Webtechnology crash course

Thomas Helling| Leiden

Business Intelligence & Proces Modeling

Universiteit Leiden

Wie van jullie heeft al gewerkt met....

HTML?

CSS ?

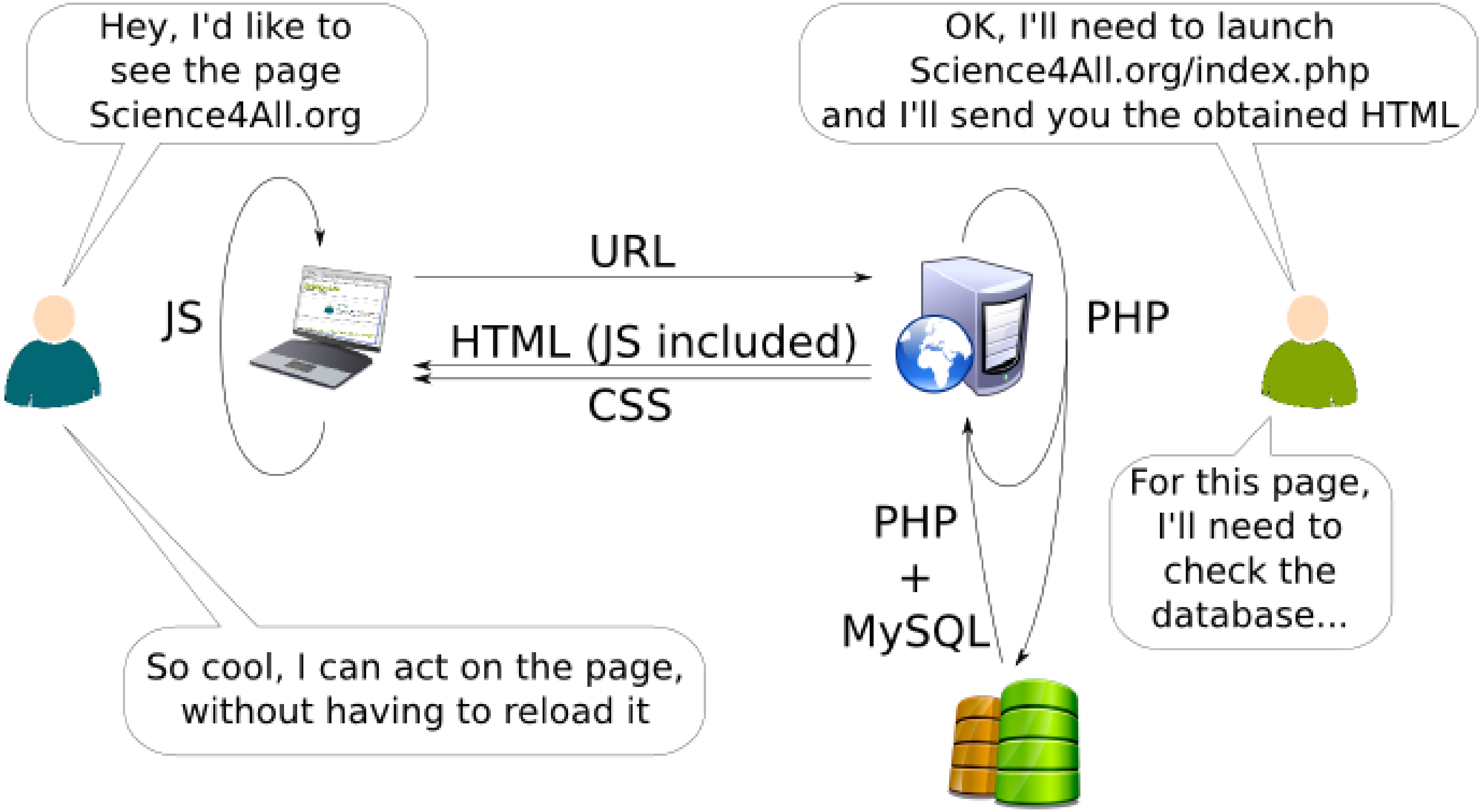
Bootstrap?

PHP?

MySQL?

jQuery & JSON?

