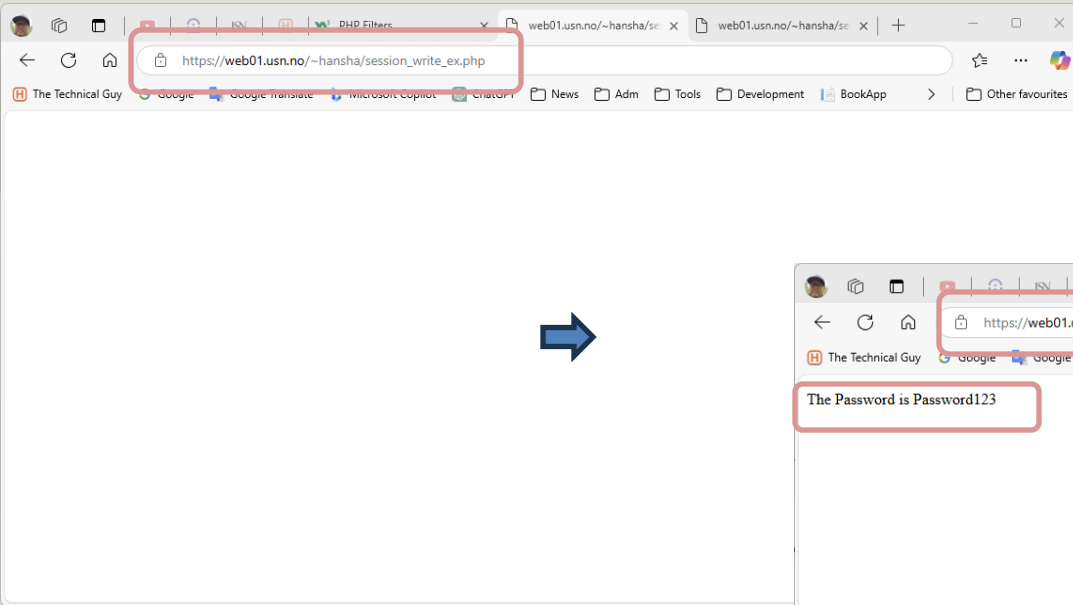
```
1 <?php
2 session_start();
3 ?>
4
5 <!DOCTYPE html>
6 <html>
7 <body>
8
9 <?php
10 $_SESSION["password"] = "Password123";
11 ?>
12
13 </body>
14 </html>
```



A screenshot of a code editor window titled 'session\_read\_ex.php'. The editor shows a PHP script that starts with a PHP opening tag, calls `session_start();` on line 2, and then retrieves a session variable `$password = $_SESSION["password"];` on line 3. The script is enclosed in an HTML document structure with `<!DOCTYPE html>`, `<html>`, and `<body>` tags. The `session_start();` and the retrieval line are highlighted with red boxes. The script continues with an echo statement `echo "The Password is $password";` on line 12, followed by closing tags for `<p>`, `<body>`, and `<html>`. The status bar at the bottom indicates 'Restricted Mode' and '0 0 0'.

```
1 <?php
2 session_start();
3 $password = $_SESSION["password"];
4 ?>
5
6 <!DOCTYPE html>
7 <html>
8 <body>
9
10 <p>
11 <?php
12 echo "The Password is $password";
13 ?>
14 </p>
15
16 </body>
17 </html>
```

# Session Example



# Session Example #2

web01.usn.no/~hansha/enter\_p x +

web01.usn.no/~hansha/enter\_password.php

## Please enter your Password

Password:

Verify



web01.usn.no/~hansha/verify\_p x +

web01.usn.no/~hansha/verify\_password.php

## User Information

Here [here](#) to see secret information.



web01.usn.no/~hansha/show\_in x +

web01.usn.no/~hansha/show\_information.php

## User Information

Here is secret information presented:

Your Code to the secret locker is 1234.

