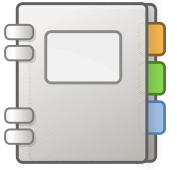
14.03.18 Lecture

Web applications
Normalization



Today

- Normalization
- Web applications
  - Web server
  - Internet structure
  - HTML
  - PHP
  - javascript

Normalization

- We want to avoid redundancy
  - Save space
  - Reduce inconsitency
- Normalization is a set of states and rules to achieve those states

Normalization

- There are three normal forms
  - 0. normal form (kinda)
  - 1. normal form
  - 2. normal form
  - 3. normal form

The book mentions 2 more, but they are not necessary

0. normal form

- Contains non atomic values
  - Columns that contain more than one value or multiple columns that store the same type of values

ID	Name	Phone number
1	Frank	99299399;44033392
2	Ola	4432;991;1142
3	Jens	777;886

ID	Name	Phone 1	Phone 2
1	Frank	99299399	4430202
2	Ola	4432	1131342
3	Jens	777	1102944

1. normal form

- Does not contain non atomic values
  - Achieved by decomposing the table

People

ID	Name
1	Frank
2	Ola
3	Jens

Phone numbers

ID*	**Phone number**
1	99299399
1	44033392
2	777

2. normal form

- On the 1. normal form
- Does not have partial dependencies

Partial dependency

Ordernumber	**Item Number**	Date	Description
1	12	14.02	Hat
2	13	12.02	Cat
3	19	18.02	Fat

Partial dependency:
Part of the primary key determines a column in the table

Item number determines description

2. normal form

- Achieved by decomposing (splitting) tables

Ordernumber	**Item Number***	Date
1	12	14.02
2	13	12.02
3	19	18.02

Item Number	Description
12	Hat
13	Cat
19	Fat

3. normal form

- No transitive dependencies

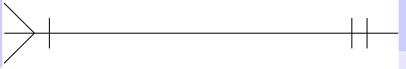
ID	name	Zip code	Zip adress
1	Ola	1442	Oslo
2	Frank	1112	Ås
3	James	7764	Moss

