

# FROM DESKTOP TO WEB APP

Business College Helsinki

Kari KA Häkkinen

2019-09-15

# ALGORITHM

- algorithm is a description on how one specific problem will be solved
- in this material we'll use two different algorithms to solve one specific problem ...
- ... which is to sum figures from 1 to given number N

# ALGORITHM # 1: (SLOW)

- first algorithm is ...
  - easy to find out
  - slow when figures are big
- big  $O(n)$ 
  - linear time consumption
- set  $\text{sum} = 0$
- go through figures from 1 to N
  - calculate  
 $\text{sum} = \text{sum} + \text{current figure}$



# ALGORITHM #2: (FASTER)

- second algorithm is ...
  - not so obvious
  - much quicker when figures are big
- big  $O(1)$ 
  - constant time consumption
- set  $\text{sum} = 0$
- sum biggest and smallest (call it BisMa)
- multiply with integer division  $N/2$
- add half of BisMa, if  $N$  is odd

# SUMMING 1...N

Algorithm #1 [big  $O(N)$ ]:

Loop through 1 ... N

Sum them

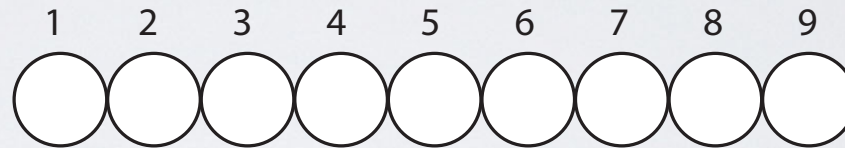
...

1 2 3 4

1 2 3

1 2

1



$$1+4=5$$

$$4:2=2$$

$$2 \times 5 = 10$$

$$\left( \begin{array}{l} 1+4=5 \\ (4 \div 2) \times \text{hesu} + (4 \bmod 2) * (\text{hesu}) = 2 \\ 2 \times 5 = 10 \end{array} \right)$$

$$1 \times (6/2)$$

$$1+5=6$$

$$5:2=2 \text{ kokonaista} + 1$$

$$2 \times 6 + (1 \times (6/2)) = 12$$

Algorithm #2 [big  $O(1)$ ]:

Smallest to be summed is **one**=1.

Biggest number to sum is **biggest**.

**helpersum=one+biggest**

Divide helpersum with two (**intdivpertwo=helpersum/2**)

Need remainder (modulus) too (**moduluspertwo=helpersum/2**)

# IMPLEMENTATION WITH C

```
[PoliTrukki:dataStructure&algorithm karikahakkinen$ ./conut 44444  
Summing numbers from 1 to 44444 ...
```

```
Algorithm #1: Sum is 987656790  
Time used: 0.000131
```

```
44444 - 987656790  
Algorithm #2: Sum is 987656790  
Time used: 0.000000
```

```
[PoliTrukki:dataStructure&algorithm karikahakkinen$ ./conut 44443  
Summing numbers from 1 to 44443 ...
```

```
Algorithm #1: Sum is 987612346  
Time used: 0.000145
```

```
44443 - 987612346  
Algorithm #2: Sum is 987612346  
Time used: 0.000001
```

```
#include <stdio.h>
#include <inttypes.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

#define BASE 10

int main(int argc, char **argv){

    char* endptr;
    long tocount;
    long i;
    long sum=0;
    long hesu;
    clock_t start, end;
    double cpu_time_used;

    if (argc<2){
        printf("%s: use: %s count\n", argv[0], argv[0]);
        exit(-1);
    }

    tocount = strtoumax(argv[1], &endptr, BASE);

    printf ("Summing numbers from 1 to %ld ...\n\n", tocount );

    start = clock();
    for (i=0; i<=tocount; i++){
        sum+=i;
    }
    end = clock();
    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

    printf ("Algorithm #1: Sum is %ld\n", sum );
    printf ("Time used: %lf\n\n", cpu_time_used);

    start = clock();
    hesu = (tocount+1) ;
    sum=((tocount/2)*hesu) + ((tocount%2)*(hesu/2));
    end = clock();
    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
    printf("%ld - %ld\n", tocount, sum);
    printf ("Algorithm #2: Sum is %ld\n", sum );
    printf ("Time used: %lf\n\n", cpu_time_used);
    exit(0);
}
```

