

# Backend Selektion=1;

Utbildare: Mahmud Al Hakim

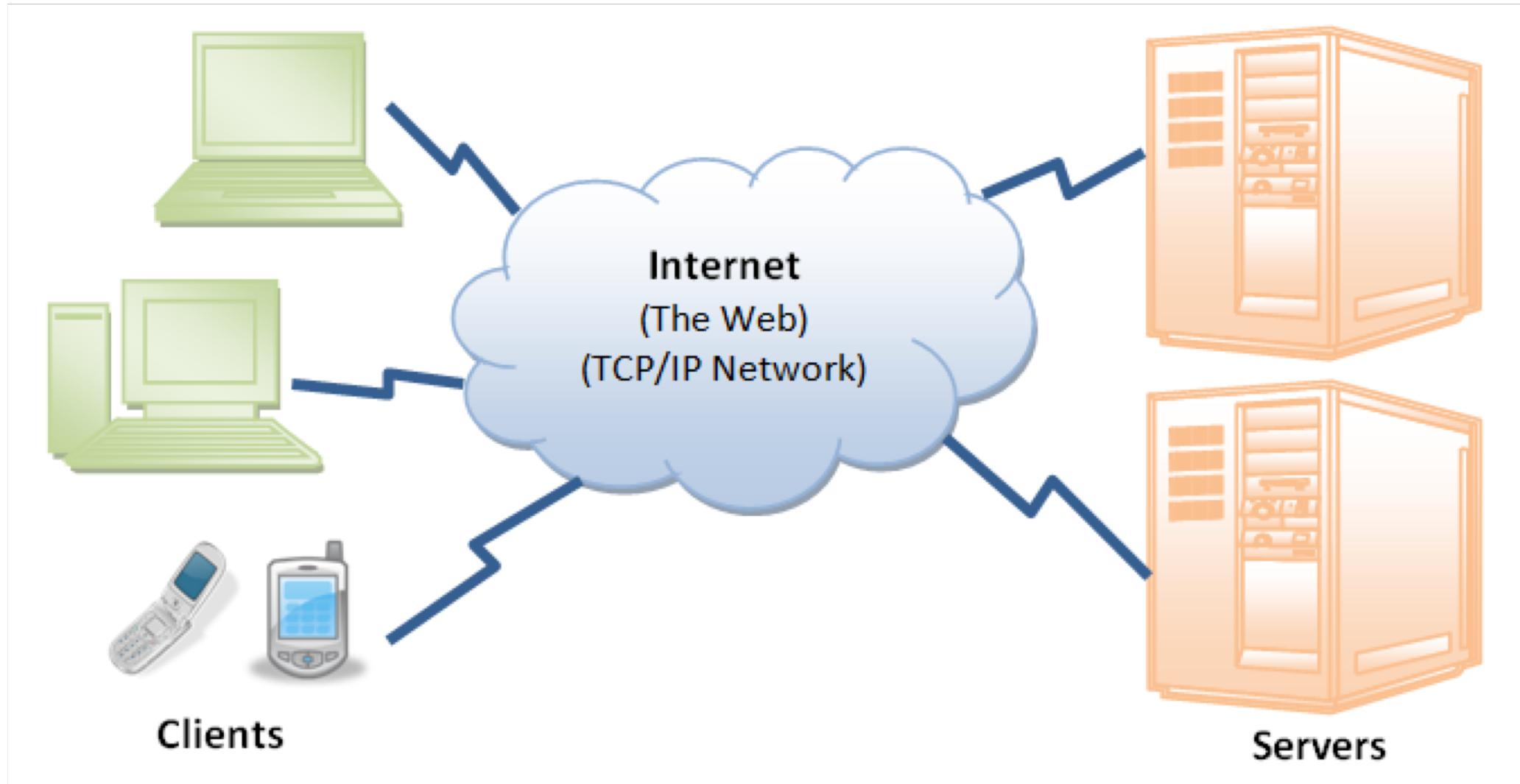
**NACKADEMIN**

# Lektionstillfällets mål

- **Introduktion till Webbserverprogrammering**
- MAMP: Portar, Dokument Root
- Apache Webbserver, localhost
- PHP.INI: display\_errors
- Webbhotell och Domännamn
- Namnserver, DNS
- FTP
- **Introduktion till PHP**
- Grundläggande syntax
- Variabler
- Indexerade arrayer
- Associativa arrayer
- Multidimensionella arrayer
- Arbeta med \$\_GET

**NACKADEMIN**

# Klient – Server



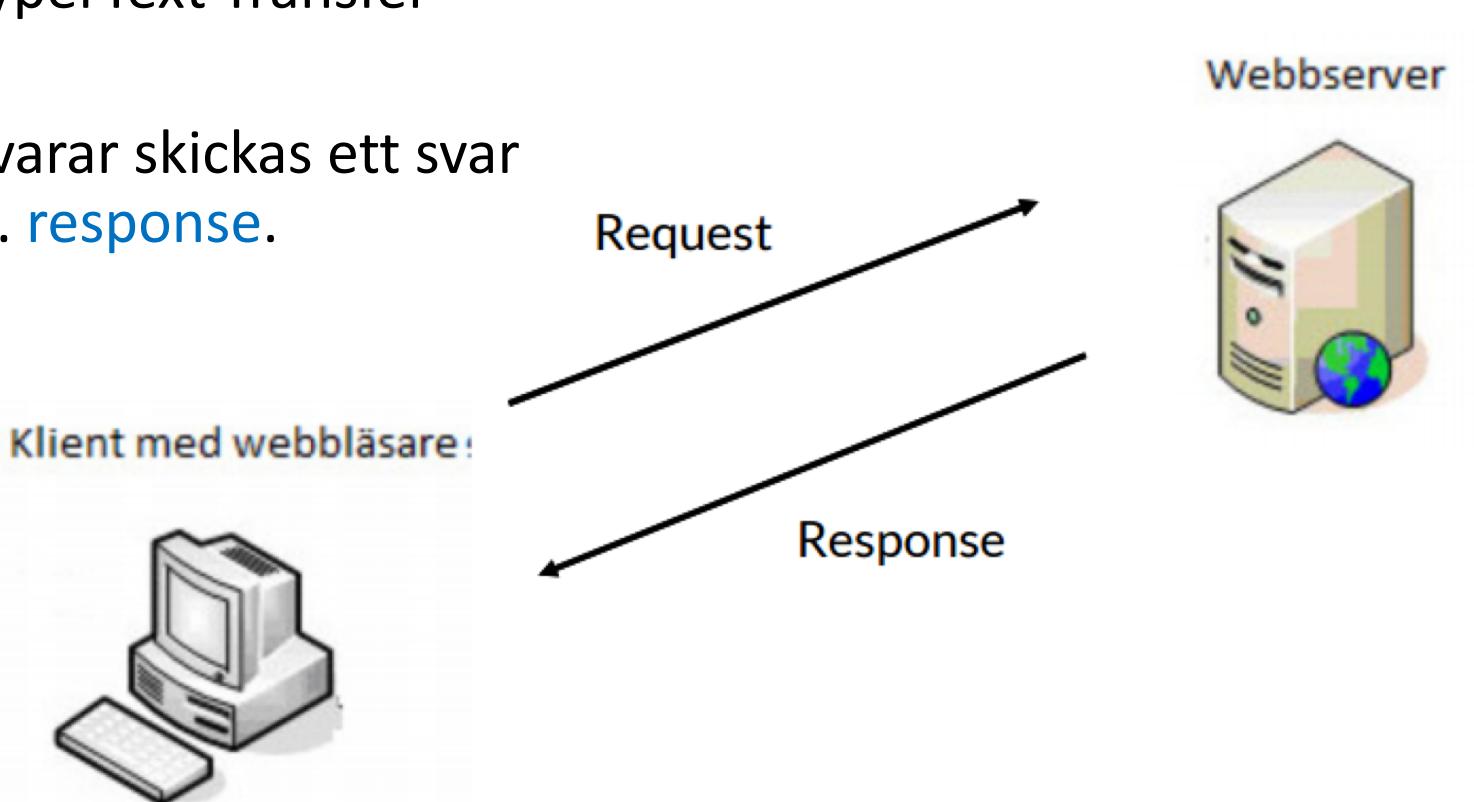
# Webbserver

- En webbserver är en uppkopplad dator på Internet och har en fast IP-adress.
- En webbserver innehåller en programvara som kan kommunicera via HTTP.
- Exempel på webbserverprogramvara
  - Apache (Open Source)
  - Nginx (Open Source)
  - IIS (Microsoft)
- Webbserverprogram kommunicerar vanligtvis via port 80.



# Request – Response

- När en klient gör en förfrågan till en webbserver skickas en fråga s.k. **request** över HTTP (HyperText Transfer Protocol).
- När servern svarar skickas ett svar tillbaka en s.k. **response**.