Transitive dependency:
Non-primary key column determines another column

3. normal form

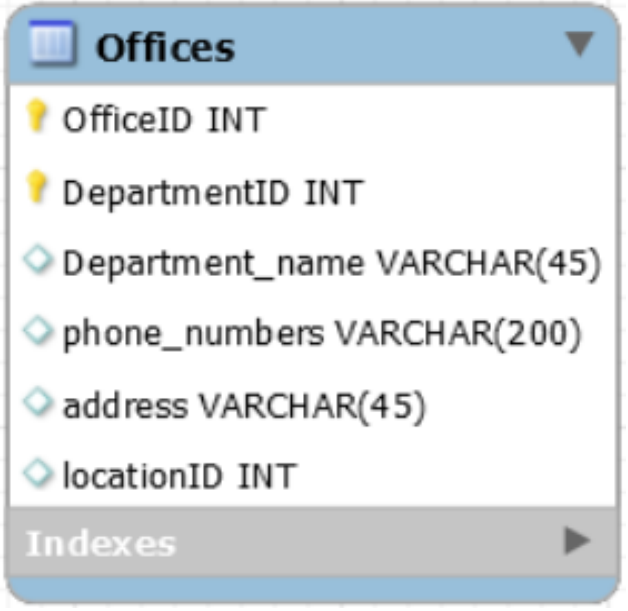
- Achieved by decomposing the table

ID	name	Zip code*
1	Ola	1442
2	Frank	1112
3	James	7764

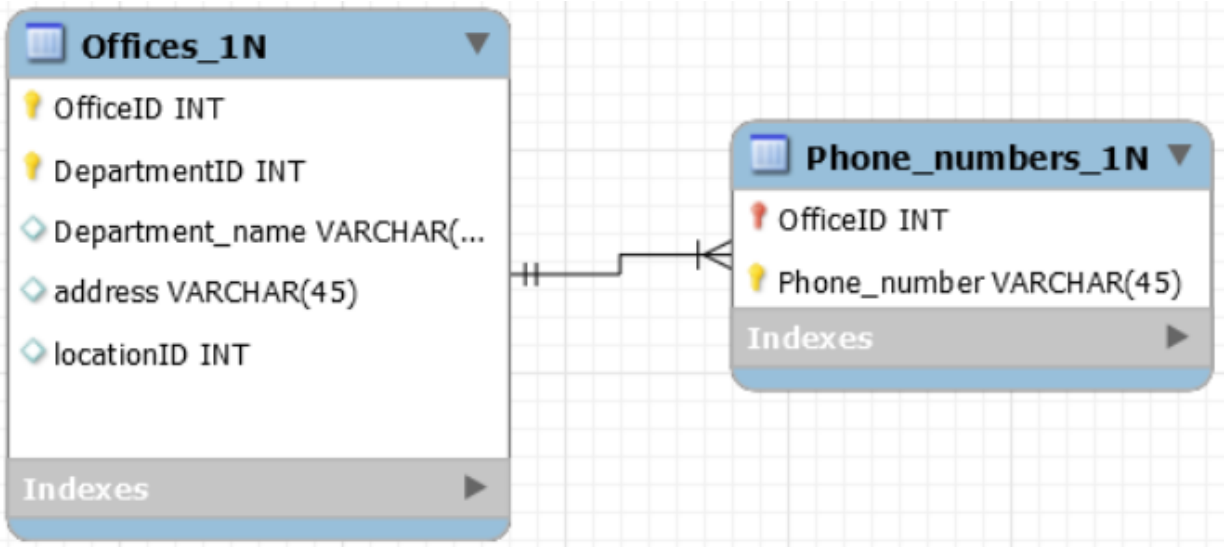


Zip code	Zip adress
1442	Oslo
1112	Ås
7764	Moss

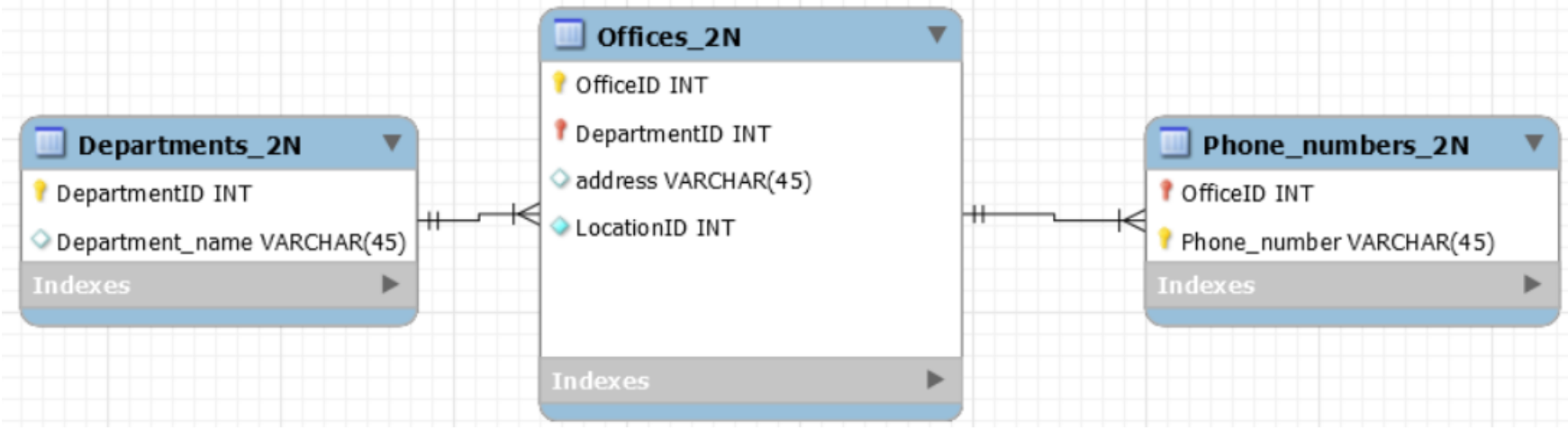
From 0. to 3. Normal form



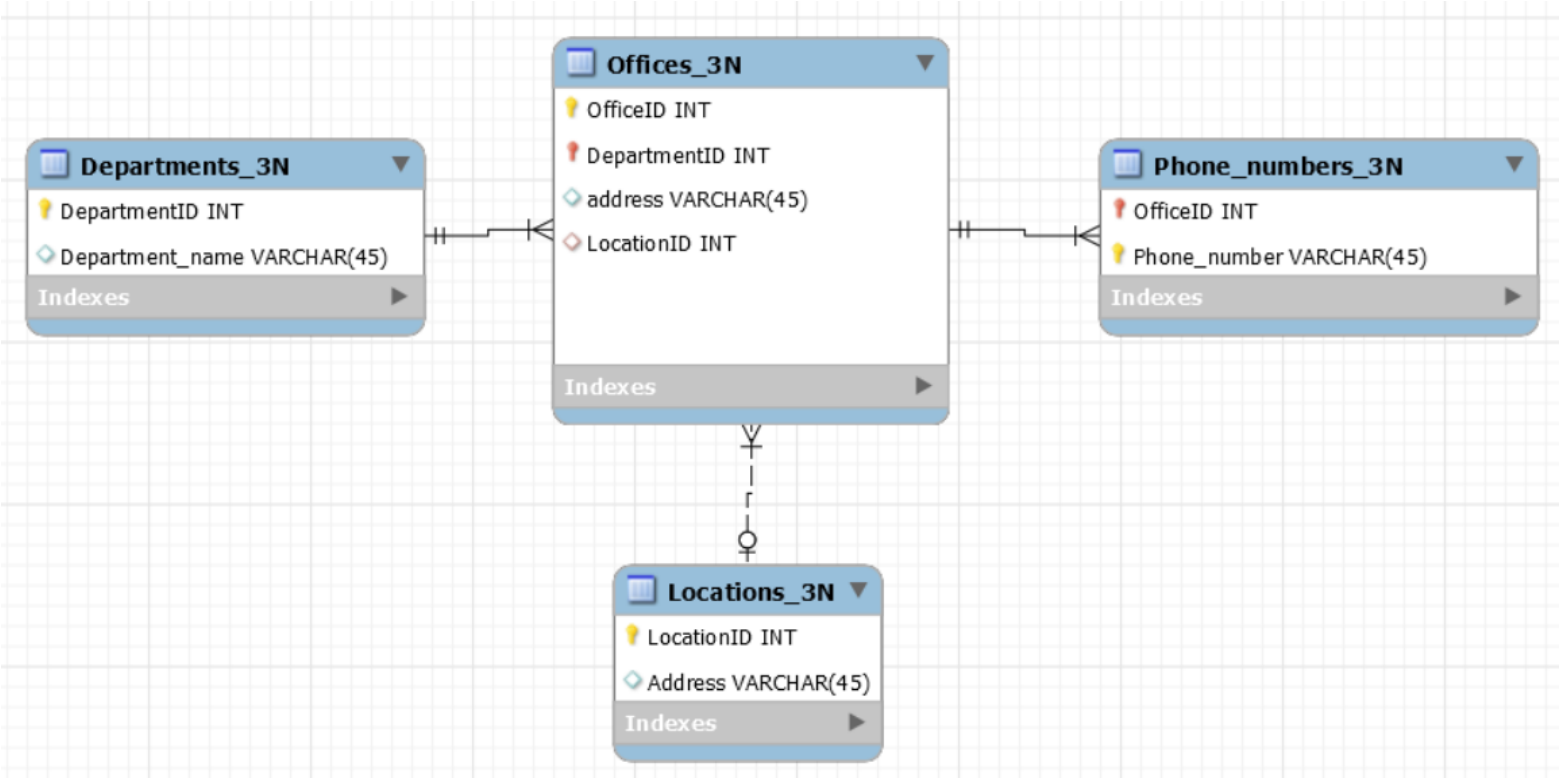
From 0. to 3. Normal form



From 0. to 3. Normal form



From 0. to 3. Normal form



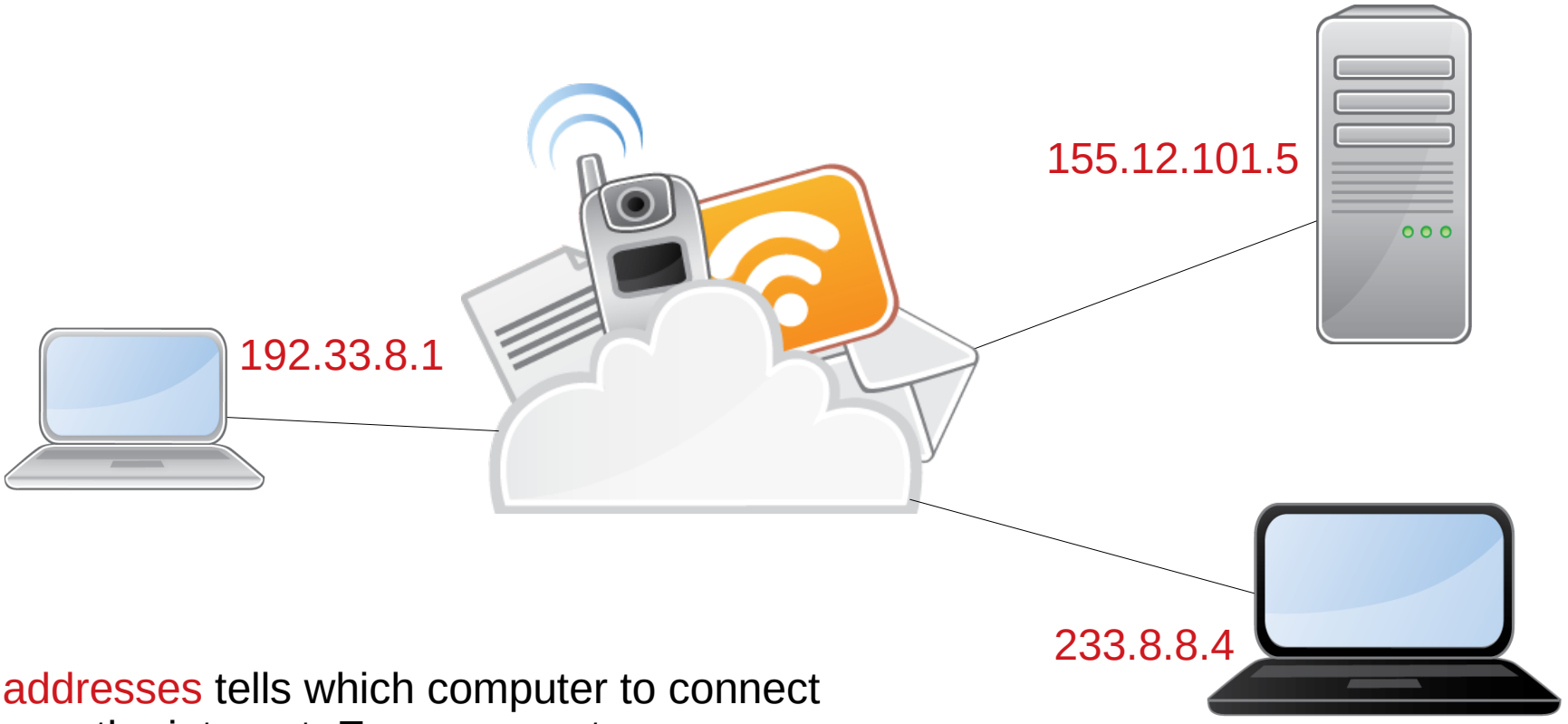
Normalize everything?