**Learn:**  
**\* programming language C**



# WEB APPLICATION ENVIRONMENT

## WEB Application Environment



# JSON THAT WE'LL USE

```
{ "testing":  
  { "input": "4", "output":  
    [ { "sum": "800020000", "time":  
      "0.000121" },  
      { "sum": "800020000", "time": "  
0.000001" } ] } }
```

But is it *valid JSON*?

**Yes**, according

[jsonlint.com](https://jsonlint.com)

**Learn:**  
**\* validating JSON**

[jsonlint.com](https://jsonlint.com)

```
1 {  
2   "testing": {  
3     "input": "4",  
4     "output": [{  
5       "sum": "800020000",  
6       "time": "0.000121"  
7     }, {  
8       "sum": "800020000",  
9       "time": "0.000001"  
10    }]  
11  }  
12 }
```

Validate JSON

Clear

## Results

Valid JSON



# CREATING FIRST BACKEND

myjson.com

- It is easy to create first JSON backend by copying valid JSON to myjson.com
- then you can continue developing frontend without working, real backend

```
{
  "testing": {
    "input": "4",
    "output": [
      {
        "sum": "800020000",
        "time": "0.000121"
      },
      {
        "sum": "800020000",
        "time": "0.000001"
      }
    ]
  }
}
```

Your JSON was saved.

URI to access this JSON directly.

<https://api.myjson.com/bins/nfy29>



**Learn:**

**\* using internet services in program development**

# BUILDING OWN BACKEND #1

**Learn:**  
**\* basic html**

- it is easy to echo json to humans ...

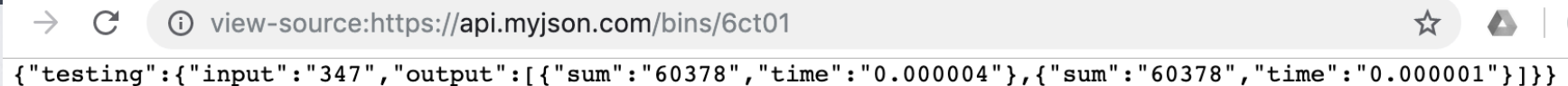
```
<!doctype html>
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <p>{"testing":{"input":"4","output":
[{"sum":"800020000","time":"0.000121"},
{"sum":"800020000","time":"0.000001"}]}}</p>
  </body>
</html>
```

# BUILDING OWN BACKEND #2

**Learn:**  
**\* using web browser  
developer tools**

- ... but it is not ok to our frontend!

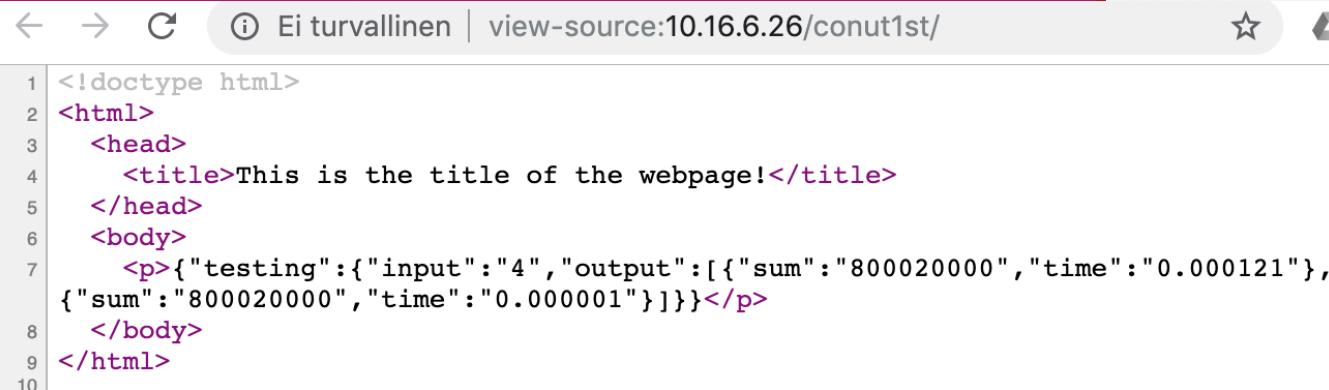
myjson.com source (this is good) json source