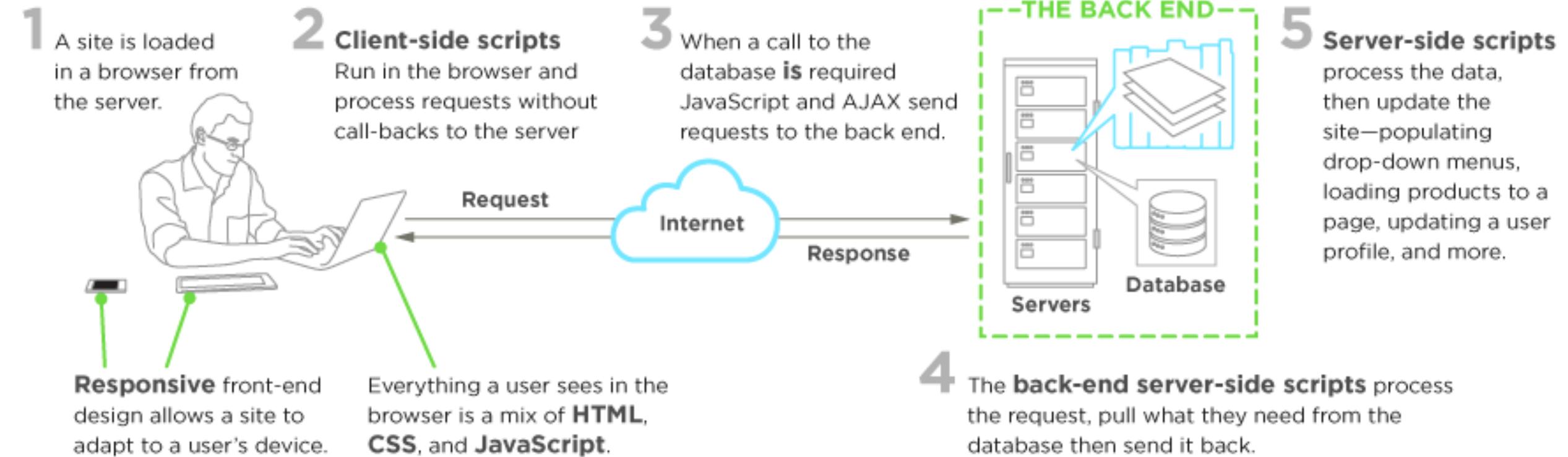
# Client-side scripting

- För att undvika onödig trafik mellan klient och server skapas skript med kod som körs i webbläsaren.
- Används ofta för att ge en bättre användarupplevelse.
- Det dominerande skriptspråket som körs i webbläsaren är **JavaScript**.

# Server-side scripting

- **Serverkod** innebär att programkod körs på servern och genererar HTML kod som skickas tillbaka till klienten.
- Detta gör sidorna dynamiska till skillnad från rena HTML-sidor som är statiska.
- Vad som kan göras är t.ex. att hämta data från databaser som skall visas på webbsidan.
- Det finns ett antal tekniker för detta.  
De vanligaste är **PHP**, Java och ASP.NET.

# Klientskript och Serverskript



# Webbserverprogrampaket

- Webbserverprogrampaket är en samlings program som används för att arbeta med Apache, PHP och databaser **lokalt**.
- Exempel på webbserverprogrampaket



MAMP

[www.mamp.info/en/](http://www.mamp.info/en/)

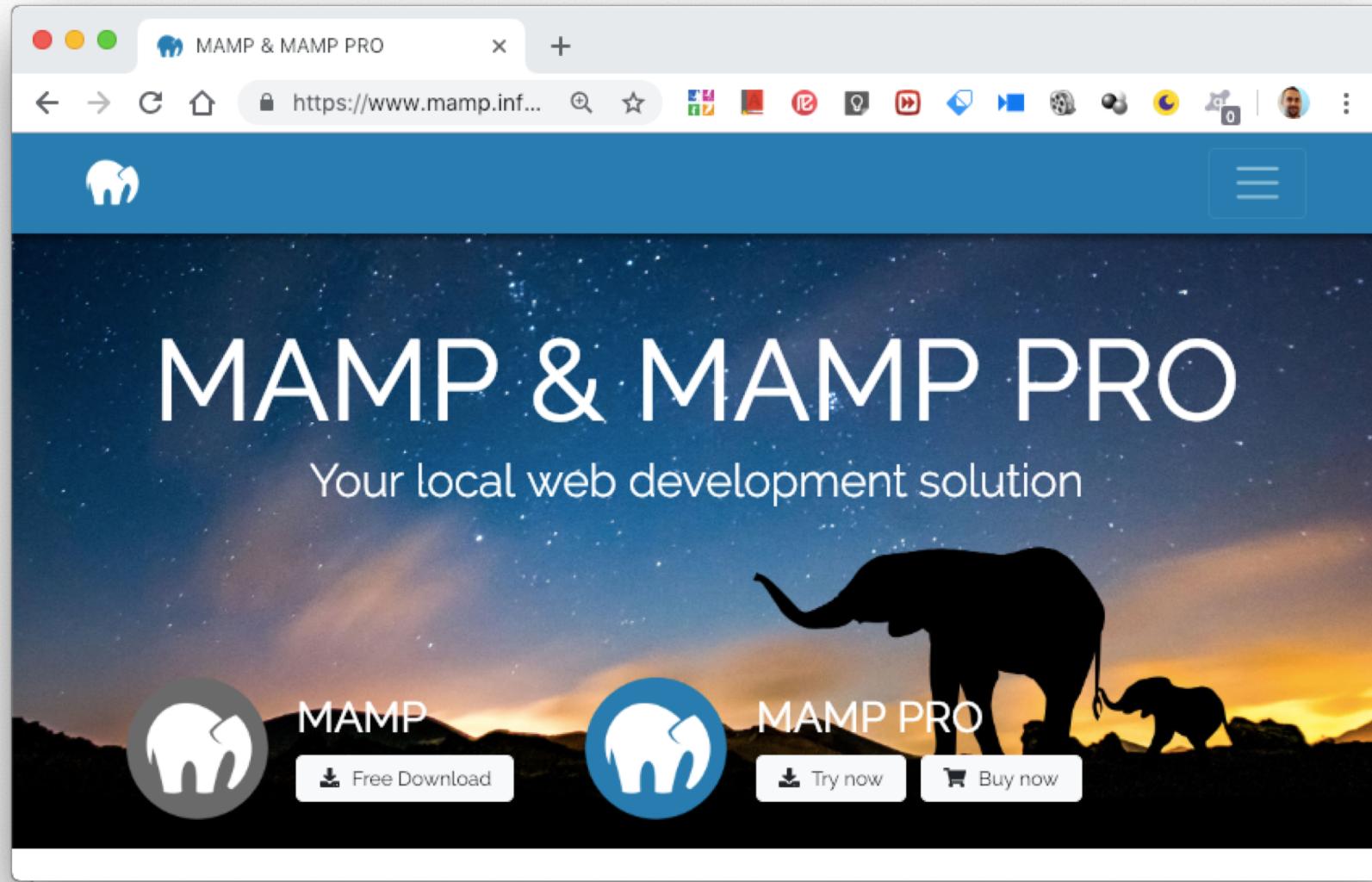


XAMPP

[www.apachefriends.org](http://www.apachefriends.org)