- Benefits of normalization
  - Reduces redundancy
    - Less storage space use
    - Prevents update and delete anomolies
- Downsides of normalization
  - Joining tables are computationally expensive

Web Applications

- Internet structure
- Web server
- HTML
- PHP

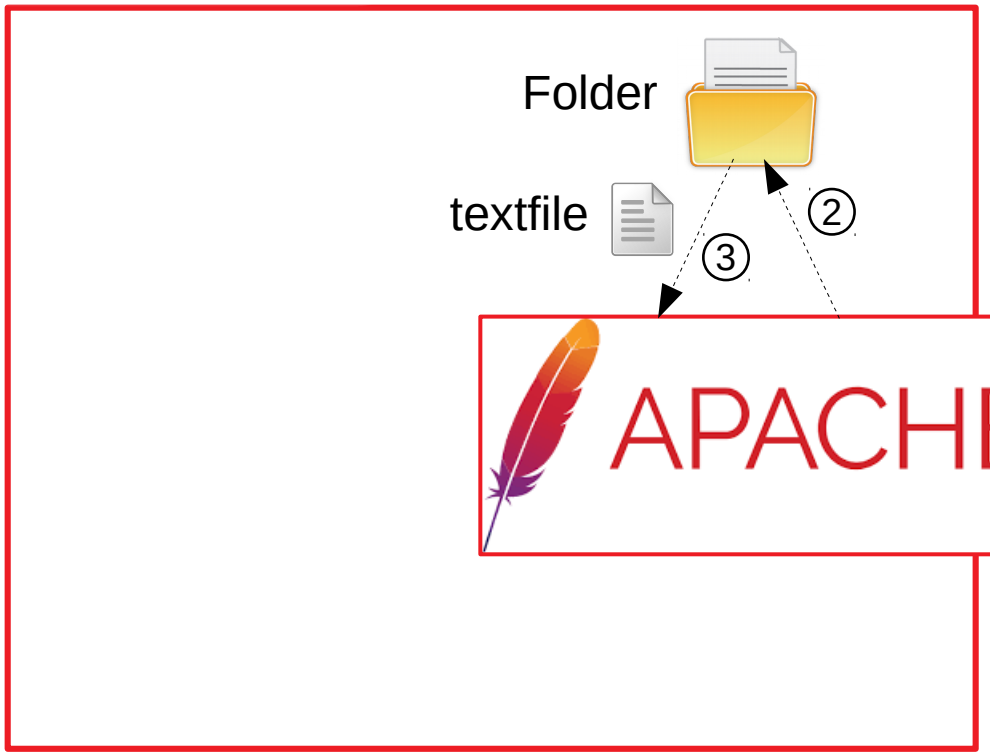
Web structure



IP addresses tells which computer to connect to over the internet. Every computer connected to the internet has one.

Apache webserver

155.12.101.5



Hi 155.12.101.5
I want a webpage

1

192.33.8.1

4

Here it is 192.33.8.1

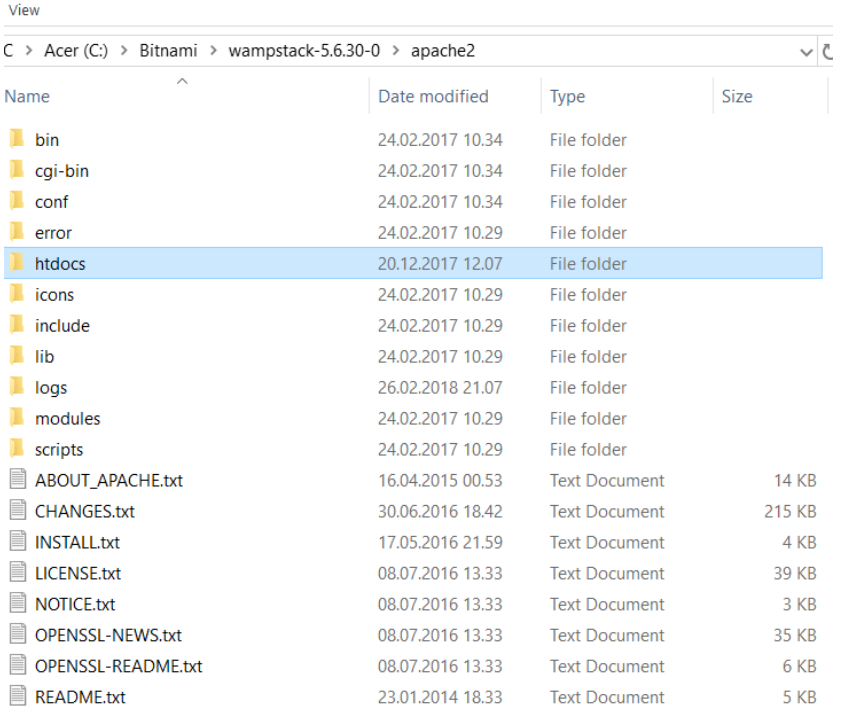
Our webserver setup

- The folder apache webserver goes to
  - Htdocs in the Bitnami\wamp\apache2 folder