→ ↺ ⓘ view-source:https://api.myjson.com/bins/6ct01 ☆

```
{"testing":{"input":"347","output":[{"sum":"60378","time":"0.000004"}, {"sum":"60378","time":"0.000001"}]}}
```

our source (not good, but ok starter!)



← → ↺ ⓘ Ei turvallinen | view-source:10.16.6.26/conut1st/ ☆

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>This is the title of the webpage!</title>
5   </head>
6   <body>
7     <p>{"testing":{"input":"4","output":[{"sum":"800020000","time":"0.000121"},
8 {"sum":"800020000","time":"0.000001"}]}}</p>
9   </body>
10  </html>
```



# ALGORITHMS, IMPLEMENTED USING PHP

This echoes still  
html page - not  
json as meta!

**Learn:**  
**\* php functions**

```
<!doctype html>
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <?php
      function soslow(int $nmbrin) {
        // https://www.php.net/manual/en/function.microtime.php
        $start = microtime(true);
        $sum=0;
        for ($i=0;$i<=$nmbrin;$i++){
          $sum += $i;
        };
        $time_elapsed_secs = microtime(true) - $start;
        $time=sprintf("%f",$time_elapsed_secs);
        return ('{"sum":"' . $sum . '", "time":"' . $time . '"}');
      }

      function faster(int $tocount) {
        $start = microtime(true);
        $sum=0;
        $hesu = ($tocount+1) ;
        $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*($hesu/2));
        $time_elapsed_secs = microtime(true) - $start;
        $time=sprintf("%f",$time_elapsed_secs);
        return ('{"sum":"' . $sum . '", "time":"' . $time . '"}');
      }

      $in=40000000;
      $jsonStr = '{"testing":{"input":"' . $in . '", "output":[';
      $jsonStr .= soslow($in);
      $jsonStr .= ',';
      $jsonStr .= faster($in);
      $jsonStr .= ']]}' ;
      echo '<p>' . $jsonStr . '</p>';

    ?>
  </body>
</html>
```

# ALGORITHMS ... PHP IMPLEMENTED - JSON BACKEND

echoes JSON

json as  
content-type

**Learn:**

**\* www content-types**

```
<?php

// https://alexwebdevelop.com/php-json-backend/

function soslow(int $nmbrin) {
    // https://www.php.net/manual/en/function.microtime.php
    $start = microtime(true);
    $sum=0;
    for ($i=0;$i<=$nmbrin;$i++){
        $sum += $i;
    };
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"' . $sum . '", "time":"' . $time . '"}');
}

function faster(int $tocount) {
    $start = microtime(true);
    $sum=0;
    $hesu = ($tocount+1) ;
    // PHP integer division intdiv()-> https://secure.php.net/manual/en/function.intdiv.php
    $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*(intdiv($hesu,2)));
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"' . $sum . '", "time":"' . $time . '"}');
}

https://www.w3schools.com/php/php_superglobals.asp
echo 'given number was -> ' . $_GET['number'] ;
$in=$_GET['number'];

$jsonStr  = '{"testing":{"input":"' . $in . '", "output":[';
$jsonStr .= soslow($in);
$jsonStr .= ',';
$jsonStr .= faster($in);
$jsonStr .= ']}]';

header('Content-Type: application/json'); //json backend header
echo $jsonStr;