# Ladda ner och installera MAMP

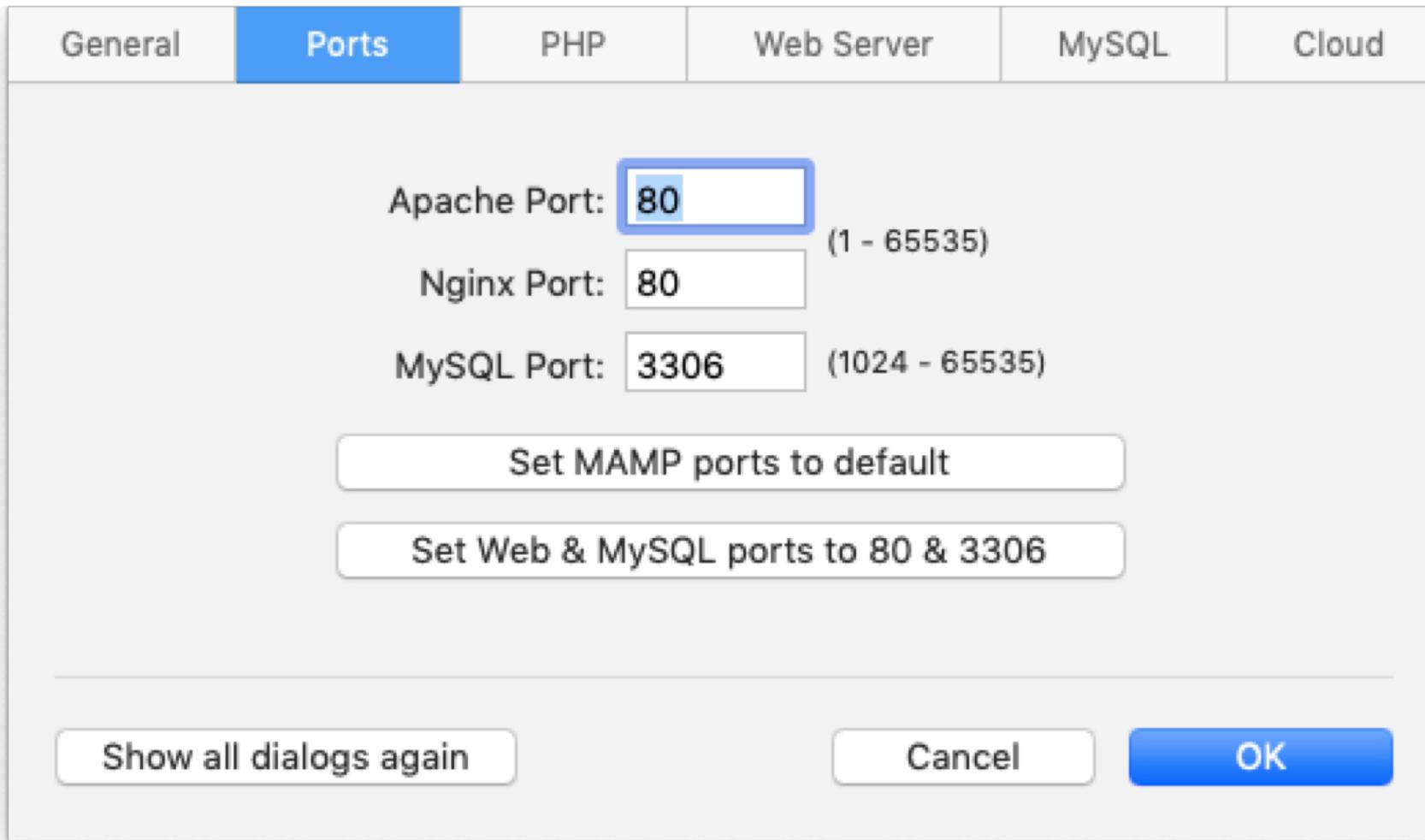
<https://www.mamp.info/en>



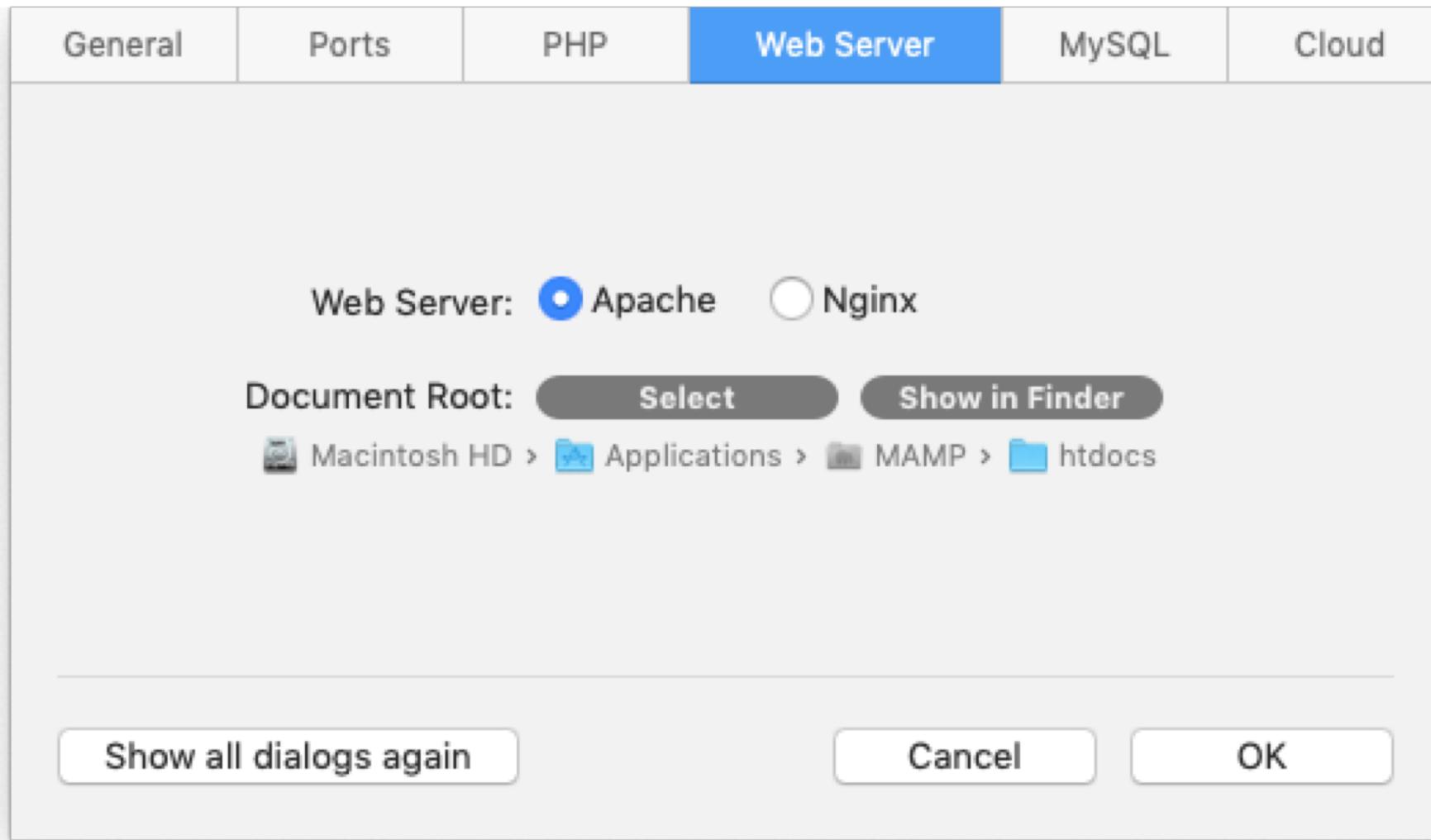
# Starta MAMP (OBS! Ej MAMP PRO)