htdocs

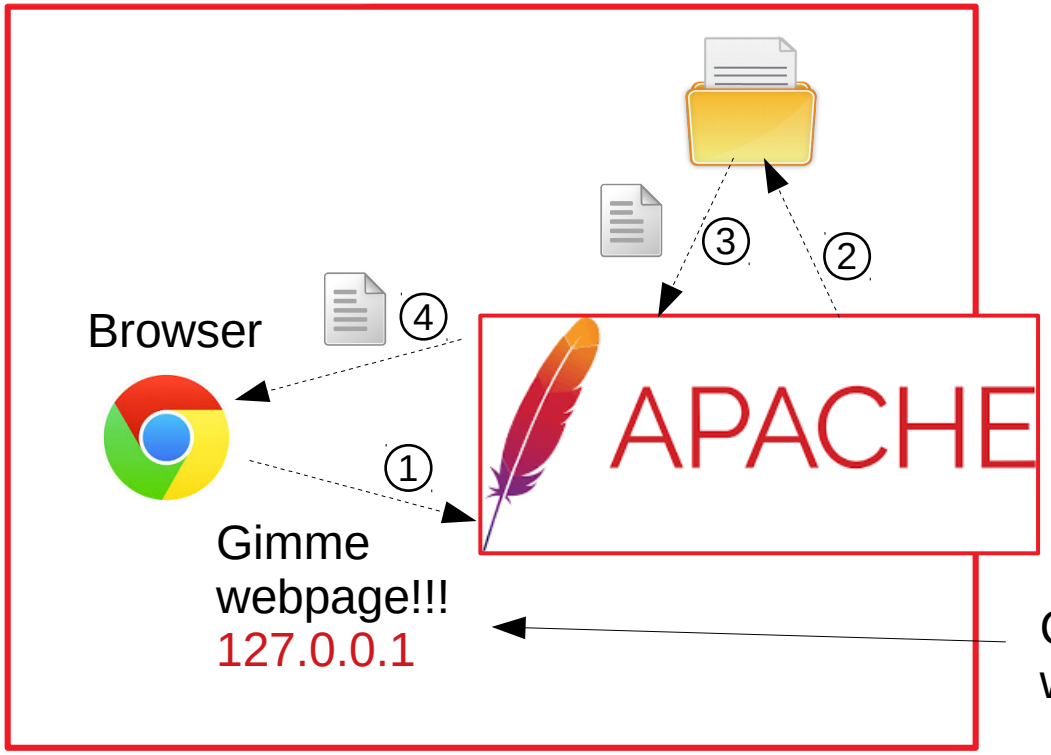
Looks for index.html or index.php

Index.html



Connecting to ourselves

127.0.0.1 is also known as localhost

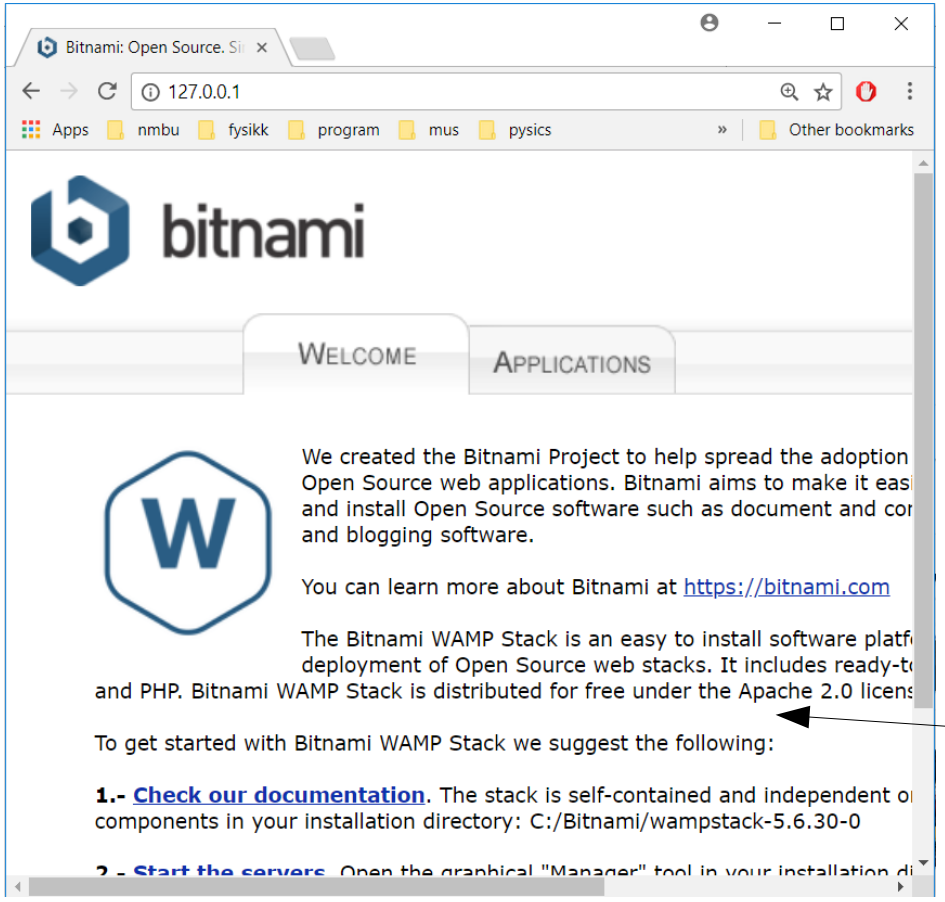


External IP adress changes often

155.12.101.5

Can connect to ourselves
with 127.0.0.1

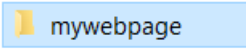
Bitnami default webpage



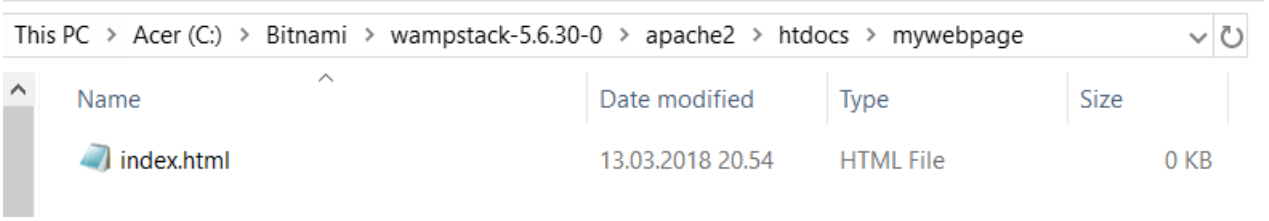
htdocs

cssguide	20.12.2017 13.48	File folder	
homepage	13.03.2018 08.01	File folder	
homepage2	06.03.2017 20.06	File folder	
img	24.02.2017 10.32	File folder	
phpMyEditKK	06.03.2017 22.02	File folder	
phptutorial	20.12.2017 11.51	File folder	
tutrepublic	06.03.2017 21.40	File folder	
503.html	01.04.2016 16.04	HTML File	2 KB
applications.html	24.02.2017 10.42	HTML File	3 KB
bitnami.css	01.04.2016 16.04	Cascading Style Sh...	3 KB
favicon.ico	01.04.2016 16.04	Icon	2 KB
index.html	24.02.2017 10.42	HTML File	3 KB
tutrepublic_2017.zip	06.03.2017 21.40	Compressed (zipp...	8 KB

Lets create our own webpage

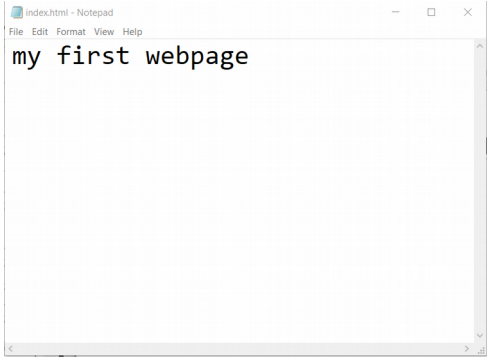
- Create a new folder in htdocs
  - Mywebpage 