?>
```

# TESTING BACKEND

**Learn:**  
\* **Postman**  
\* **curl**

The screenshot shows the Postman application interface. At the top, there's a toolbar with buttons for 'New', 'Import', 'Runner', and 'My Workspace'. Below this, the URL bar shows a GET request to `http://10.16.6.26/conut3rd/index.php?number=345`. The 'Send' button is visible. Below the URL bar, there's a tabbed interface with 'Params', 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Params' tab is active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The table contains one row with 'number' as the key and '345' as the value. Below the table, there's a 'Body' tab with 'Pretty', 'Raw', and 'Preview' options. The 'Pretty' option is selected, showing a JSON response in a code editor. The response is a JSON object with a 'testing' property containing an object with 'input' and 'output' properties. The 'output' property is an array of two objects, each containing 'sum' and 'time' properties.

```
{
  "testing": {
    "input": "345",
    "output": [
      {
        "sum": "59685",
        "time": "0.000003"
      },
      {
        "sum": "59685",
        "time": "0.000001"
      }
    ]
  }
}
```



# FIRST FRONTEND

## Learn:

- \* **basic JavaScript**
- \* **AJAX**
- \* **URLs and URIs**
- \* **GETs, PUTs, ...**
- \* **CRUD principles (RESTless API)**

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

```
<h2>The XMLHttpRequest Object</h2>
```

```
<button type="button" onclick="loadDoc()">Request data (4 in this example)</button>
```

```
<p id="demo"></p>
```

```
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "http://192.168.100.15/conut3rd/index.php?number=4", true);
  xhttp.send();
}
</script>
```

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

# SECOND FRONTEND

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Enter positive integer into field above!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://192.168.100.15/conut3rd/index.php?number="+numberValue
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("demo").innerHTML = this.responseText;
        }
      };
      xhttp.open("GET", URL, true);
      xhttp.send();
    }
  </script>

</body>
</html>
```