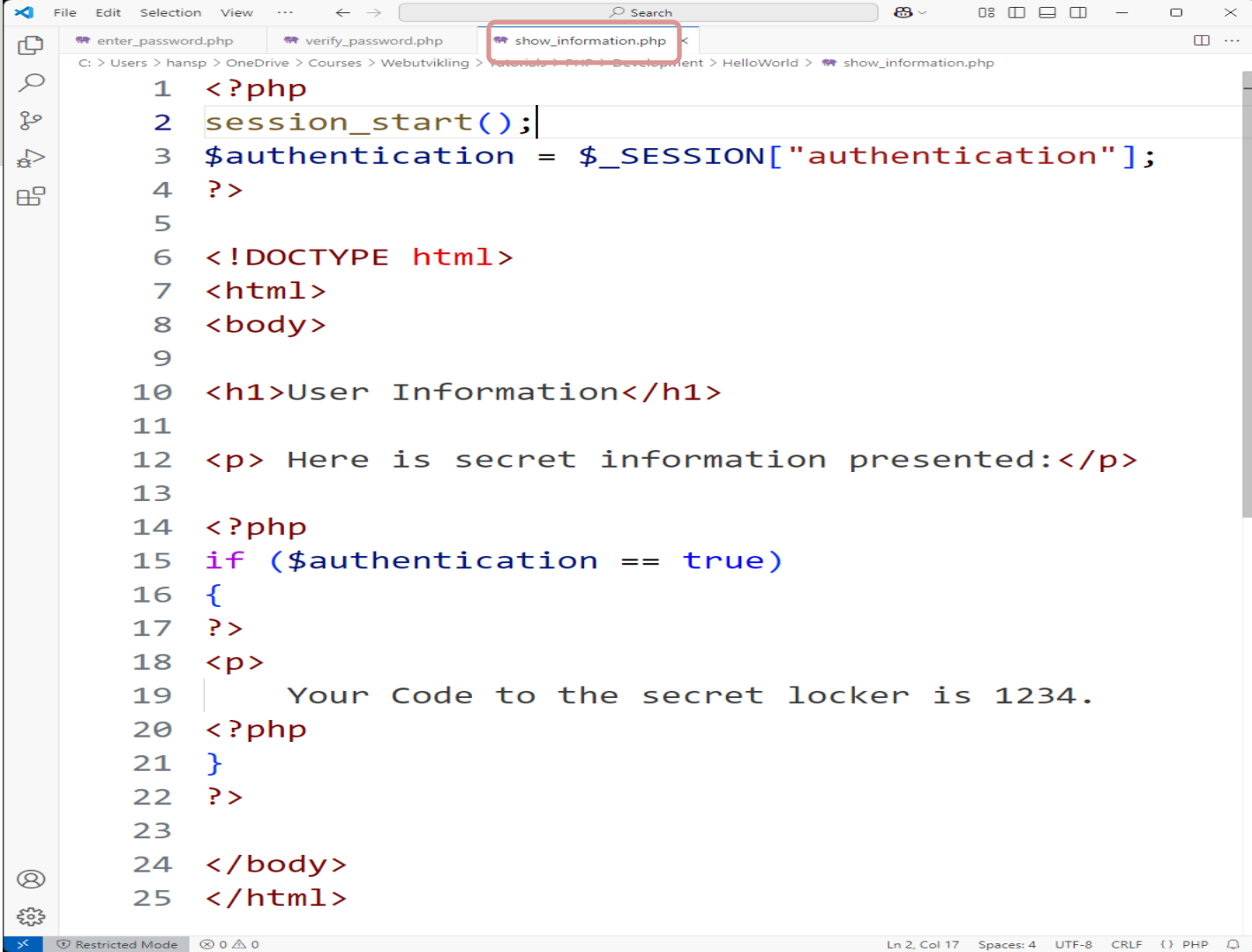


```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Please enter your Password</h1>
6
7 <form action="verify_password.php" method="POST">
8
9     <label for="password">Password:</label><br>
10    <input type="password" id="password" name="password">
11    <br>
12
13    <input type="submit" value="Verify">
14 </form>
15
16 </body>
17 </html>
```

Ln 17, Col 8 Spaces: 4 UTF-8 CRLF () PHP

```
File Edit Selection View Search
enter_password.php verify_password.php show_information.php
C:\Users\hansp>OneDrive\Documents\Programming\Tutorials>PHP>Development>HelloWorld>verify_password.php

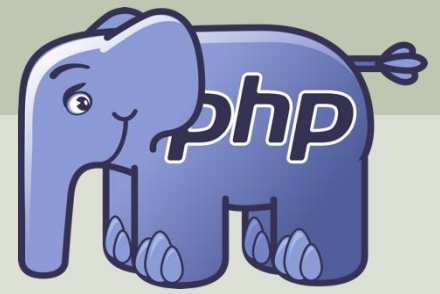
1 <?php
2 session_start();
3 $password = $_POST["password"];
4
5 $authentication = false;
6 $_SESSION["authentication"] = false;
7 if ($password == "Password123")
8 {
9     $authentication = true;
10    $_SESSION["authentication"] = true;
11 }
12 ?>
13 <!DOCTYPE html>
14 <html>
15 <body>
16 <h1>User Information</h1>
17
18 <?php
19 if ($authentication == true)
20 {
21 ?>
22 <p> Here <a href="show_information.php">here</a> to see secret information.</p>
23 <?php
24 }
25 else
26     echo "No information is available.";
27 ?>
28
29 </body>
30 </html>
```



```
1 <?php
2 session_start();
3 $authentication = $_SESSION["authentication"];
4 ?>
5
6 <!DOCTYPE html>
7 <html>
8 <body>
9
10 <h1>User Information</h1>
11
12 <p> Here is secret information presented:</p>
13
14 <?php
15 if ($authentication == true)
16 {
17 ?>
18 <p>
19     Your Code to the secret locker is 1234.
20 <?php
21 }
22 ?>
23
24 </body>
25 </html>
```