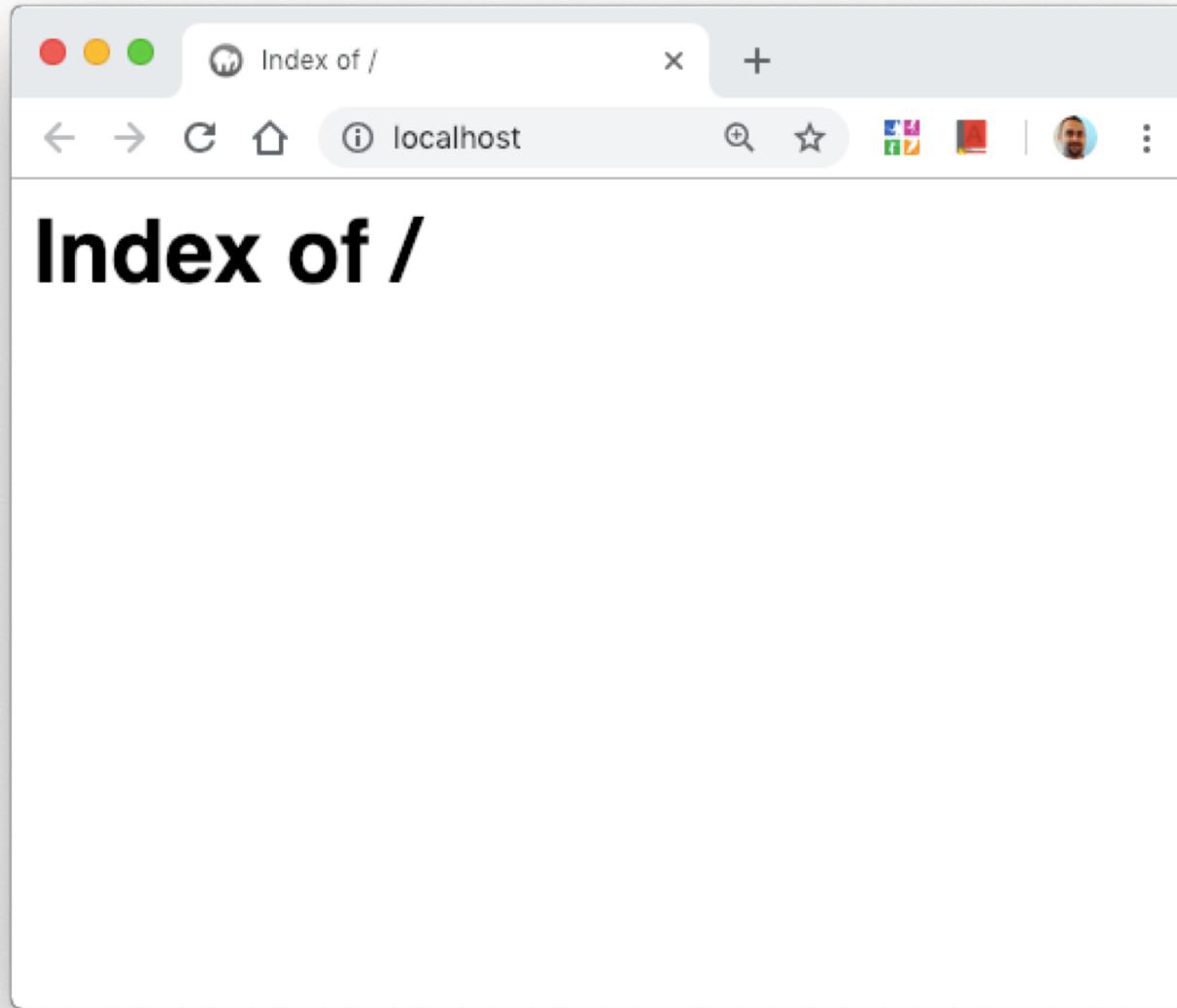
# Konfigurera portar



# Konfigurera Document Root



Testa Apache genom att gå till <http://localhost>



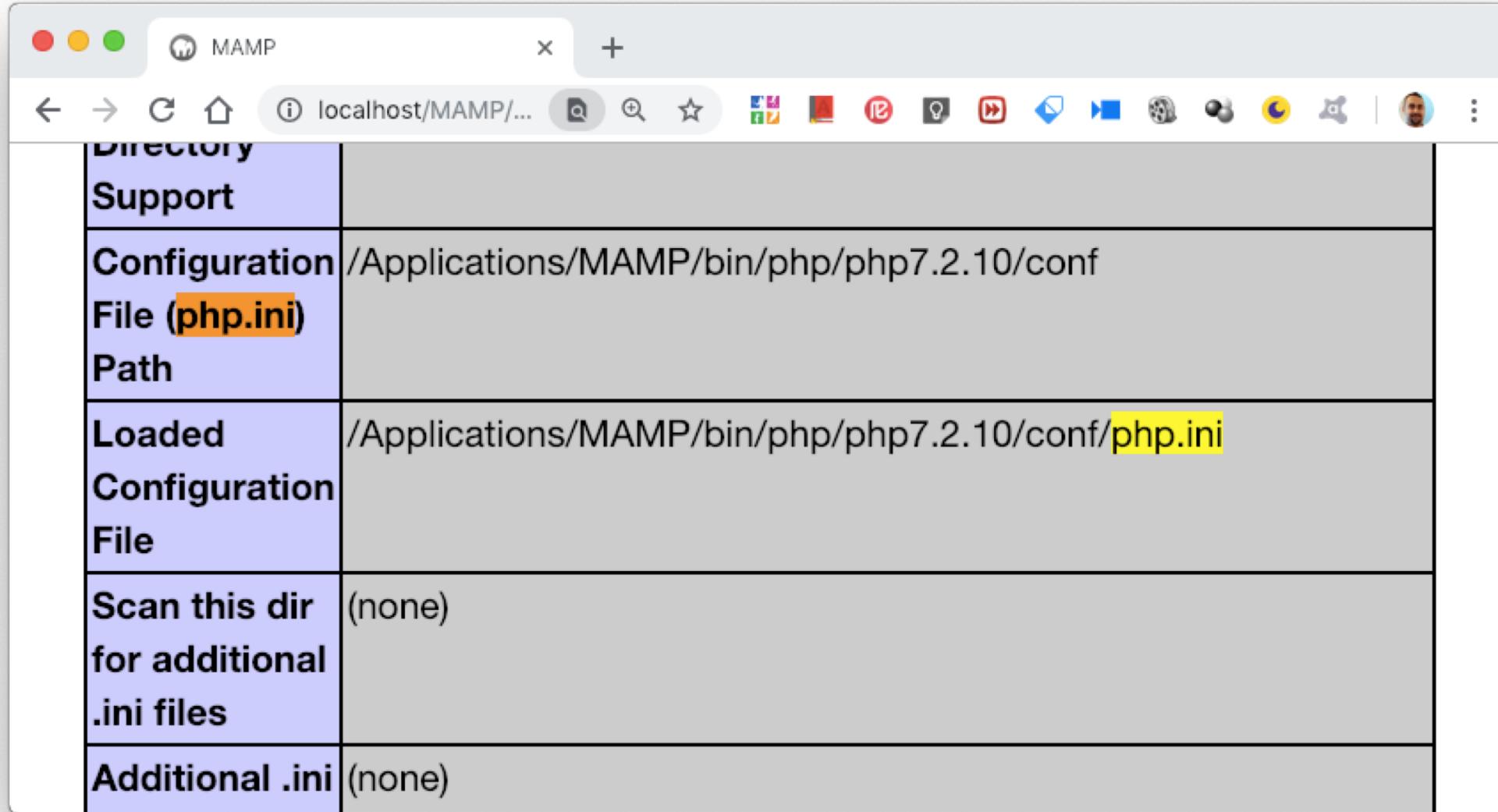
# PHPINFO

A screenshot of a web browser window titled "MAMP". The address bar shows "localhost/MAMP/index...". The page content includes a navigation bar with links: START, MY WEBSITE, TOOLS (which is underlined), MAMP WEBSITE, and BUY MAMP PRO. Below this is a section titled "PHPINFO" which contains links to "PHPMYADMIN" and "PHPLITEADMIN (NEEDS PHP 5.2.4 TO 7.0.X)". A table provides system information:

|                   |  |
|-------------------|--|
| System            | Darwin Mahmuds-MacBook-Air.i<br>4903.201.2~1/RELEASE_X86_64  |
| Build Date        | Oct 9 2018 14:22:23  |
| Configure Command | ./configure' '--with-apxs2=/Appl<br>png-dir=/Applications/MAMP/Lib<br>dir=/Applications/MAMP/Library/<br>prefix=/Applications/MAMP/bin/p<br>path=/Applications/MAMP/bin/p<br>mysqli=mysqlnd' '--enable-mbst<br>imap=shared./Applications/MAM |

The page also lists PHP extensions status: APC (NOT LOADED), EACCELERATOR (NOT LOADED), XCACHE (NOT LOADED), and OPCACHE (NOT LOADED).

# Filen php.ini



The screenshot shows a web browser window titled "MAMP" displaying the configuration page for the PHP module. The URL in the address bar is "localhost/MAMP/...". The page contains a table with the following data:

| Directory Support                              |   |
|--|---|
| <b>Configuration File (php.ini)</b>            | /Applications/MAMP/bin/php/php7.2.10/conf         |
| <b>Path</b>                                    |   |
| <b>Loaded Configuration File</b>               | /Applications/MAMP/bin/php/php7.2.10/conf/php.ini |
| <b>Scan this dir for additional .ini files</b> | (none)  |
| <b>Additional .ini</b>                         | (none)  |

# Sätt `display_errors = on`

⚙️ php.ini ✖️

```
464 ; Possible Values:  
465 ; Off = Do not display any errors  
466 ; stderr = Display errors to STDERR  
467 ; On or stdout = Display errors to  
468 ; Default Value: On  
469 ; Development Value: On  
470 ; Production Value: Off  
471 ; http://php.net/display-errors  
472 display_errors = on
```

# Webbhotell

- Ett webbhotell är en tjänst som möjliggör för den som inte vill upprätthålla en egen offentlig webbserver att publicera sig på World Wide Web.
- Ett webbhotell har en internetansluten dator (webbserver) – i praktiken flera datorer, för att klara behovet av kapacitet och tillgänglighet – där flera användare kan lägga upp sina hemsidor eller webbplatser (under olika domännamn).
- Källa: <https://sv.wikipedia.org/wiki/Webbhotell>

# Domännamn

- Ett domännamn är på Internet ett namn på en administrativ helhet på Internet.
- Organisationers offentliga tjänster (såsom e-post och webbplatser) och enskilda datorer kan ofta nås med hjälp av domännamnet.
- Domännamnen finns till för att människor ska slippa komma ihåg de numeriska IP-adresser som används som "telefonnummer" på internet mellan datorer.
- Källa: <https://sv.wikipedia.org/wiki/Dom%C3%A4nnamn>
- Tips: Lista över toppdomäner  
[https://sv.wikipedia.org/wiki/Toppdom%C3%A4n#Lista\\_%C3%B6ver\\_toppdom%C3%A4ner](https://sv.wikipedia.org/wiki/Toppdom%C3%A4n#Lista_%C3%B6ver_toppdom%C3%A4ner)

# Webbhotell – Några exempel

- <https://misshosting.se>
- <https://loopia.se>
- <https://binero.se>
- <https://manufrog.com>
- <https://oderland.se>

# Namnserver – DNS-server

## DNS (Domain Name System)

- Namnserver eller DNS är ett system som används för att tala om för webbservern vilken "adress" ett domännamn har.
- När du har ett webbkonto hos ett webbhotell så måste webbhotellet koppla rätt Namnservrar (DNS-servrar) till ditt domännamn.
- Hos domänregistrator (företaget som tar hand om domännamnet) måste Du koppla rätt namnservrar till Ditt domännamn.
- Tips: <https://it-ord.idg.se/ord/namnserver>

# Ändra Namnservrar

Exempel från  
Loopia.se

## Fler inställningar för alhakim.se:

Namnservrar

DNSSEC

Återställ webbackup

Återställ DNS-backup

Lösenordsskydd

Räknare

Formmail

DNS-editor

## Namnservrar

**Observera:** Informationen om DNS-servrar i fälten nedan kommer från root-servrar på Internet. Om du gör ett byte av DNS-servrar så kommer det ta några timmar innan den ändring du gjort slår igenom. Till dess visas den gamla informationen i fälten.

Om du vill byta till Loopias namnservrar klicka först [här](#) och sedan på spara.

DNS-server 1 \*

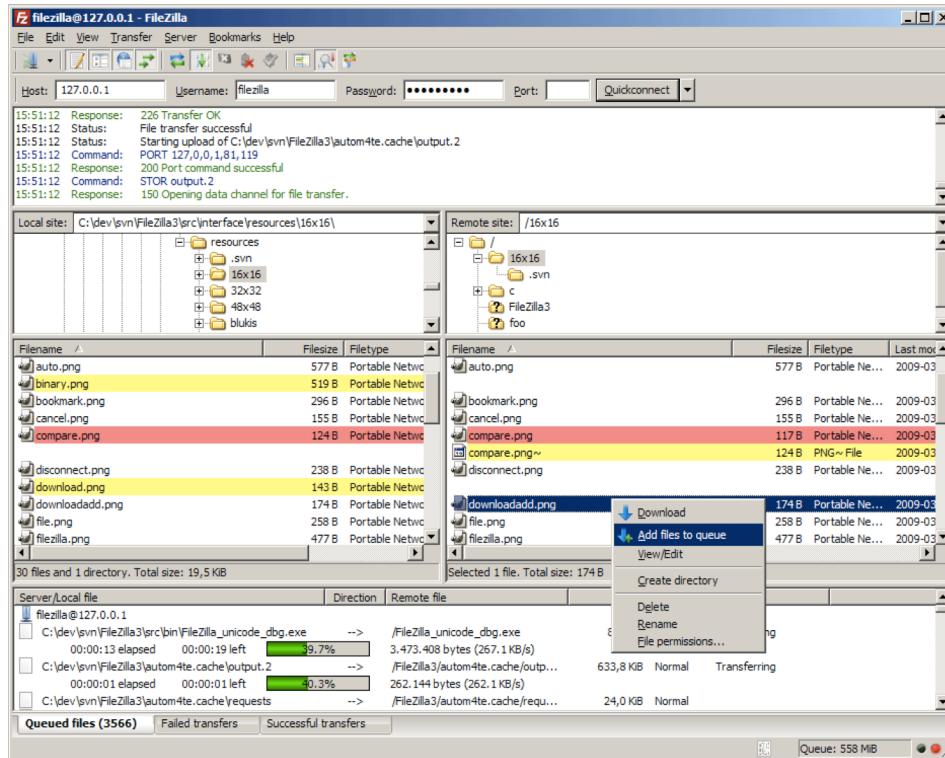
ns46.manufrog.com

DNS-server 2 \*

ns45.manufrog.com

# FTP – File Transfer Protocol

- FTP är ett av de tidigaste populära filöverföringsprotokollen för Internet.
- FileZilla är en välkänd FTP-klient <https://filezilla-project.org>



# Introduktion till PHP

- PHP står för **PHP: Hypertext Preprocessor**.
- PHP är ett skriptspråk som används för att bygga dynamiska och databasdrivna webbapplikationer.
- PHP utvecklades av Rasmus Lerdorf år 1995.
- PHP-dokument bearbetas av en webbserver.
- PHP-kod interpreteras av webbservern och resultatet skickas som ren HTML-kod vidare till webbläsaren.
- Officiell webbplats: <http://php.net>