**Learn:**  
\* **basic JavaScript**  
\* **AJAX**

# THIRD FRONTEND

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object
</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Enter positive integer into field above!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://10.16.6.26/conut3rd/index.php?number="+numberValue;
      var XHR = new XMLHttpRequest();
      XHR.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          let jsonObj = JSON.parse(this.responseText);
          let outPutText="With input data "+jsonObj.testing.input+":<br><br>"+
            "Algorithm #1 gives result "+jsonObj.testing.output[0].sum +
            " in time "+jsonObj.testing.output[0].time+":<br><br>"+
            "Algorithm #2 gives result "+jsonObj.testing.output[1].sum +
            " in time "+jsonObj.testing.output[1].time + "<br><br>";
          document.getElementById("demo").innerHTML = outPutText;
        }
      };
      XHR.open("GET", URL, true);
      XHR.send();
    }
  </script>
</body>
</html>
```

parse JSON string -> JS object...

... and use it

## Learn:

- \* objects principles
- \* OOP principles
- \* JSON parsing in JavaScript
- \* JavaScript objects



# DATABASE BASICS

What is data,  
database, DBMS

[https://www.guru99.com/  
introduction-to-database-  
sql.html](https://www.guru99.com/introduction-to-database-sql.html)

← → ↻ 🔒 <https://www.guru99.com/introduction-to-database-sql.html> ☆ 🗑️ 🌐

consideration.

For example your name, age, height, weight, etc are some data related to you.

A picture , image , file , pdf etc can also be considered data.

## What is a Database?

Database is a systematic collection of data. Databases support storage and manipulation of data. Databases make data management easy. Let's discuss few examples.

An online telephone directory would definitely use database to store data pertaining to people, phone numbers, other contact details, etc.

Your electricity service provider is obviously using a database to manage billing , client related issues, to handle fault data, etc.

Let's also consider the facebook. It needs to store, manipulate and present data related to members, their friends, member activities, messages, advertisements and lot more.

We can provide countless number of examples for usage of databases .

## SQL Tutorials

1) Introduction

2) MySQL  
Workbench

3) Database  
Designing

4) Normalization

5) ER Modeling

6) Creating a  
Database

7) SELECT  
Statement

8) WHERE  
Clause

9) INSERT INTO

10) DELETE &  
UPDATE

11) ORDER BY,  
ASC, DESC

# LEARNING SQL

www.sqlcourse.com

← → ↻ ⓘ Ei turvallinen | www.sqlcourse.com 🔍 ☆ 🏠 👤 🔴

Advertisement

READY YOUR ENTERPRISE  
FOR THE API REVOLUTION



**SQLCourse.com**<sup>®</sup>  
Interactive Online SQL Training

HOME News MS SQL Oracle DB2 Access MySQL PHP SQL Etc Sc

**DATABASE JOURNAL**

**SQL Courses**

- 1 [What is SQL?](#)
- 2 [Table basics](#)
- 3 [Selecting data](#)
- 4 [Creating tables](#)
- 5 [Inserting into a table](#)