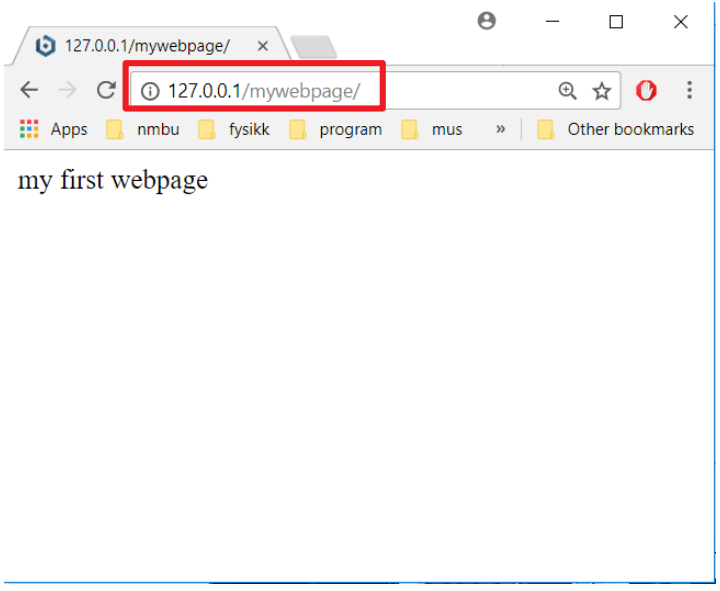
- Create a new textfile in mywebpage
  - Index.html



Lets create our own webpage

- Edit index.html
  - Write "my first webpage"
  - Save
  - Enter 127.0.0.1/mywebpage in browser



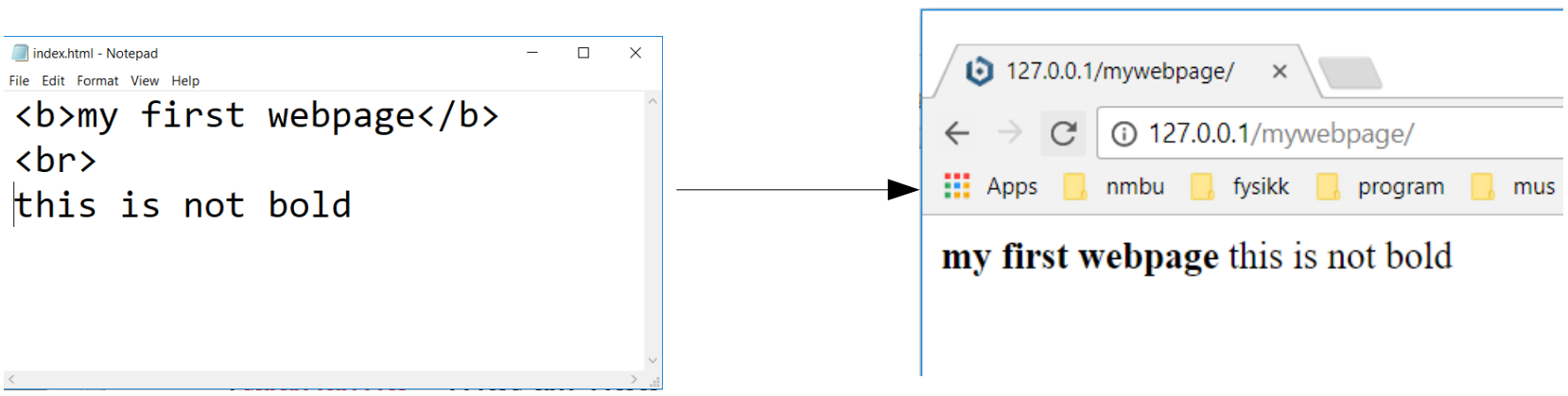


Hypertext markup language (HTML)

- Webpages are written in HTML and has the extentsion .html
  - Contains "tags" which changes the look and behavior of text
- We are going to go through basic html

Basic HTML tags

- A tag starts with <tagname> and ends with </tagname> (not all need an end)
  - this gives bold text
  -
  (this gives a new line)



Structure HTML tags

- A standard html document consists of
  - <html> tag
    - Contains everything (<head>,<body>)
  - <head> tag
    - Usualy contains meta information (title etc)
  - <body> tag
    - Contains the main part of the webpage

Structure HTML tags

index.html - Notepad

File Edit Format View Help

```
<html>
 <head>
  <title>webpage title</title>
 </head>
 <body>
  <b>my first webpage</b>
  <br>
  this is not bold
 </body>
</html>
```

webpage title

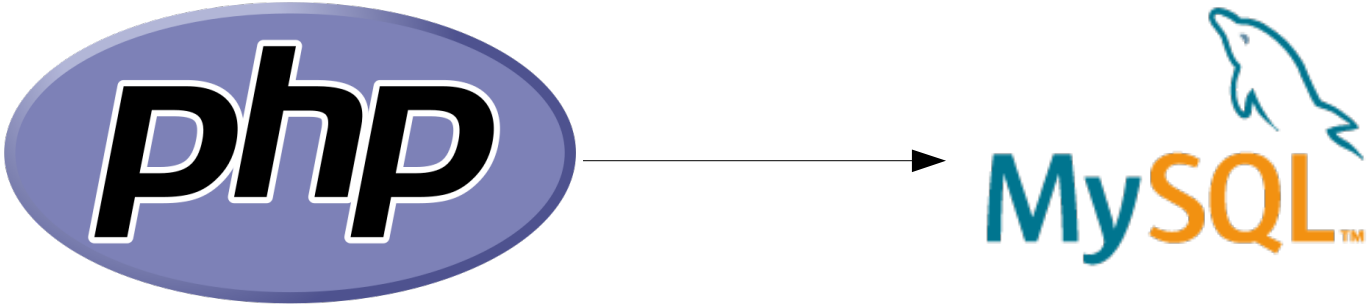
127.0.0.1/mywebpage/

Apps nmbu fysikk program mus » Other bookmarks

my first webpage
this is not bold

Problem with HTML

- Cannot connect to databases
- PHP can



PHP

- Hypertext Preprocessor
  - Why is the acronym PHP?
- Programming language
  - Similar to python,matlab,R etc...
- PHP code is written inside HTML files (renamed to .php files)