Hoe het werkt



Hey, I'd like to see the page Science4All.org

OK, I'll need to launch Science4All.org/index.php and I'll send you the obtained HTML

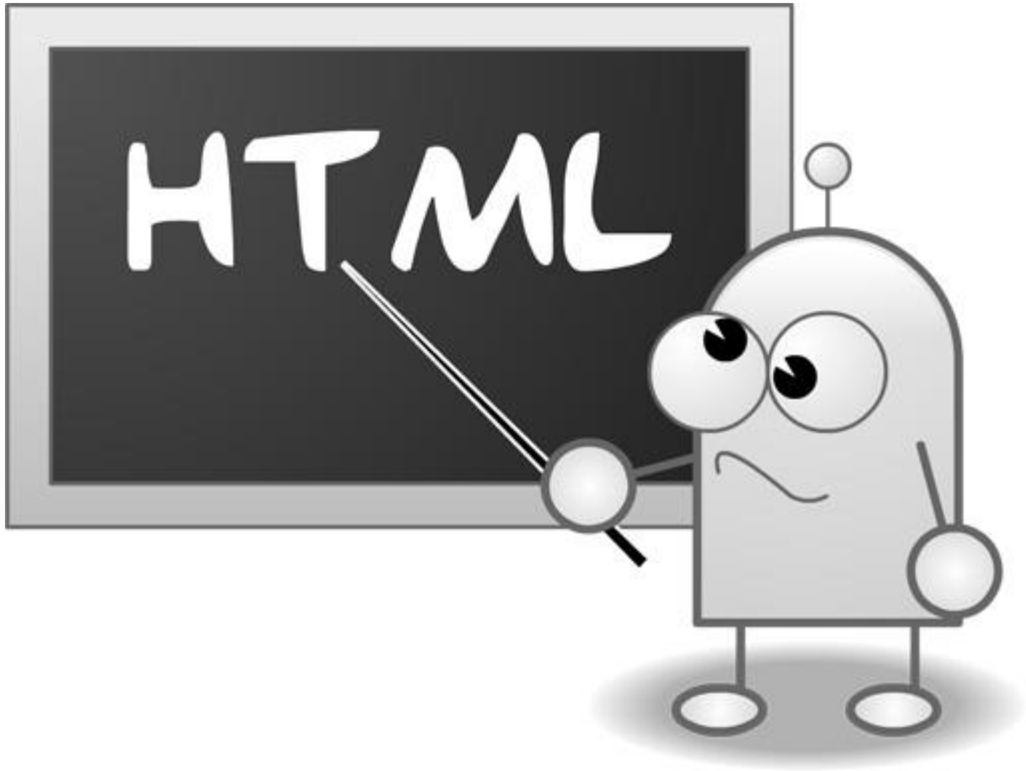
So cool, I can act on the page, without having to reload it

For this page, I'll need to check the database...