<https://www.halvorsen.blog>



PHP Tutorial

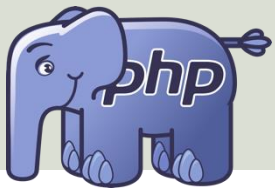
# PHP and MySQL Database



Hans-Petter Halvorsen



[Table of Contents](#)

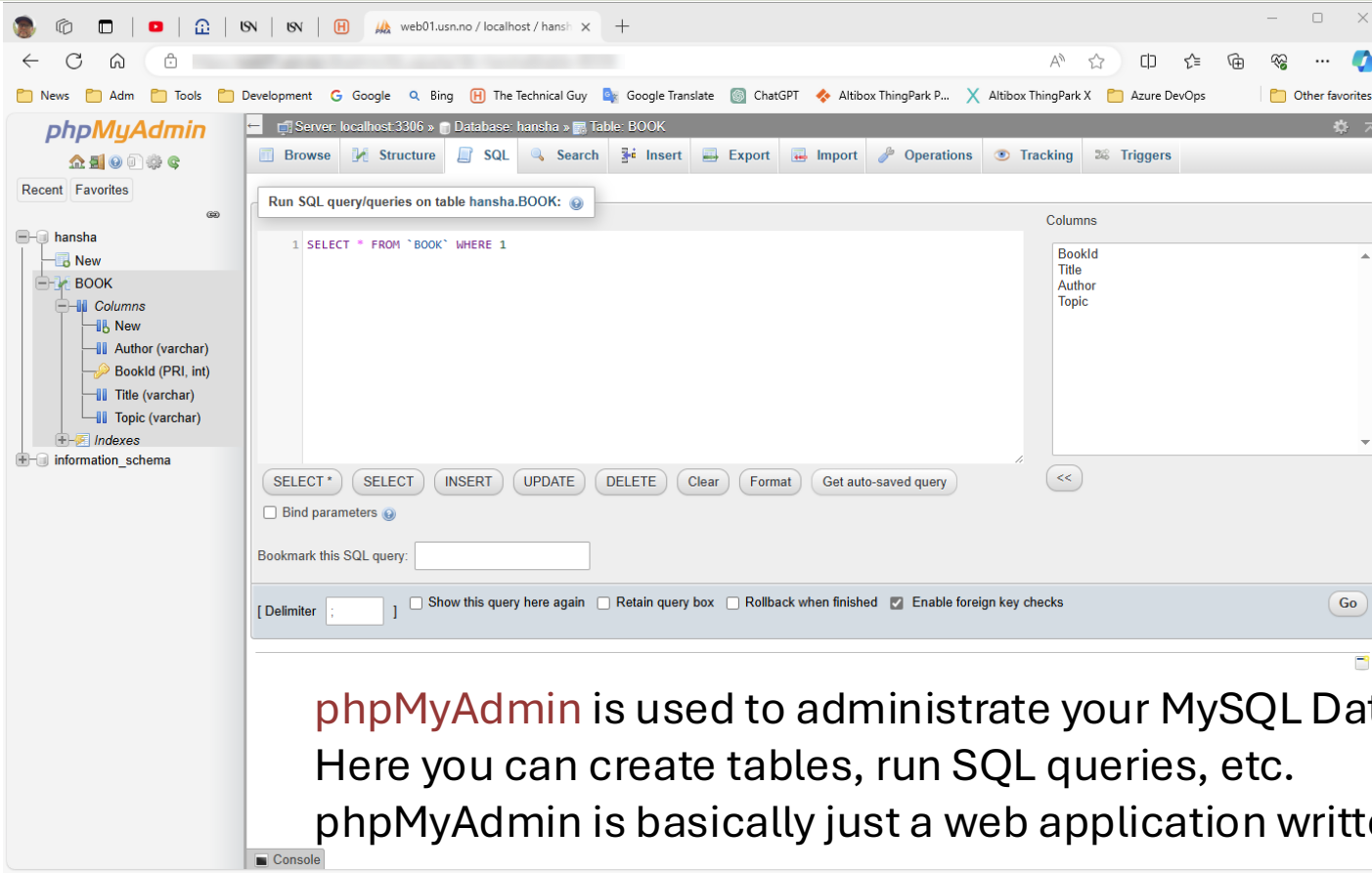


# MySQL and PHP



- **MySQL** is a popular relational database system.
- It is free and open source.
- MySQL uses **SQL** (Structured Query Language)
- The combination of **PHP** and MySQL is very popular and powerful.
- You can create powerful web applications where you can show, save, update and delete data in a MySQL database from the PHP code.