<http://lerdorf.com/bio>

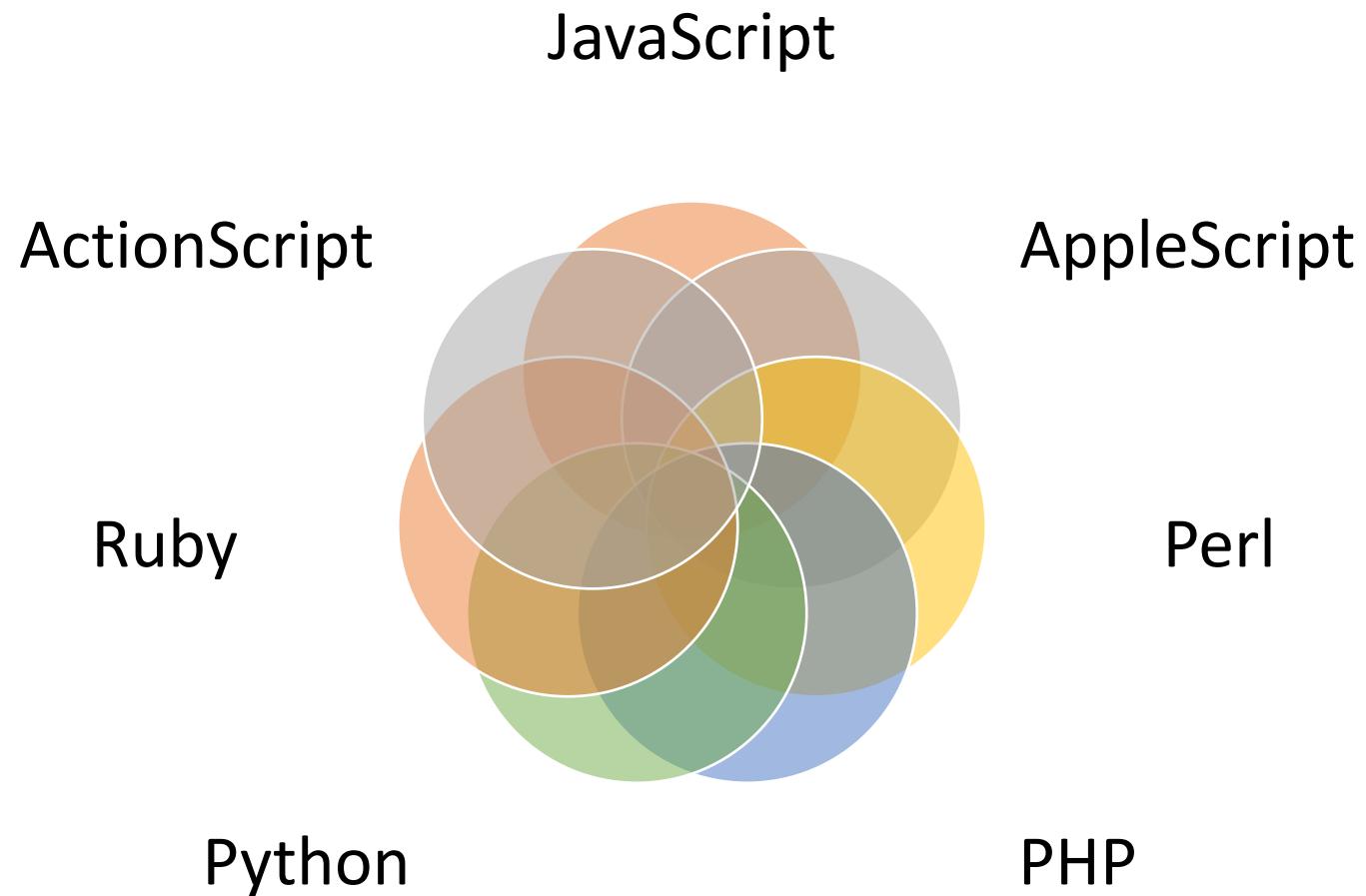
# Olika slags programmering

- Programmering kan ske på olika abstraktionsnivåer.
- Närmast hårdvaran, förutom maskinkod, ligger assemblerprogrammering där man använder ett assemblerspråk.
- På nästa nivå kommer allmänna programspråk (engelska: general purpose programming languages)  
Till exempel Java, C++, C#, PHP och JavaScript.
- Källkoden i dessa språk kompileras till maskinkod av en kompilator eller översätts (interpretaras) av en tolk.

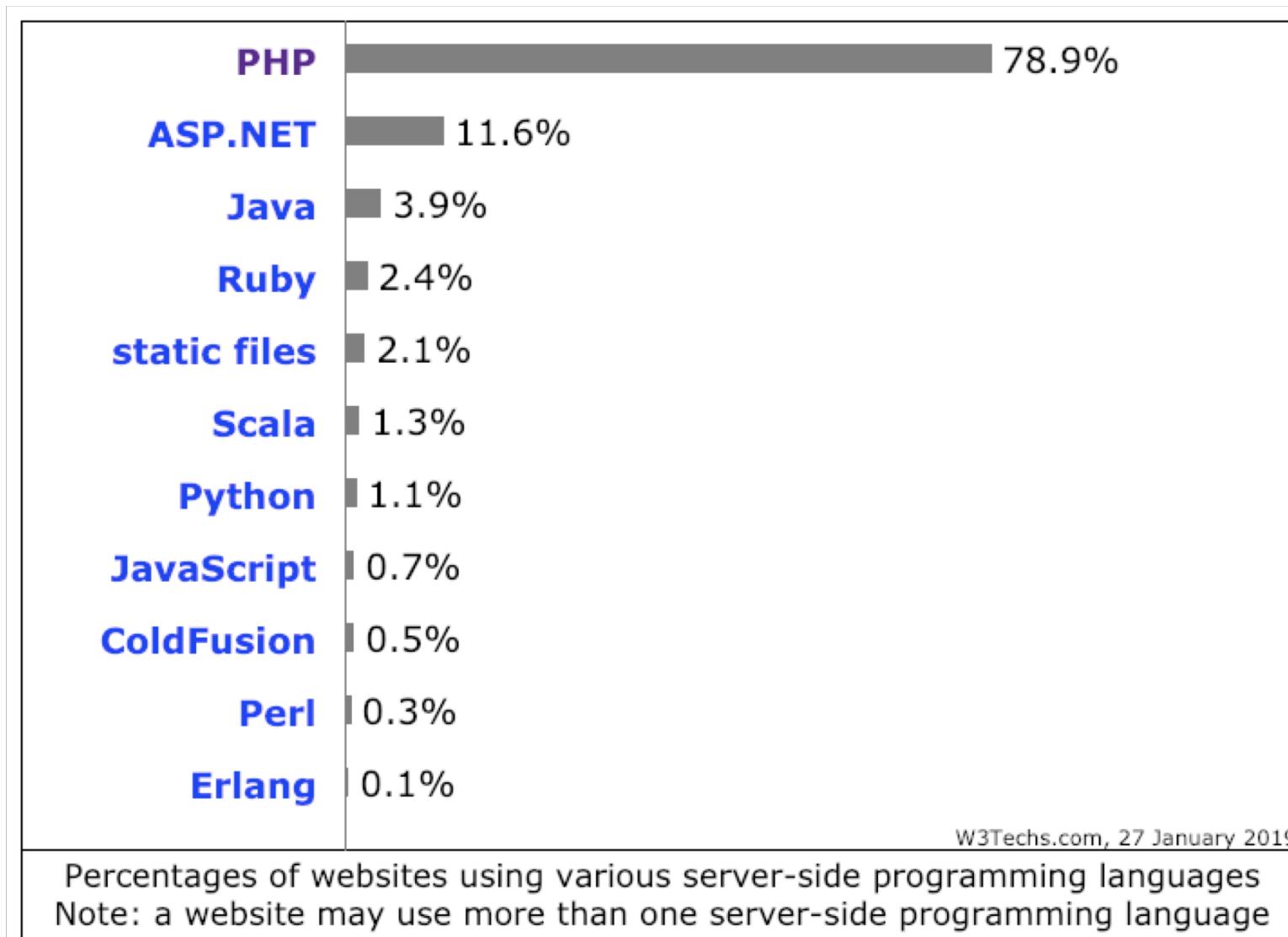
# Vad är skriptspråk?