- 6 [Updating records](#)
- 7 [Deleting records](#)
- 8 [Drop a table](#)
- 9 [Advanced Queries](#)
- 10 [Standalone SQL interpreter](#)
- 11 [Advertise on SQLCourse.com](#)
- 12 [Database Links](#)

**Dbasupport.com**<sup>™</sup>

**SQLCourse2.com**<sup>™</sup>

**Welcome to SQLCourse.com!**

This unique introductory SQL tutorial not only to-understand S

AdChoices

www.sqlcourse2.com

← → ↻ ⓘ Ei turvallinen | www.sqlcourse2.com 🔍 ☆ 🏠 👤 🔴

Advertisement

THE CHANGING FACE OF BI: PR  
COGNITIVE SERVICES, AND AI



**SQLCourse2.com**<sup>™</sup>  
Interactive Online SQL Training

HOME News MS SQL Oracle DB2 Access MySQL PHP SQL Etc Scripts Links F

**DATABASE JOURNAL**

**SQL Courses**

- 1 [Start Here - Intro](#)
- 2 [SELECT Statement](#)
- 3 [Aggregate Functions](#)
- 4 [GROUP BY clause](#)
- 5 [HAVING clause](#)
- 6 [ORDER BY clause](#)
- 7 [Combining Conditions & Boolean Operators](#)
- 8 [IN and BETWEEN](#)
- 9 [Mathematical Functions](#)
- 10 [Table Joins, a must](#)
- 11 [SQL Interpreter](#)
- 12 [Advertise on SQLCourse.com](#)
- 13 [Other Tutorial Links](#)

**Dbasupport.com**<sup>™</sup>

**SQLCourse.com**<sup>®</sup>

**Welcome to SQLCourse2.com!**

This unique SQL Tutorial is the "sequel" to the highly successful [SQLCourse.com](#) site

AdChoices

# PHP CONNECTION TO SQL DATABASE

[https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)

<https://www.php.net/manual/en/book.pdo.php>

← → ↻ <https://www.php.net/manual/en/book.pdo.php>

php

Downloads

Documentation

Get Involved

Help

• [PDOException](#) — The PDOException class

• [PDO Drivers](#)

• [CUBRID \(PDO\)](#) — CUBRID Functions (PDO\_CUBRID)

• [MS SQL Server \(PDO\)](#) — Microsoft SQL Functions (PDO\_MYSQL)

• [Firebird \(PDO\)](#) — Firebird Functions (PDO\_FIREBIRD)

• [IBM \(PDO\)](#) — IBM Functions (PDO\_IBM)

• [Informix \(PDO\)](#) — Informix Functions (PDO\_INFORMIX)

• [MySQL \(PDO\)](#) — MySQL Functions (PDO\_MYSQL)

• [MS SQL Server \(PDO\)](#) — Microsoft SQL (PDO\_SQLSRV)

• [Oracle \(PDO\)](#) — Oracle Functions (PDO\_ORACLE)

• [ODBC and DB2 \(PDO\)](#) — ODBC and DB2

• [PostgreSQL \(PDO\)](#) — PostgreSQL Functions (PDO\_PGSQL)

• [SQLite \(PDO\)](#) — SQLite Functions (PDO\_SQLITE)

• [4D \(PDO\)](#) — 4D Functions (PDO\_4D)

← → ↻ [https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)



HTML

CSS

JAVASCRIPT

SQL

PYTHON

PHP

MORE ▾

PHP Advanced

PHP Arrays Multi

PHP Date and Time

PHP Include

PHP File Handling

PHP File Open/Read

PHP File Create/Write

PHP File Upload

PHP Cookies

PHP Sessions

PHP Filters

PHP Filters Advanced

MySQL Database

MySQL Database

MySQL Connect

MySQL Create DB

MySQL Create Table

MySQL Insert Data

## Example (PDO)

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

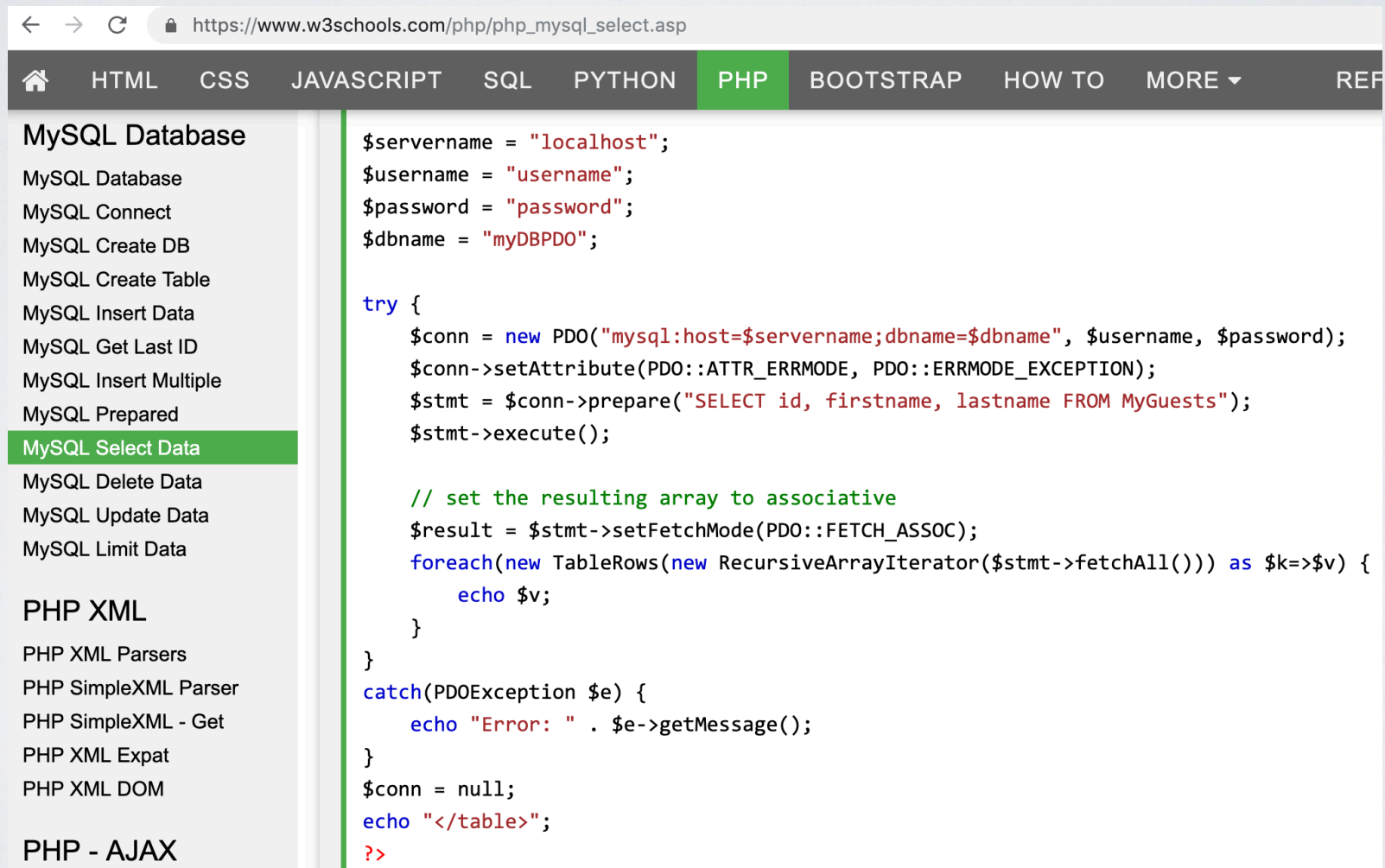
try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username,
$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
}
catch(PDOException $e)
{
    echo "Connection failed: " . $e->getMessage();
}

?>
```