# phpMyAdmin



The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL `web01.usn.no / localhost / hansha`. The interface includes a sidebar on the left with a tree view of the database structure, showing the 'hansha' database with tables 'New', 'BOOK', and 'Columns'. The 'BOOK' table is selected, and its columns are listed: 'BookId (PRI, int)', 'Author (varchar)', 'Title (varchar)', and 'Topic (varchar)'. The main panel shows the 'SQL' tab with a query editor containing the query `1 SELECT * FROM `BOOK` WHERE 1`. Below the query editor are buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', and 'Get auto-saved query'. There is also a checkbox for 'Bind parameters' and a 'Bookmark this SQL query' field. At the bottom, there are checkboxes for 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks', along with a 'Go' button. The browser's address bar and tabs are visible at the top.

phpMyAdmin is used to administrate your MySQL Database. Here you can create tables, run SQL queries, etc. phpMyAdmin is basically just a web application written in PHP.

# Connect to the Database

There are 2 different methods that you can use to connect to your MySQL Database from PHP:

- **MySQLi** – Only works together with MySQL
- **PDO** – This option will also work for many other types of database systems.

# Open Connection to Database

In this tutorial we will use **MySQLi**. Here you see an example how we can connect to the database:

```
<?php
$servername = "localhost";
$dbname = "dbname";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully.";
?>
```

Close Connection after we have communicated with the database:

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

# CRUD

Typically, we want to do the following operations:

- **C**reate (Insert) Data
- **R**ead (Select) Data
- **U**ppdate Data
- **D**eleete Data

=> This is referred to as CRUD functionality.

Typically, all Applications today need to communicate with a Database and have CRUD functionality.

When you have learned to create a basic CRUD Application, you have all the necessary tools you need to create any kind of Application.

# SQL

- Structured Query Language (SQL) is used to write, read and update data from Database Systems.
- SQL is a standardized language used by most database systems.
- You can use SQL inside the “phpMyAdmin” or/and inside your PHP Web Application.
- SQL Example: `select * from SCHOOL`

# SQL Query Examples

- **insert** into STUDENT (Name , Number, SchoolId)  
values ('John Smith', '100005', 1)
- **select** SchoolId, Name from SCHOOL
- **select** \* from SCHOOL where SchoolId > 100
- **update** STUDENT set Name='John Wayne' **where** StudentId=2
- **delete** from STUDENT **where** SchoolId=3

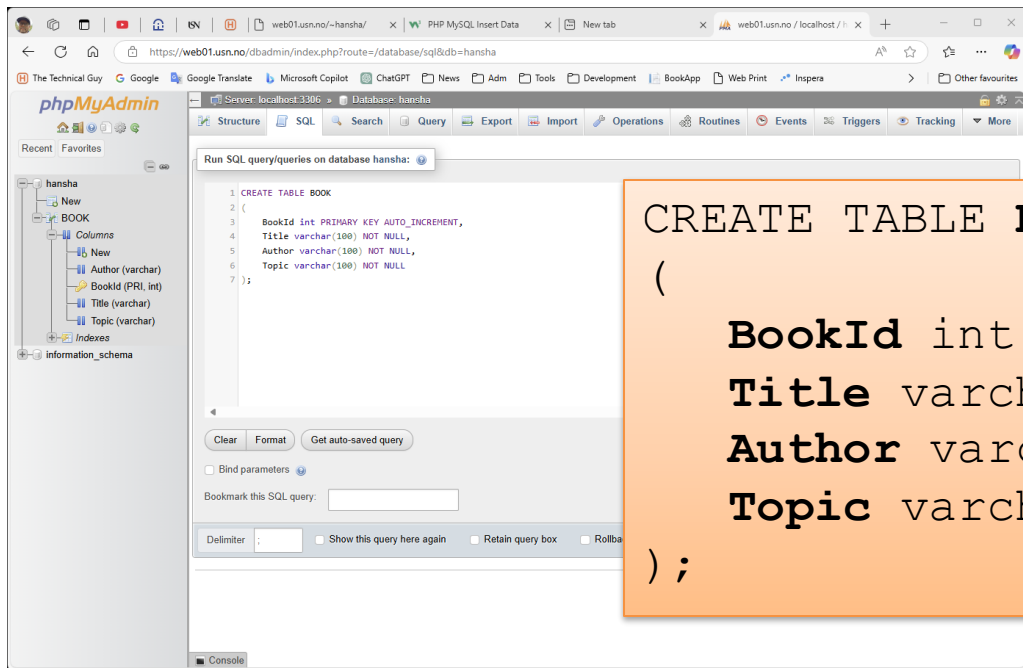
We have 4 different Query Types: **INSERT**, **SELECT**, **UPDATE** and **DELETE**

**CRUD**: **C** – Create or Insert Data, **R** – Retrieve (Select) Data, **U** – Update Data, **D** – Delete Data



# Create Database

We can create Databases and Database Tables using PHP. But typically, we create a Database and the necessary Tables in advance before we start coding the Web Application. We use the **phpMyAdmin** tool.



```
CREATE TABLE BOOK
(
    BookId int PRIMARY KEY AUTO_INCREMENT,
    Title varchar(100) NOT NULL,
    Author varchar(100) NOT NULL,
    Topic varchar(100) NOT NULL
);
```