- Skriptspråk är inom datavetenskap en benämning på "små" högnivåspråk inriktade på specialiserade uppgifter inom redan befintliga miljöer, i motsats till systemspråk, som används för programmering av tillämpningsprogram.
- Det finns ingen exakt avgränsning mellan systemspråk och skriptspråk, men skriptspråk är ofta interpreterande programspråk som använder dynamisk typning.
- Källa  
<https://sv.wikipedia.org/wiki/Skriptspr%C3%A5k>

# Exempel på skriptspråk



# Usage of server-side programming languages for websites

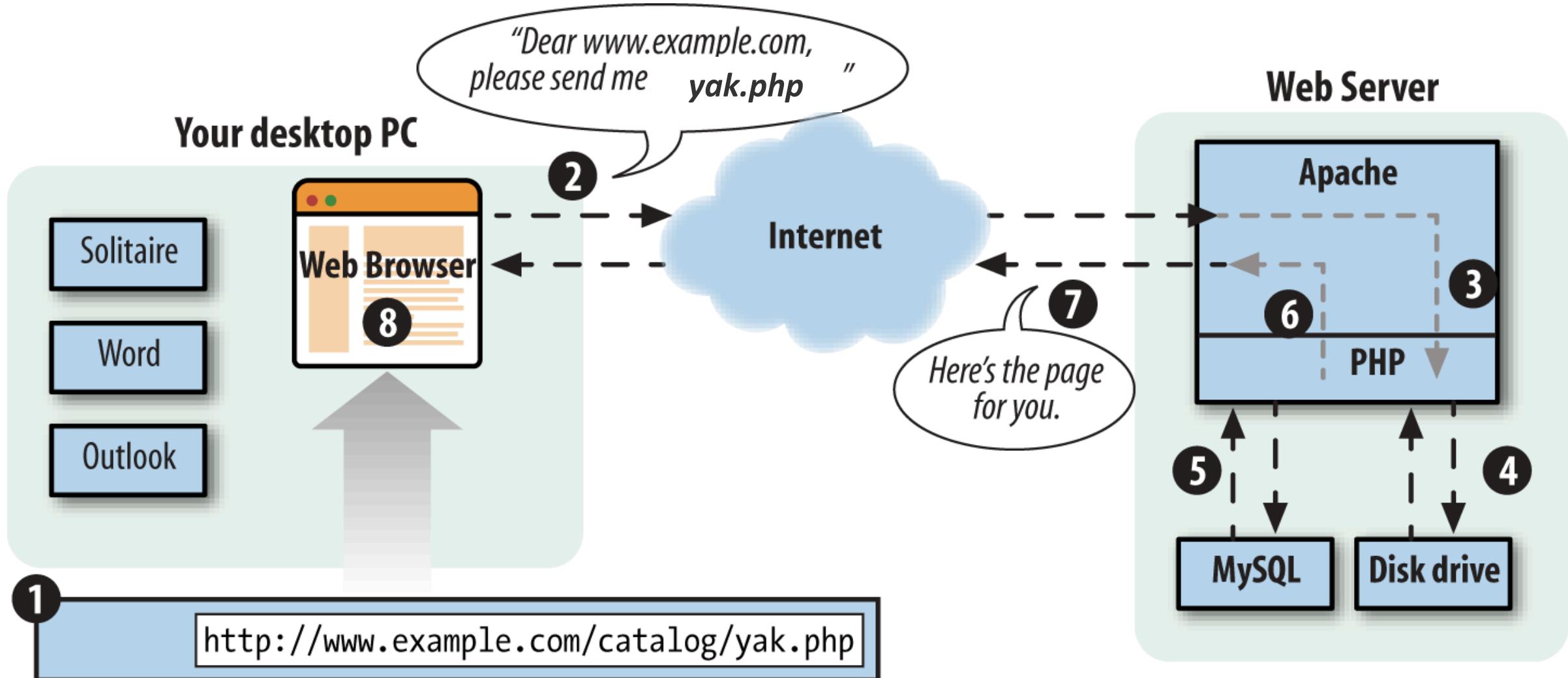


# Varför PHP?

- Med PHP kan du skapa databasdrivna webbapplikationer och tjänster t.ex.
- Blogg:** Världens största bloggsystem, WordPress är uppbyggt av PHP.
  - Webbutik (Webbshop):** Många e-handelslösningen såsom Magento och opencart är skapade med PHP.
  - Sociala nätverk:** Facebook är uppbyggt av bl.a. PHP.



# Händelseförfloppet vid anrop av en PHP-fil



# Syntax

---

Syntax är språkets regler och grammatik

---

Syntax handlar om att skriva korrekt källkod

---

En duktig programmera måste känna till språkets syntax!

# PHP – Grundläggande syntax

1. PHP-filer måste sparas med filändelsen **.php** t.ex. exempel-1.php
2. PHP-kod placeras mellan s.k. PHP tags **<?php** och **?>**

```
<?php  
    // Skriv PHP-kod här  
?>
```

3. PHP-filer måste sparas på serverns "Document Root"  
Oftast kallas dokument-roten på riktiga online servrar **public\_html** och **htdocs** lokalt i t.ex MAMP eller XAMPP.

# Hello World!

```
<?php  
    echo "Hello World";  
    // echo är ett PHP-kommando som skickar text till webbläsaren.  
?>
```



# Kommentarer

- Att kommentera källkod är en konst.
- Skriv i kommentaren VAD som görs och inte HUR det görs.
- Kommentera i en sammanhängande längre kommentar före ett avancerat block vad som görs.
- Undvik ”Papegoja -kommentarer”!

# Kommentarer i PHP

```
// Detta är en kommentar på en rad  
# Detta är också en kommentar på en rad
```

```
// Nedanstående sats är bortkommenterad  
// echo "Hello World";
```

```
echo "Hello World"; # skriver ut "Hello World"  
// Ovanstående kommentar är en "Papegoja-kommentarer"
```

```
/* Detta är en kommentar som sträcker sig  
   över flera rader.  
*/
```

En PHP-fil kan bestå av både HTML och PHP-kod

```
<!DOCTYPE html>
<html lang="sv">
<head>
    <meta charset="UTF-8">
    <title>Exempel</title>
</head>
<body>
    <?php
        echo "Hello World";
    ?>
</body>
</html>
```

# Övning: Min första PHP-sida

1. Skapa en php-fil och spara filen som exempel-01.php  
OBS! Spara filen i webbserverns dokument-rot (t.ex. htdocs)

2. Skriv nedanstående kod

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

3. Starta webbläsaren och gå till <http://localhost/>

OBS! MAMP använder port 8888, så du behöver lägga till port så här  
<http://localhost:8888/>

Tips: Ändra port till 80 i MAMPs inställningar!

4. Klicka på filen för att köra skriptet!

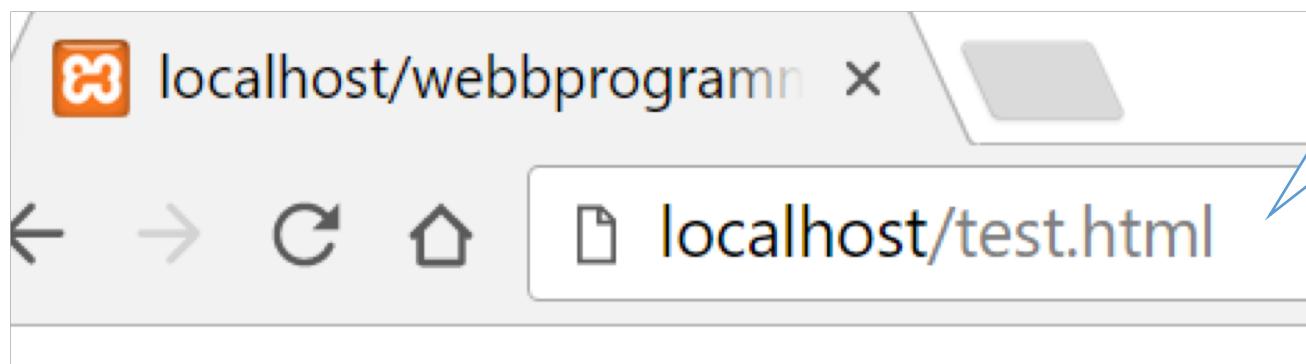
# Visa felmeddelanden

```
<?php  
// Lägg till dessa rader för att visa felmeddelanden  
ini_set('display_errors', '1');  
error_reporting(E_ALL);  
?>
```

# Vanliga fel – Nr 1

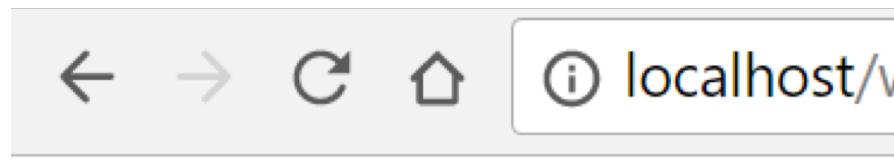
```
<?php  
    echo "Hello World";  
?>
```

Du har sparat  
filen som .html  
och inte .php



## Vanliga fel – Nr 2

```
echo "Hello World";
```



```
echo "Hello World";
```

Du har glömt  
<?php ?>

## Vanliga fel – Nr 3

```
<?php  
echo "Hello World";
```

```
<p>Text efter PHP-koden.</p>
```

Du måste stänga med ?>  
Om du har text eller HTML-  
kod efter PHP-koden.  
I annat fall behöver du inte  
stänga med ?>



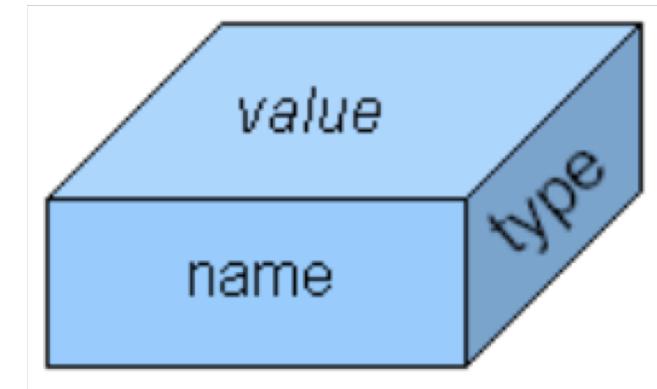
# Echo, print och <?= ?>

```
<?php
    echo "<h1>Hello World</h1>";
    echo "<p>Text som sträcker sig<br>
        över flera rader.<p>";
    print "<p>Du kan använda print istället för echo.</p>";
?>
<hr>
<?= "Ett förkortat syntax för att skicka text till webbläsaren
istället för echo/print. Detta kallas Short Echo Tag" ?>
```

Läs mer: <http://php.net/manual/en/function.echo.php>

# Variabler och värde

- En variabel är en platshållare för ett värde som kan ändras under programmets gång.
- I datorprogram använder man variabler för att lagra data.
- Data kan vara av olika slag (typer)  
t.ex. text eller tal.
- I PHP kan en variabel innehålla data av en valfri typ.
- Innan man använder en variabel i ett program måste man deklarera den.