PHP

.php file

HTML code

PHP code

HTML code continued

PHP scripts are started with
<?php

Ended with
?>

index.php - Notepad

File Edit Format View Help

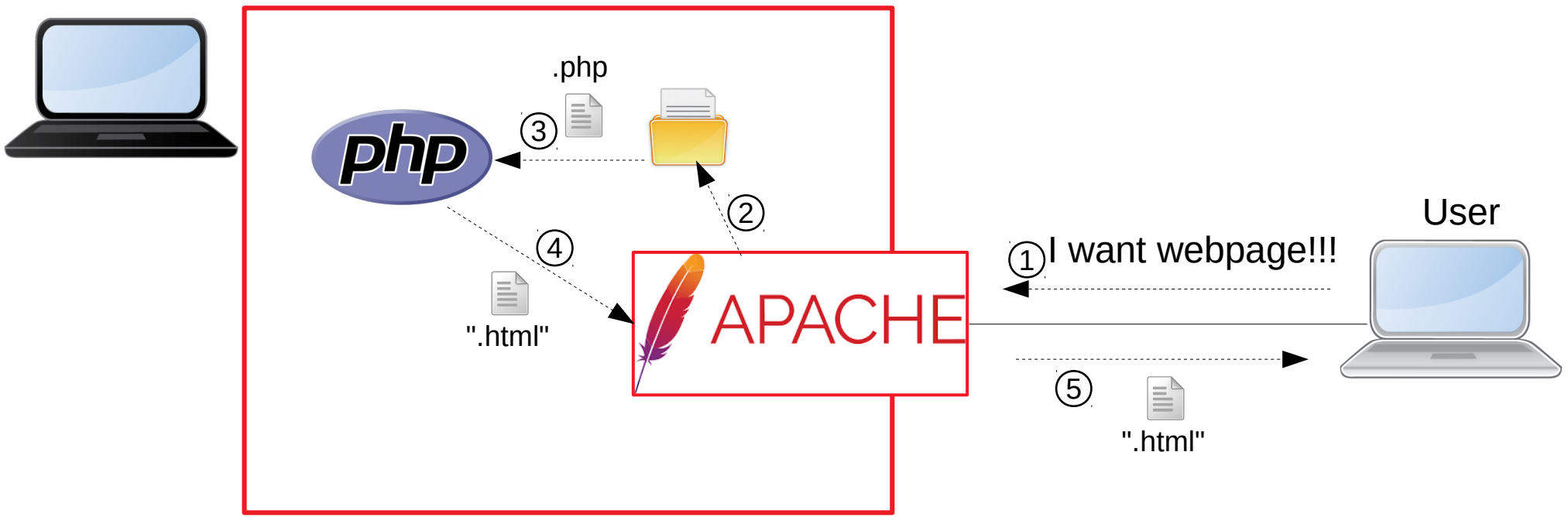
```
<html>
 <head>
  <title>webpage title</title>
 </head>
 <body>
  <b>my first webpage</b>
  <br>
  this is not bold
<?php
echo "<br>this is from php";
?>
 </body>
</html>
```

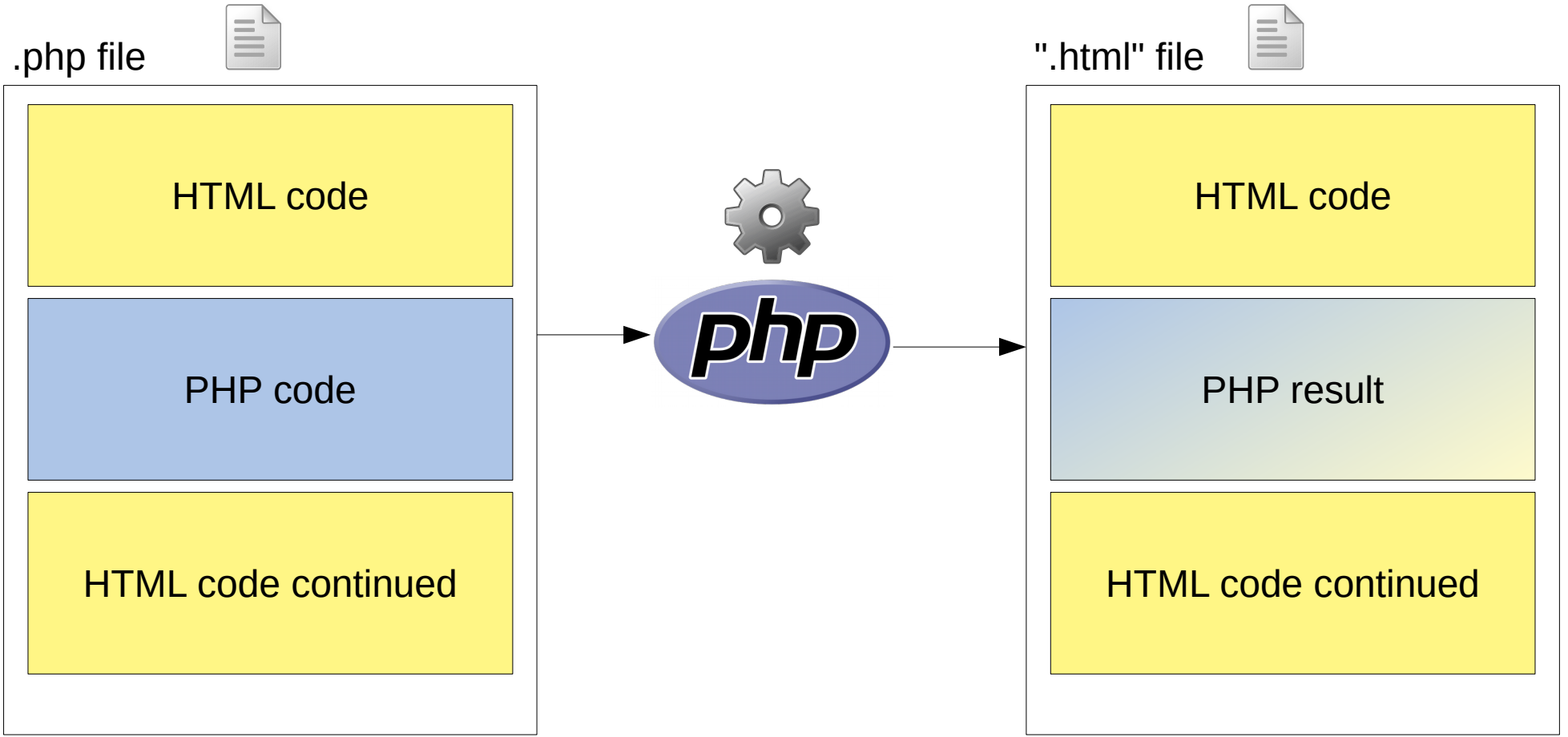
Purpose of PHP

- PHP is used to do programming tasks that HTML can't do
  - "Spits" out text which could include HTML code
  - echo function prints out text
    - similar print()/disp() in python/matlab

Purpose of PHP

- PHP is a server side language
  - User never sees the code, only the result




.php file
HTML code
PHP code
HTML code continued
php
".html" file
HTML code
PHP result
HTML code continued

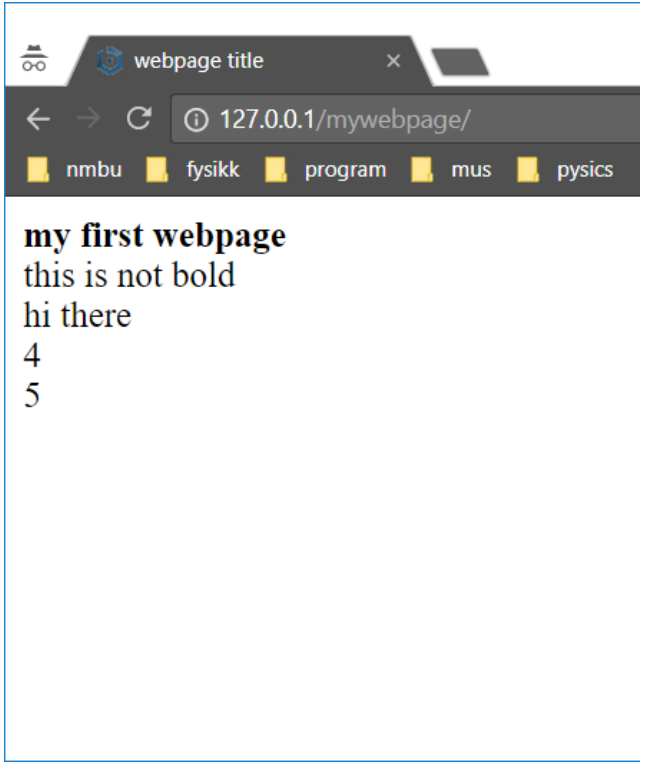
PHP

- Variables start with $
- Lines must end with ;
- Has loops (while/for)
- If statements
- echo function prints