# Database

We can also insert some data into the Table, e.g.:

```
insert into BOOK (Title, Author, Topic) values  
( 'Web Apps', 'Elvis Presly', 'Programming');
```

```
insert into BOOK (Title, Author, Topic) values  
( 'IoT and Cloud', 'John Wayne', 'IoT');
```

```
insert into BOOK (Title, Author, Topic) values  
( 'C#', 'Rune Hansen', 'Programming');
```

# PHP Config File

Typically, we want to hide the Connection to the database, so, we can put it into a separate PHP file called, e.g., “config.php”. Then in the different PHP files we can include this file. This file will contain username, password, etc. for the MySQL Server database.

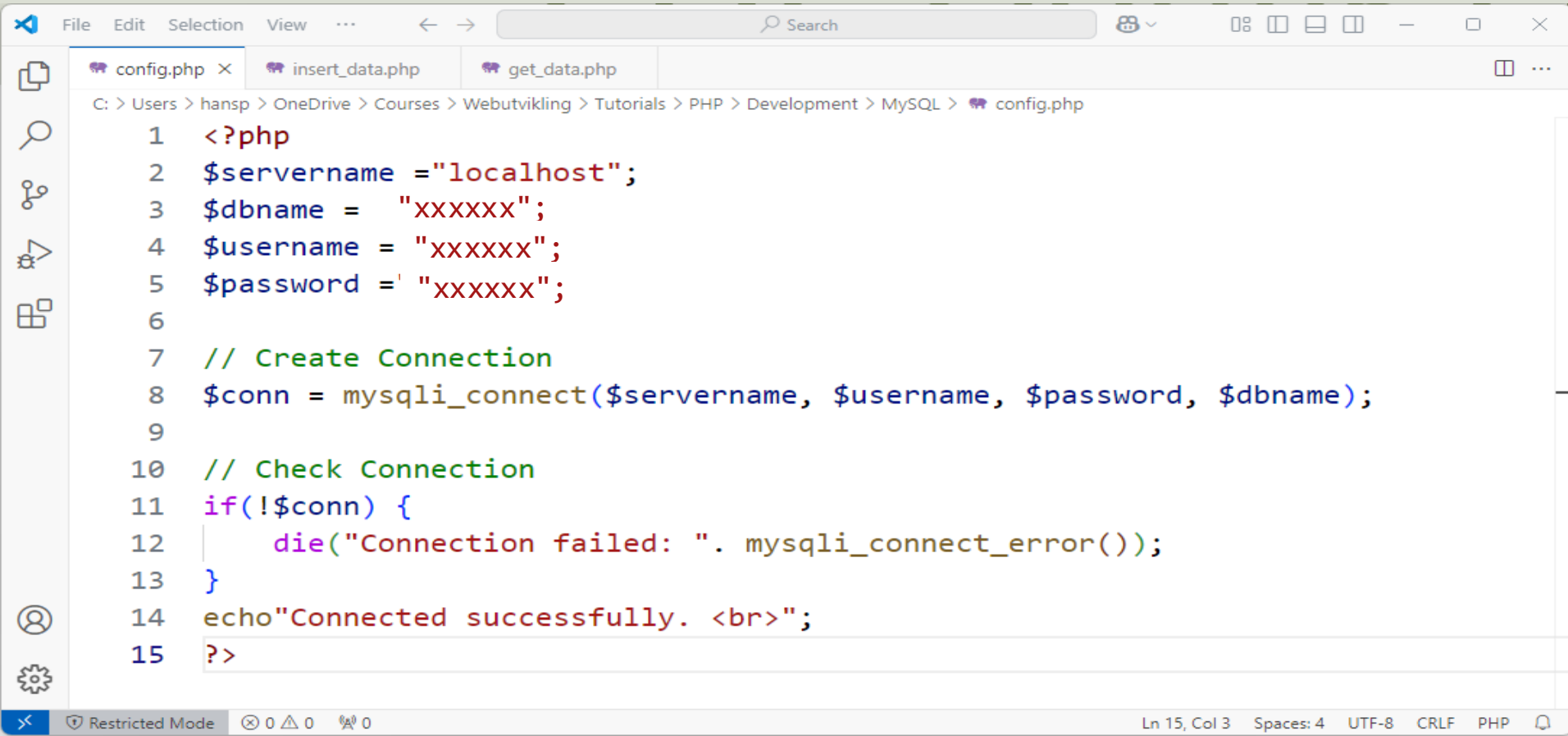
```
<?php
$servername = "localhost";
$dbname = "xxxxxx";
$username = "xxxxxx";
$password = "xxxxxx";

// Create Connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check Connection
if(!$conn) {
    die("Connection failed: ". mysqli_connect_error());
}
echo "Connected successfully.";
?>
```

**config.php**

# PHP Config File



The image shows a code editor window with a menu bar (File, Edit, Selection, View, ...), a search bar, and a tab bar. The active tab is 'config.php'. The breadcrumb path is 'C: > Users > hansp > OneDrive > Courses > Webutvikling > Tutorials > PHP > Development > MySQL > config.php'. The code is a PHP script for connecting to a MySQL database. It includes comments for creating and checking the connection. The status bar at the bottom shows 'Restricted Mode', 'Ln 15, Col 3', 'Spaces: 4', 'UTF-8', 'CRLF', 'PHP', and a refresh icon.