# Deklarera en variabel



PHP-variabler  
inleds alltid med  
ett dollartecken \$

**\$namn;**

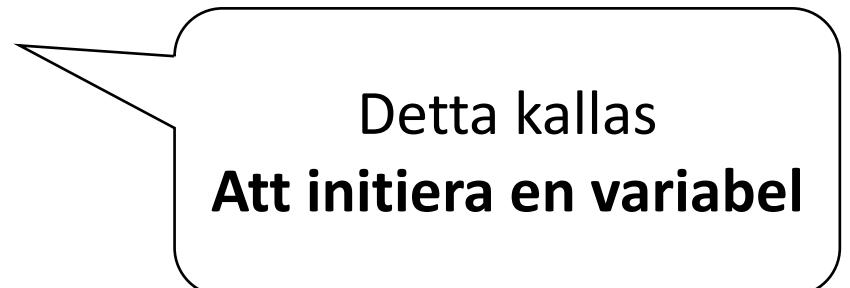
# Tilldelning

- Vill du ge en variabel ett värde använder du likhetstecknet (=) som kallas **tilldelningsoperator (assignment operator)**.
- Då du ger en variabel ett värde kallas det att du **tilldelar (assign)** variabeln ett värde t.ex.

```
$namn = "Mahmud";
```

- Du kan deklarera och tilldela en variabel ett värde på en och samma gång t.ex.

```
$age = 45;
```



Detta kallas  
**Att initiera en variabel**

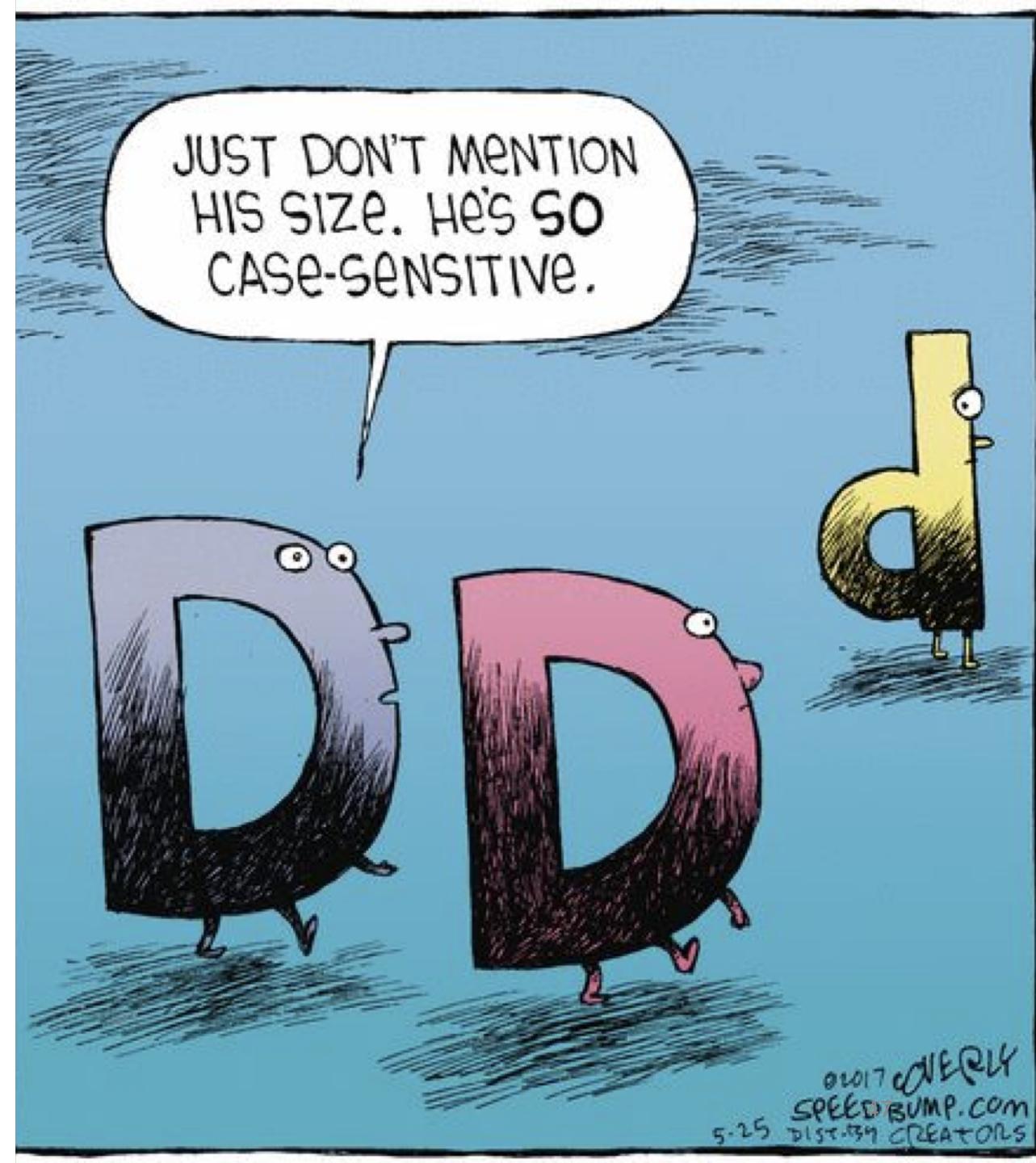
# Variabler – Exempel

- PHP-variabler inleds alltid med ett dollartecken \$

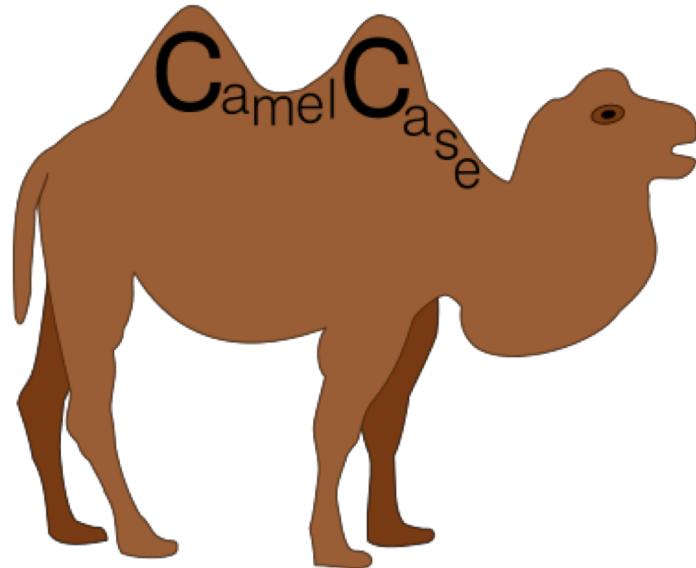
```
<?php  
    $webmaster = "Mahmud Al Hakim";  
    echo $webmaster;  
?>
```

# Variabler i PHP är Case Sensitive

```
// OBS! tre olika variabler  
$firstname;  
$firstName;  
$FIRSTNAME;
```



# camelCase och snake\_case



- Du kan använda camelCase för att skriva variabelnamn.  
`$firstname; // Ok men...`  
`$firstName; // mycket bättre variabelnamn`
- Inbyggda funktioner i PHP använder snake\_case,  
t.ex. `ini_set` och `error_reporting`

# OBS! Variabler får inte börja med siffror

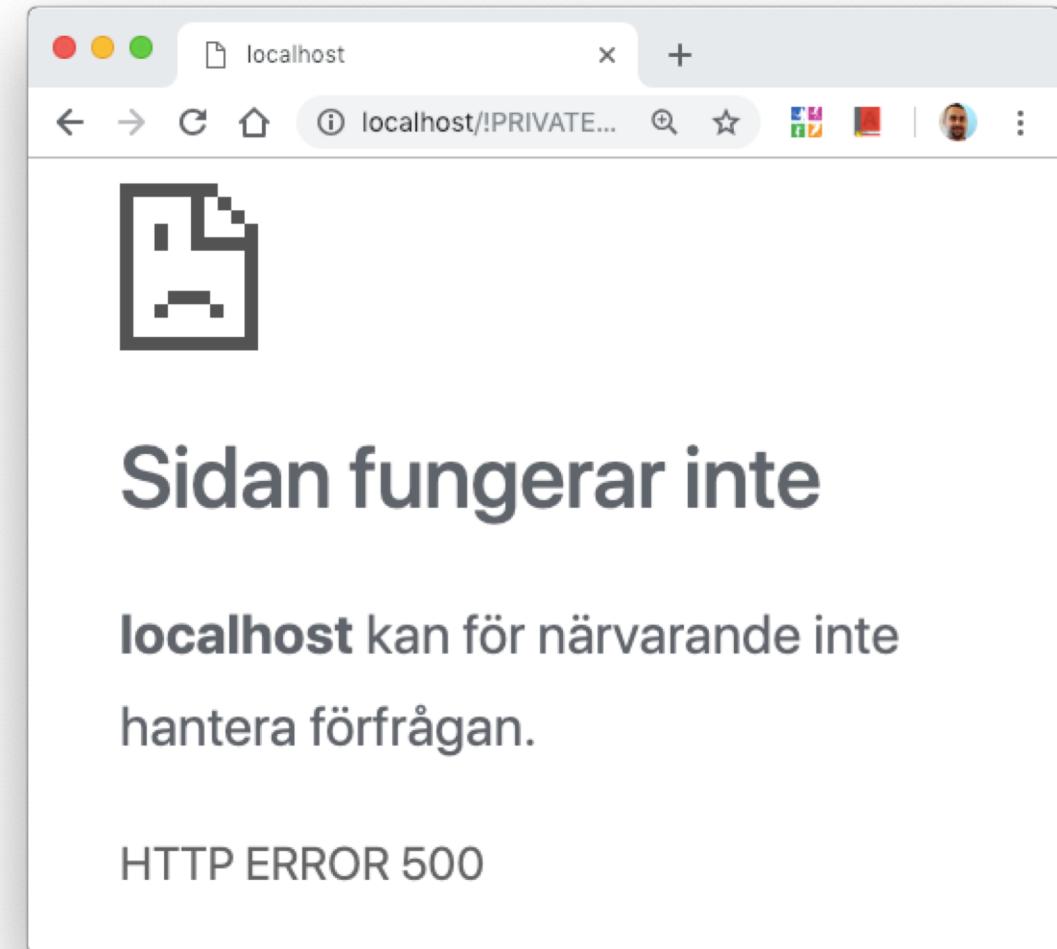
```
<?php
```

```
// OBS! Syntax Error  
// HTTP ERROR 500
```

```
$123;
```

```
// Men du kan börja med understreck  
$_123 = "DEMO";  
echo $_123;
```

```
?>
```