# PHP PDO SELECT

[https://www.w3schools.com/php/php\\_mysql\\_select.asp](https://www.w3schools.com/php/php_mysql_select.asp)



← → ↻ 🔒 [https://www.w3schools.com/php/php\\_mysql\\_select.asp](https://www.w3schools.com/php/php_mysql_select.asp)

🏠 HTML CSS JAVASCRIPT SQL PYTHON **PHP** BOOTSTRAP HOW TO MORE ▾ REF

**MySQL Database**

- MySQL Database
- MySQL Connect
- MySQL Create DB
- MySQL Create Table
- MySQL Insert Data
- MySQL Get Last ID
- MySQL Insert Multiple
- MySQL Prepared
- MySQL Select Data**
- MySQL Delete Data
- MySQL Update Data
- MySQL Limit Data

**PHP XML**

- PHP XML Parsers
- PHP SimpleXML Parser
- PHP SimpleXML - Get
- PHP XML Expat
- PHP XML DOM

**PHP - AJAX**

```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {
        echo $v;
    }
}
catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

# be01.php

```
<?php
// https://alexwebdevelop.com/php-json-backend/
function soslow(int $nmbrin) {
    // https://www.php.net/manual/en/function.microtime.php
    $start = microtime(true);
    $sum=0;
    for ($i=0;$i<=$nmbrin;$i++){
        $sum += $i;
    };
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
}

function faster(int $tocount) {
    $start = microtime(true);
    $sum=0;
    $hesu = ($tocount+1) ;
    // PHP integer division intdiv()-> https://secure.php.net/manual/en/function.intdiv.php
    $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*(intdiv($hesu,2)));
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
}

// https://www.w3schools.com/php/php_superglobals.asp
// echo 'given number was -> ' . $_GET['number'] ;
$in=$_GET['number'];

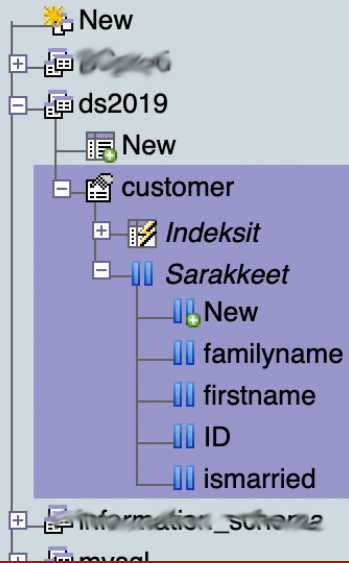
$jsonStr = '{"testing":{"input":"' . $in . '","output":[';
$jsonStr .= soslow($in);
$jsonStr .= ',';
$jsonStr .= faster($in);
$jsonStr .= ']}';