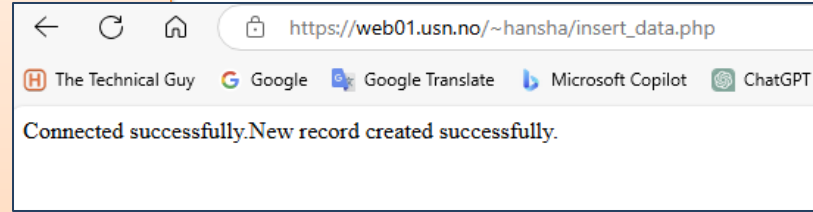
```
1  <?php
2  $servername = "localhost";
3  $dbname = "xxxxxx";
4  $username = "xxxxxx";
5  $password = "xxxxxx";
6
7  // Create Connection
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9
10 // Check Connection
11 if(!$conn) {
12     die("Connection failed: ". mysqli_connect_error());
13 }
14 echo"Connected successfully. <br>";
15 ?>
```

# Save Data to the Database

```
<?php
require_once 'config.php';
// Insert Data

$sql = "INSERT INTO BOOK (Title, Author, Topic)
VALUES ('AI', 'Allan Johnsen', 'Data')";

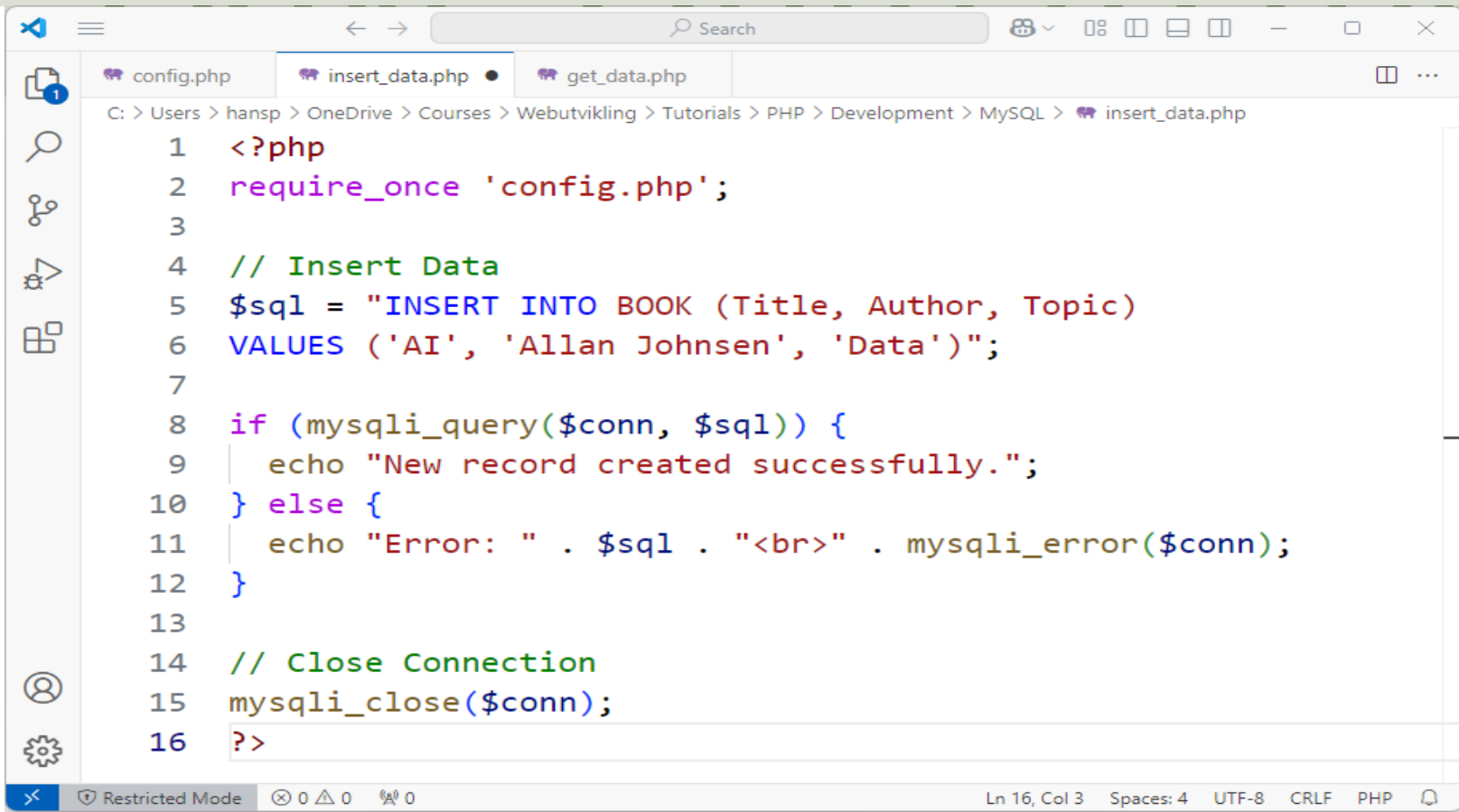
if (mysqli_query($conn, $sql)) {
    echo "New record created successfully.";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
// Close Connection
mysqli_close($conn);
?>
```