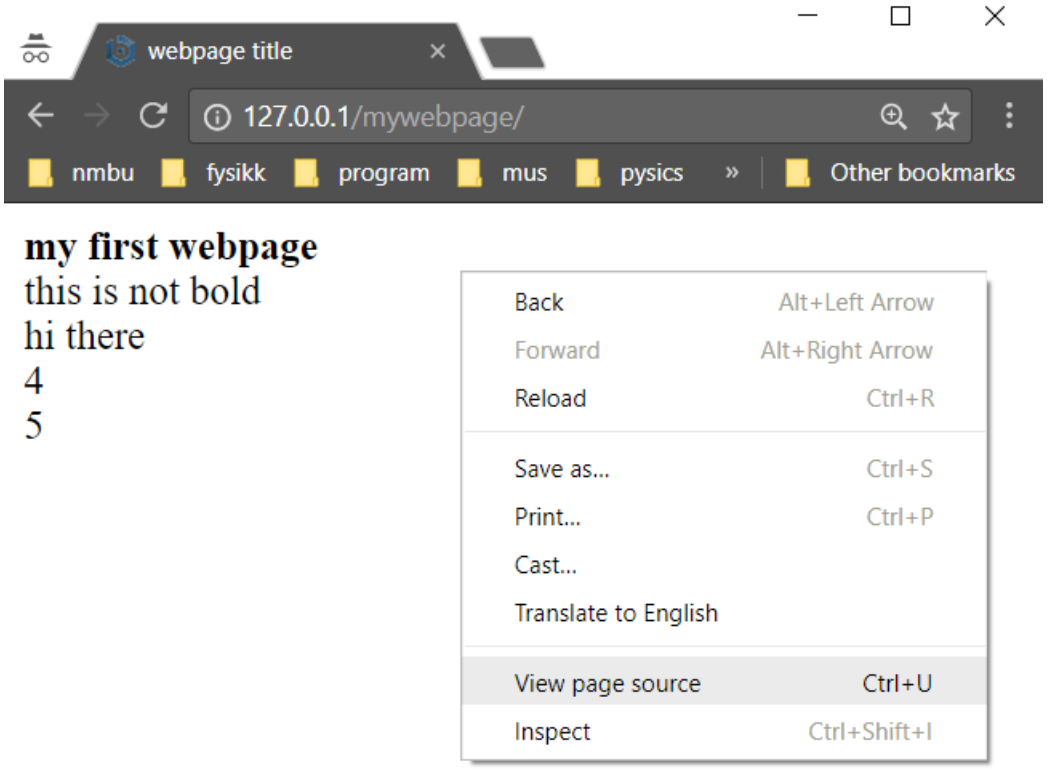
If the page isn't refreshing properly add opcache_reset(); to the top of the script (forces refresh)

PHP example

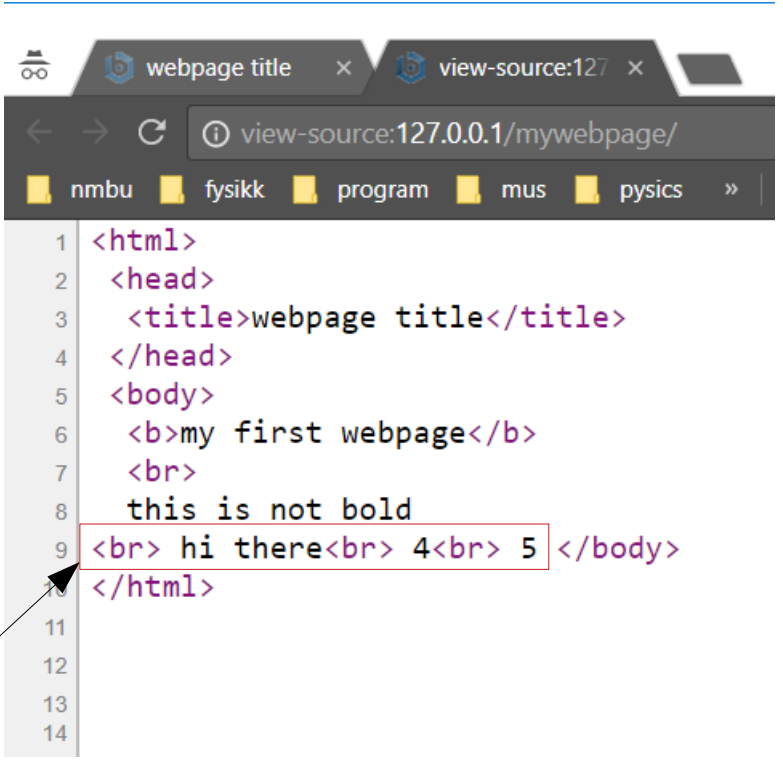
```
<?php
$anumbervariable = 4;
$atextvariable = "hi there";

echo "<br> $atextvariable";
echo "<br> $anumbervariable";

$anumbervariable = $anumbervariable +1;
echo "<br> $anumbervariable";

?>
```



PHP example



From PHP



Connecting to a database

```php
$servername = "localhost";
$username = "root";
$password = "inf130";
$dbname = "hobbyhuset";
```

Save connection info in variables
Localhost or 127.0.0.1

Connection function

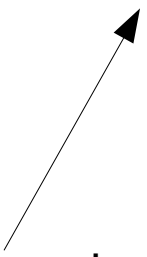
```php
$conn = new mysqli($servername, $username, $password, $dbname);
```

Connection is stored in the variable $conn

Connecting to a database

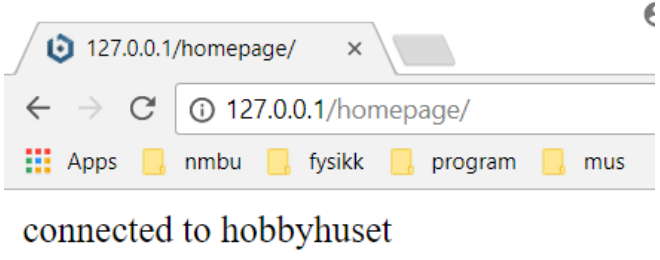
- Test to see if we connected or not

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```



Kills connection and prints error message

```php
<?php
opcache_reset();
$servername = "localhost";
$username = "root";
$password = "inf130";
$dbname = "hobbyhuset";

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

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else { echo "Connected to $dbname<br><br>";}
```



Sending queries with PHP

```php
$sql = "SELECT ansnr,fornavn FROM ansatt";
$result = $conn->query($sql);
```

Query function

Gets rows

```php
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "ansnr: " . $row["ansnr"]. " name: " . $row["fornavn"]."<br>";
    }
}
```

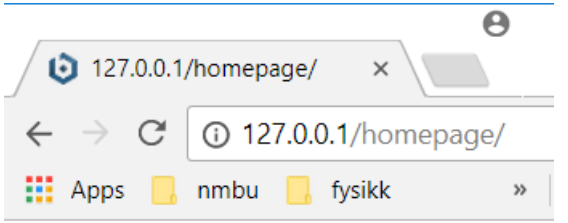
Prints each row

Get row value from column name

Lets implement a PHP mysql query

- Collect the employee number and name of the employees of hobbyhuset

```php
<html>
<body>
<?php
opcache_reset();
$servername = "localhost";
$username = "root";
$password = "inf130";
$dbname = "hobbyhuset";

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

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else { echo "Connected to $dbname<br><br>";}

$sql = "SELECT ansnr,fornavn FROM ansatt";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "ansnr: " . $row["ansnr"]. " name: " . $row["fornavn"]. "<br>";
    }
}
?>
</body>
</html>
```