# Variabler – Datatyper

- Funktionen `var_dump()` returnerar datotyp och värde.

```
$text="Text";
var_dump($text);      // string(4) "Text"
```

```
$heltal=123;
var_dump($heltal);    // int(123)
```

```
$flyttal=123.99;
var_dump($flyttal);   // float(123.99)
```

```
$boolesk=true;
var_dump($boolesk);   // bool(true)
```

```
$ingenting;
var_dump($ingenting); // NULL
```

# Variabler – Att testa Datatyper

```
$text="123";
echo is_string($text); // 1 (sant)
echo "<br>";
```

```
$heltal=123;
echo is_int($heltal); // 1 (sant)
echo "<br>";
```

```
echo is_int($text); // Tomt resultat (falskt)
echo "<br>";
```

```
echo is_numeric($text); // 1 (sant)
// is_numeric används för att undersöka
// om en variabel är ett nummer
// eller en numerisk sträng
```

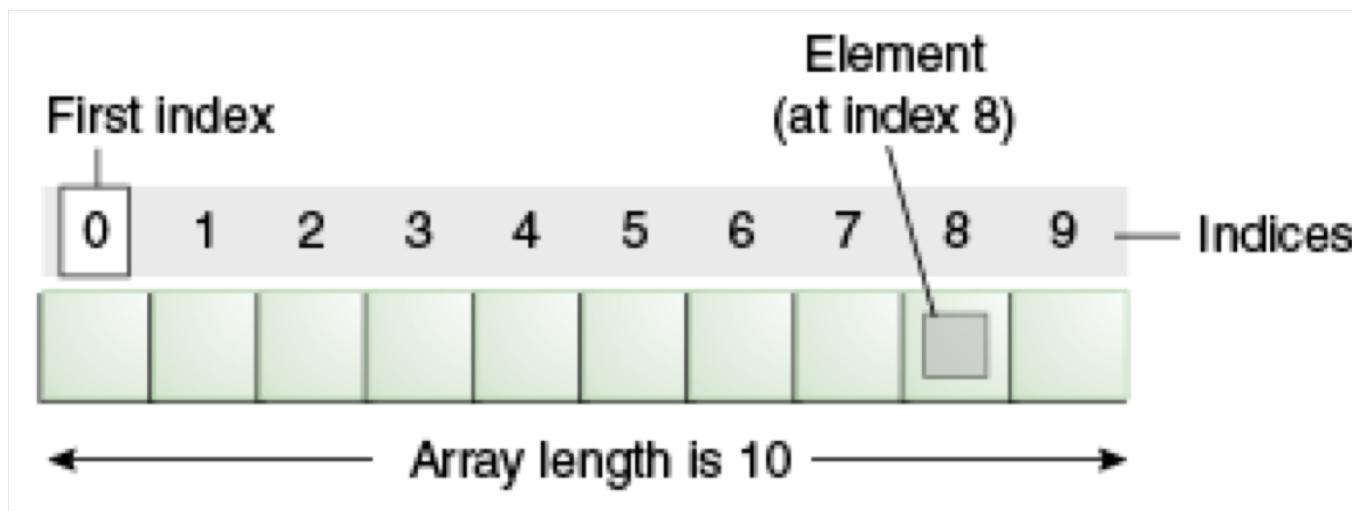
# Strängar

- I PHP anges strängar med enkla eller dubbla citationstecken.
- Skillnaden är att variabler inom dubbla citationstecken byts ut mot innehållet (värdet).

```
<?php  
    $fname = "Mahmud";  
    $lname = "Al Hakim <br>";  
    echo "$fname $lname"; // => Mahmud Al Hakim  
    echo '$fname $lname'; // => $fname $lname  
?>
```

# Vad är en array (vektor)?

- En array eller en vektor är en datastruktur som består av en samling av element som identifieras med ett eller flera heltaliga index.
- Till skillnad från en vanlig variabel som bara innehåller ett värde kan ett fält alltså innehålla ett godtyckligt antal värden.



# Skapa en array

## 1. Array Literal

```
$cars = ["Saab", "Volvo", "BMW"];
```

## 2. Array Constructor

```
$cars = Array("Saab", "Volvo", "BMW");
```

# Indexerad/Numerisk Array

- Adressen till varje element räknas ut via dess index.

```
echo "Jag tycker om $cars[0], $cars[1] och $cars[2].";
```

# Skriv ut en array med print\_r()

```
// Funktionen print_r används oftast vid felsökning  
print_r($cars);
```

```
// Snyggare utskrift med pre-taggen  
echo"<pre>";  
print_r($cars);  
echo"</pre>";
```

# Ändra ett element

```
$cars = ["Saab", "Volvo", "BMW"];  
$cars[0] = "Opel";
```

# Antal element

```
$cars = ["Saab", "Volvo", "BMW"];  
echo count($cars);
```

# Sortera en array

```
$cars = ["Saab", "Volvo", "BMW"];  
echo "<h2>Före sortering</h2>";  
print_r($cars);  
  
echo "<h2>Efter sortering</h2>";  
sort($cars);  
print_r($cars);
```

# Lägg till nya element

```
// En tom array
```

```
$products = [];
```

```
// Lägg till ett element
```

```
$products[] = "PC";
```

```
// Lägg till ett element
```

```
$products[] = "Mac";
```

# Associativ Array

- En associativ array använder strängar som index istället för heltal.

```
$flowers = array(  
    'ros'      => 50,  
    'liljor'   => 60  
) ;
```

```
print_r($flowers);
```

## Associativ Array – Exempel 2

```
$age = [  
    'mahmud'=> 45,  
    'kalle' => 10  
];
```

```
echo $age['mahmud'];
```

# Miljövariabler – \$\_SERVER

```
// $_SERVER är en global associativ array  
// som innehåller allmän info om servern och dess miljö  
echo "<pre>";  
print_r($_SERVER);  
echo "</pre>";  
  
echo $_SERVER['PHP_SELF']; // Skriver ut filnamnet
```

# Multidimensionell Array

```
$cars = array
(
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);

echo $cars [0] [0];
```

# Hämta data via webbadressen (URL)

- En av de främsta fördelarna med PHP är möjligheten att hantera information som skickas av en användare via webbläsarens adressfält (URLen).

```
<?php  
    echo $_GET['name'];  
?>
```

- En nyckel (namn) och ett värde skickas via URLen t.ex.

exempel.php?namn=Mahmud



# Skicka mer data via URLen med &

```
// Skriv detta i URLen  
// exempel.php?user=Mahmud&order=1234  
  
$user = $_GET["user"];  
$order = $_GET["order"];  
echo "Hej $user.<br>Ditt ordernummer är $order"
```

# Mer om \$\_GET

- \$\_GET är en global associativ array som erhåller info som skickats med metoden GET via en webbadress (URL) eller ett formulär.
- Du kan skriva ut \$\_GET med funktionen print\_r

```
echo "<pre>";  
print_r ($_GET);  
echo "</pre>";
```

# Övningar

- Läs sid. 1-90 i boken ”Webbutveckling med PHP och MySQL”.
  1. Arbeta med övningarna på sid. 23.
  2. Arbeta med övningarna på sid. 88-90.

# Tips: PHP Front To Back av Brad Traversy

<https://youtu.be/oJbfyzaA2QA>



The image shows a video player interface. On the left is a thumbnail for the first video of a series titled "PHP: Front To Back [Part 1] - Series Introduction". The thumbnail features the "php" logo inside an oval and the title text. On the right is a list of four video thumbnails for the series, each with a number (1, 2, 3, 4), the title, and a duration. The titles are: "PHP: Front To Back [Part 1] - Series Introduction", "PHP: Front To Back [Part 2] - PHP7 & Xammp Installation", "PHP: Front To Back [Part 3] - Variables, Constants & Data Types", and "PHP: Front To Back [Part 4] - Arrays". The durations are 9:13, 8:03, 15:24, and 13:08 respectively. The channel name "Traversy Media" is visible next to the titles.

| Video Number | Title   | Duration | Channel        |
|--------------|---|----------|----------------|
| 1            | PHP: Front To Back [Part 1] - Series Introduction               | 9:13     | Traversy Media |
| 2            | PHP: Front To Back [Part 2] - PHP7 & Xammp Installation         | 8:03     | Traversy Media |
| 3            | PHP: Front To Back [Part 3] - Variables, Constants & Data Types | 15:24    | Traversy Media |
| 4            | PHP: Front To Back [Part 4] - Arrays                            | 13:08    | Traversy Media |

# Summering av dagens lektion

- Vi har gått igenom
  - **Introduktion till Webbserverprogrammering**
  - MAMP: Portar, Dokument Root
  - Apache Webbserver, localhost
  - PHP.INI: display\_errors
  - Webbhotell och Domännamn
  - Namnserver, DNS
  - FTP
- **Introduktion till PHP**
- Grundläggande syntax
- Variabler
- Indexerade arrayer
- Associativa arrayer
- Multidimensionella arrayer
- Arbeta med `$_GET`
- Reflektioner kring dagens lektion?

**NACKADEMIN**

# Framåtblick inför nästa lektion

**Under nästa lektion kommer vi att jobba med**

- Konstanter
- Operatorer
- Villkorssatser
- Iterationer
- Funktioner