Normally you get these Data from an HTML Form where the user enters this information.

Then go to **phpMyAdmin** and check if the data has been stored in the database.

# Save Data to the Database



The screenshot shows a code editor window with three tabs: `config.php`, `insert_data.php` (active), and `get_data.php`. The active tab displays a PHP script for inserting data into a database. The script includes a `require_once` statement for `config.php`, an SQL `INSERT` statement, a conditional check for the query result, and a `mysqli_close` statement. The file path in the editor is `C:\> Users\hansp\OneDrive\Courses\Webutvikling\Tutorials\PHP\Development\MySQL\insert_data.php`. The status bar at the bottom indicates 'Restricted Mode' and provides statistics: 0 errors, 0 warnings, 0 suggestions, and 0 fixes. The current cursor position is at line 16, column 3.

```
1  <?php
2  require_once 'config.php';
3
4  // Insert Data
5  $sql = "INSERT INTO BOOK (Title, Author, Topic)
6  VALUES ('AI', 'Allan Johnsen', 'Data')";
7
8  if (mysqli_query($conn, $sql)) {
9      echo "New record created successfully.";
10 } else {
11     echo "Error: " . $sql . "<br>" . mysqli_error($conn);
12 }
13
14 // Close Connection
15 mysqli_close($conn);
16 ?>
```

Ln 16, Col 3 Spaces: 4 UTF-8 CRLF PHP

# Show Data from the Database

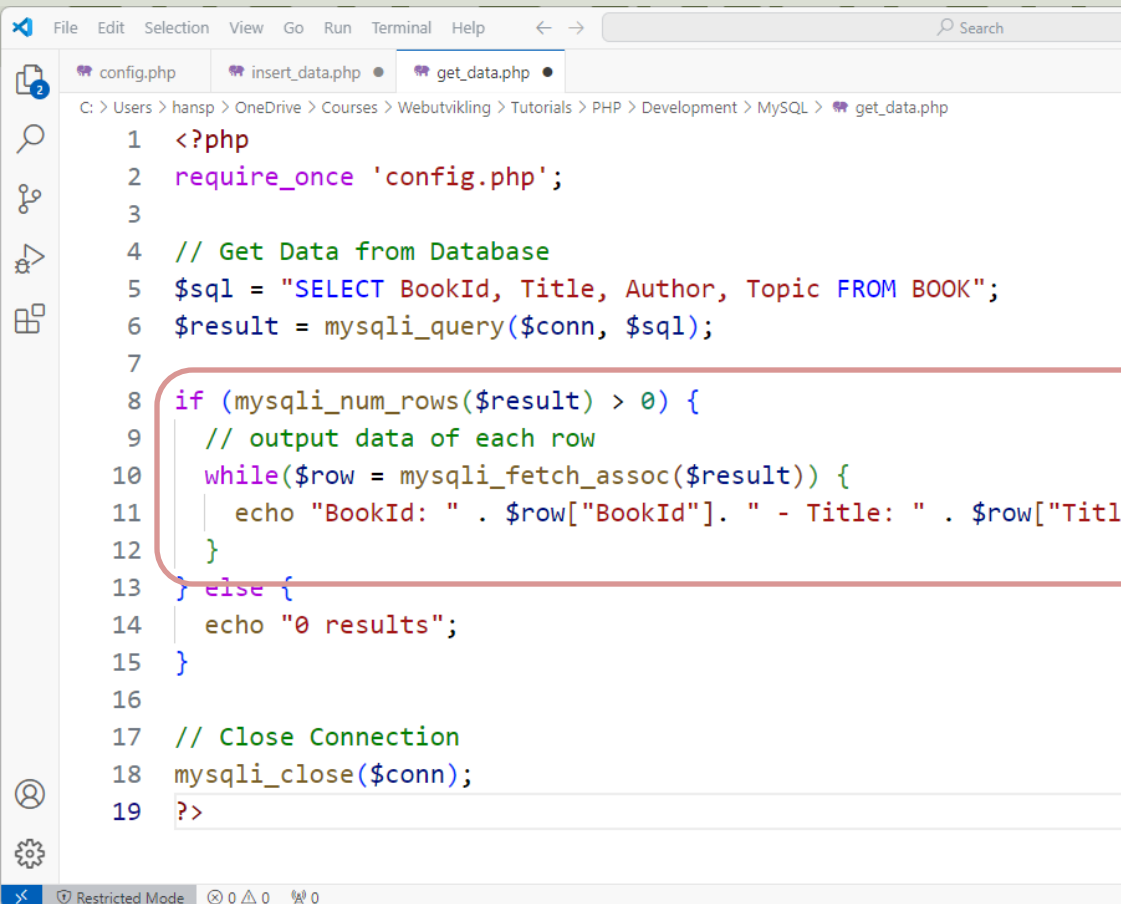
```
<?php
require_once 'config.php';

// Get Data from Database
$sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "BookId: " . $row["BookId"]. " - Title: " . $row["Title"]. " - Author: " .
$row["Author"]. " - Topic: " . $row["Topic"]. "<br>";
    }
} else {
    echo "0 results";
}

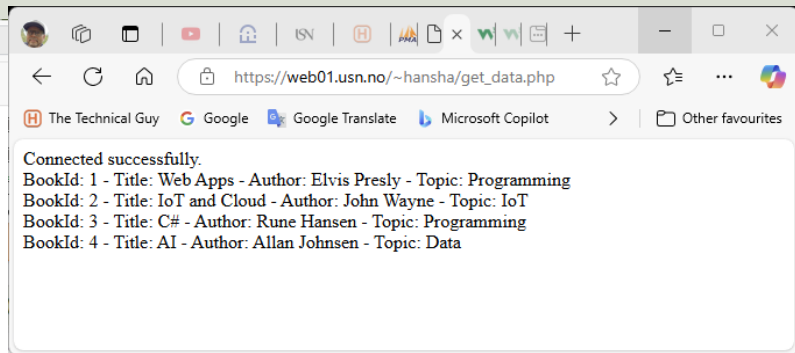
// Close Connection
mysqli_close($conn);
?>
```