header('Content-Type: application/json'); //json backend header
echo $jsonStr;
?>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Syötä positiivinen luku yläpuolella olevaan kenttään!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://localhost:8888/datastructures/backend/be01.php?number="+numberValue;
      var xhr = new XMLHttpRequest();
      xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
          let jsonObj = JSON.parse(xhr.responseText);
          let outPutText="Syötteellä " + jsonObj.testing.input + "<br><br>" +
            "algoritmi #1 antoi tulokseksi " + jsonObj.testing.output[0].sum +
            " ajassa " + jsonObj.testing.output[0].time + ".<br><br>" +
            "Algoritmi #2 antoi tulokseksi " + jsonObj.testing.output[1].sum +
            " ajassa " + jsonObj.testing.output[1].time + "<br><br>";
          document.getElementById("demo").innerHTML = outPutText;
        }
      };
      xhr.open("GET", URL, true);
      xhr.send();
    }
  </script>
</body>
</html>
```





#	Nimi	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Kommentit	Lisätiedot
1	ID	int(9)			Ei	None		AUTO_INCREMENT
2	firstname	varchar(60)	utf8_swedish_ci		Ei	None		
3	familyname	varchar(30)	utf8_swedish_ci		Ei	None		
4	ismarried	tinyint(1)			Ei	None		

Add 1 column(s) after ismarried Siirry

Toiminnot	Avaimen nimi	Tyyppi	Uniikki	Pakattu	Sarake	Kardinaliteetti	Aakkosjärjestys	Tyhjä	Kom
Muokkaa Tuhoa	PRIMARY	BTREE	Kyllä	Ei	ID	4	A	Ei	

Structure

DATA  
BASE

Data

phpMyAdmin

Palvelin: localhost:8889 » Tietokanta: ds2019 » Taulu: customer

Selaa Rakenne SQL Etsi Lisää rivi Vienti Tuonti Käyttöoikeudet

✓ Näytetään rivit 0– 4 (5 tulos, Kysely kesti 0.0015 sekuntia.)

`SELECT * FROM `customer``

Profilointi [Muokkaa inline-koodia] [Muokkaa] [Selitä]

Näytä kaikki | Rivien määrä: 25 Suodata rivejä: Hae tästä taulusta Lajitt

+ Valinnat

	ID	firstname	familyname	ismarried
Muokkaa Kopioi Poista	1	Douglas	Fairbanks	1
Muokkaa Kopioi Poista	2	Paulette	Goddard	0
Muokkaa Kopioi Poista	3	Mary	Pickford	1
Muokkaa Kopioi Poista	4	Charlie	Chaplin	1
Muokkaa Kopioi Poista	5	Harold	Lloyd	0

Valitse kaikki Valitut: Muokkaa Kopioi Poista Vienti

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Write actors id into field above!</p>
```

```
<script>
function loadDoc() {
  numberValue = document.getElementById("number").value;
  URL="http://localhost:8888/datastructures/backend/be02.php?actorid="+ numberValue;
  var xhr = new XMLHttpRequest();
  xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
      let outPutText = xhr.responseText;
      document.getElementById("demo").innerHTML = outPutText;
    }
  };
  xhr.open("GET", URL, true);
  xhr.send();
}
</script>
```

**be02.php** on next slide >>

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

```
<?php
// https://www.w3schools.com/php/php_superglobals.asp
// echo 'given actorid was -> ' . $_GET['number'];
$in=$_GET['actorid'];
$servertime = "mysql";
$servername = "localhost";
$username = "ds2019";
$password = "TopSecret";
$dbname = "ds2019";
```

```
try {
    $conn = new PDO("$servertime:host= $servername;dbname= $dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $conn = $conn->prepare("SELECT ID, firstname, familyname, ismarried FROM `customer` WHERE ID=$in");
    $conn->execute();

    // set the resulting array to associative
    $result = $conn->setFetchMode(PDO::FETCH_ASSOC);

    foreach ($conn->fetchAll() as $k=>$v){ // Associative Array https://www.w3schools.com/php/php\_arrays.asp
        $booleanIsMarried = (1 == $v['ismarried']) ? 'true' : 'false'; // ternary operator
        $output= '{"firstname":"'.$v['firstname'].'","familyname":"'.$v['familyname'].'","ismarried":"'.$booleanIsMarried.'"}';
    }
}

catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}

header('Content-Type: application/json'); //json backend header
echo $output;
?>
```