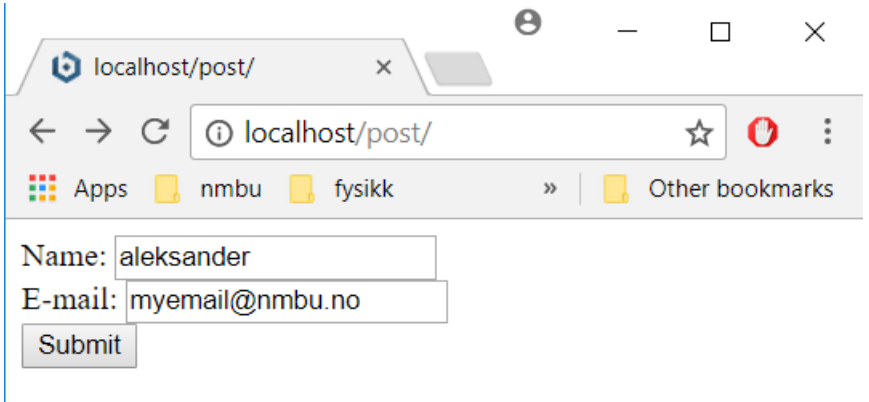
Connected to hobbyhuset

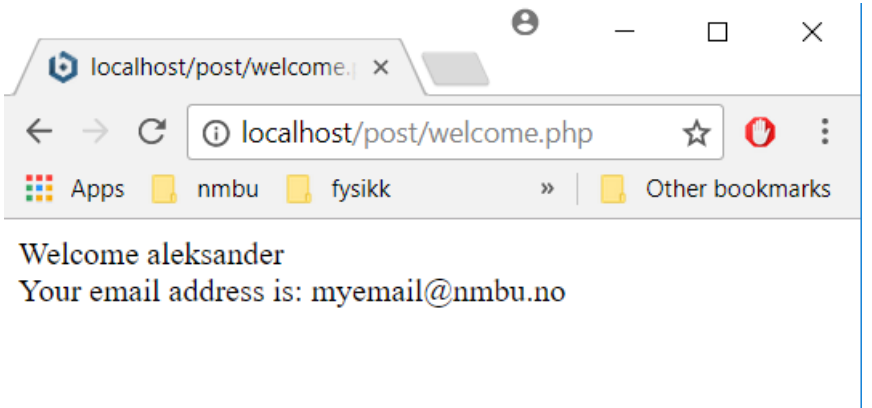
ansnr: 1 name: Georg
ansnr: 2 name: Gunnlaug
ansnr: 3 name: Morgan
ansnr: 6 name: Vilde
ansnr: 7 name: Henriette
ansnr: 8 name: Synøve
ansnr: 9 name: Ragnvald
ansnr: 11 name: Oliver
ansnr: 13 name: Oda
ansnr: 16 name: Andrine
ansnr: 17 name: Karl Anton
ansnr: 18 name: Johanna

Creating forms

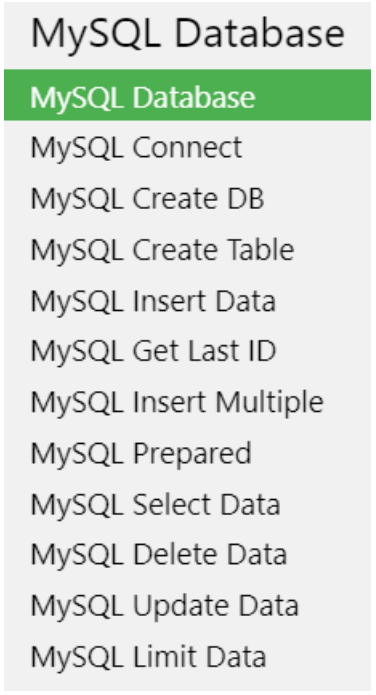
```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```





Other resources



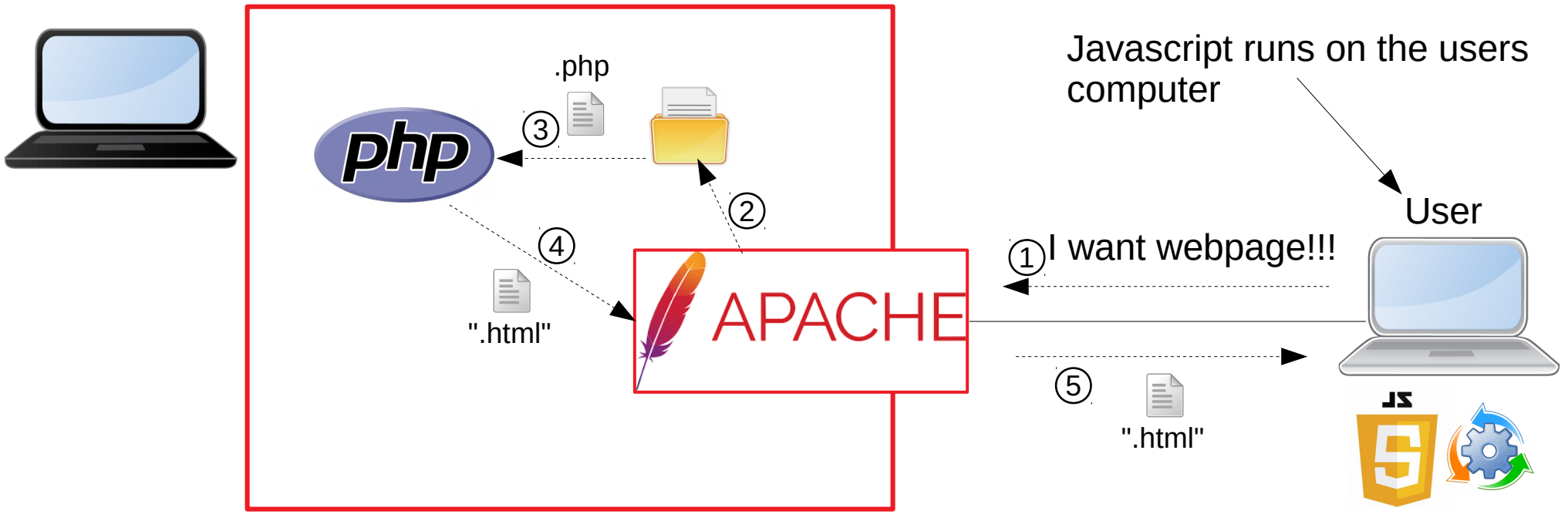
w3schools.com

https://www.w3schools.com/php/php_mysql_intro.asp

Javascript

- Can generate dynamic webpages
- Flexible
- Runs on the clients machine

Javascript


.php
php
3
2
4
".html"
APACHE
Javascript runs on the users computer
User
1 I want webpage!!!
5
".html"

Positives and negatives of Javascript

- Your server won't be overwhelmed if there are many users (php is run on the server)
- Dynamic web pages with lots of features

- Users can see your code