# Show Data from the Database



The screenshot shows a code editor with a file explorer on the left and a search bar at the top. The active file is `get_data.php`. The code is a PHP script that connects to a MySQL database, queries a table named `BOOK`, and displays the results. A red box highlights the logic that checks for data and loops through the results.

```
1 <?php
2 require_once 'config.php';
3
4 // Get Data from Database
5 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
6 $result = mysqli_query($conn, $sql);
7
8 if (mysqli_num_rows($result) > 0) {
9     // output data of each row
10    while($row = mysqli_fetch_assoc($result)) {
11        echo "BookId: " . $row["BookId"]. " - Title: " . $row["Title"]. " - Author: " . $row["Author"]. " - Topic: " . $row["Topic"] . "<br>";
12    }
13 } else {
14     echo "0 results";
15 }
16
17 // Close Connection
18 mysqli_close($conn);
19 ?>
```





# HTML Table

web01.usn.no/~hansha/get\_data2.php

web01.usn.no/~hansha/get\_data2.php

Connected successfully.

BookId	Title	Author	Topic
2	IoT and Cloud	John Wayne	IoT
3	C#	Rune Hansen	Programming
4	AI	Allan Johnsen	Data

```
get_data2.php x
C: > Temp > PHP > get_data2.php
1  <?php
2  require_once 'config.php';
3  ?>
4
5  <!DOCTYPE html>|
6  <html>
7
8  <body>
9
10 <table border="1">
11 <thead>
12 <tr>
13 <th>BookId</th>
14 <th>Title</th>
15 <th>Author</th>
16 <th>Topic</th>
17 </tr>
18 </thead>
19
20 <tbody>
21
22 <?php
23 // Get Data from Database
24 $sql = "SELECT BookId, Title, Author, Topic FROM BOOK";
25 $result = mysqli_query($conn, $sql);
26
27 if (mysqli_num_rows($result) > 0) {
28 // output data of each row
29 while($row = mysqli_fetch_assoc($result)) {
30 echo "<tr>";
31 echo "<td>" . $row["BookId"] . "</td>";
32 echo "<td>" . $row["Title"] . "</td>";
33 echo "<td>" . $row["Author"] . "</td>";
34 echo "<td>" . $row["Topic"] . "</td>";
35 echo "</tr>";
36 }
37 } else {
38 echo "0 results";
39 }
40 ?>
41
42 </tbody>
43 </table>
44
45 </body>
46 </html>
47
48 <?php
49 // Close Connection
50 mysqli_close($conn);
51 ?>
```

# Resources and References

- PHP Tutorial w3school:  
<https://www.w3schools.com/php/>
- PHP Tutorial TutorialsPoint:  
<https://www.tutorialspoint.com/php/>
- PHP Documentation:  
<https://www.php.net/manual/en/>
- MySQL Tutorial:  
<https://www.w3schools.com/mysql>

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