HTML

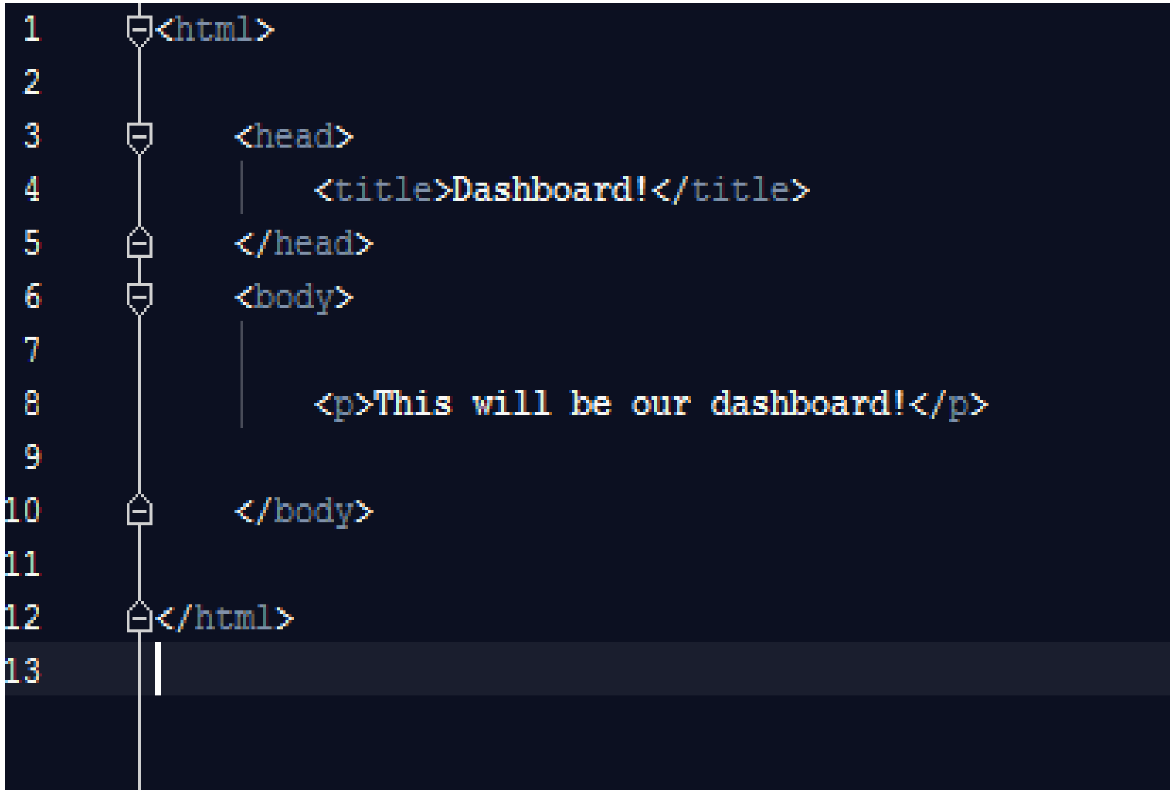
Hyper**T**ext **M**arkup **L**anguage

- Interpreted by browser
- Client-side
- Content of page



Simple HTML Page

Input



```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body>

        <p>This will be our dashboard!</p>

    </body>

</html>

```

Output

This will be our dashboard!

Tags HTML

Unaire tags: <tagname />

e.g. <image /> of <br />

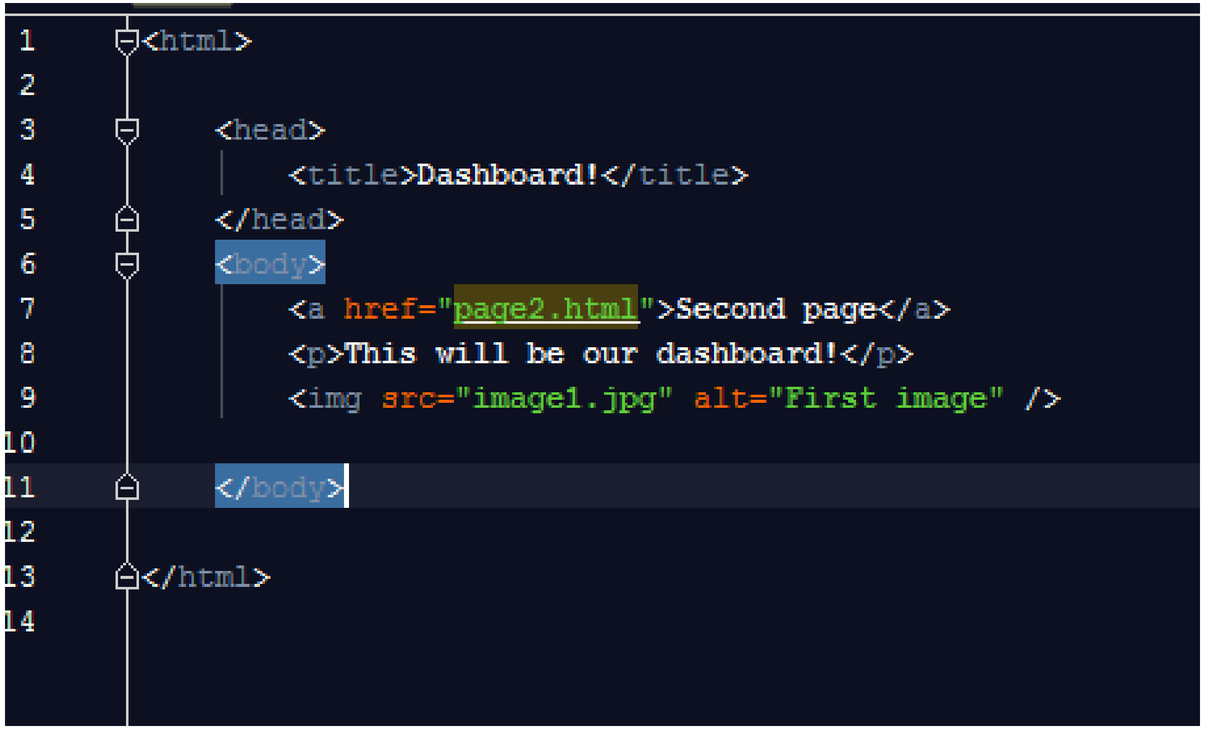
Binairy tags: <tagname>Something</tagname>

e.g. <a>a link</a> or <p>Hello</p>

Attributes: <tag attribuut="value" />

e.g. <a href="page2.html">My second page</a> or

<img src="foto.jpg" alt="foto" />



```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body>
        <a href="page2.html">Second page</a>
        <p>This will be our dashboard!</p>
        <img src="image1.jpg" alt="First image" />

    </body>

</html>
```

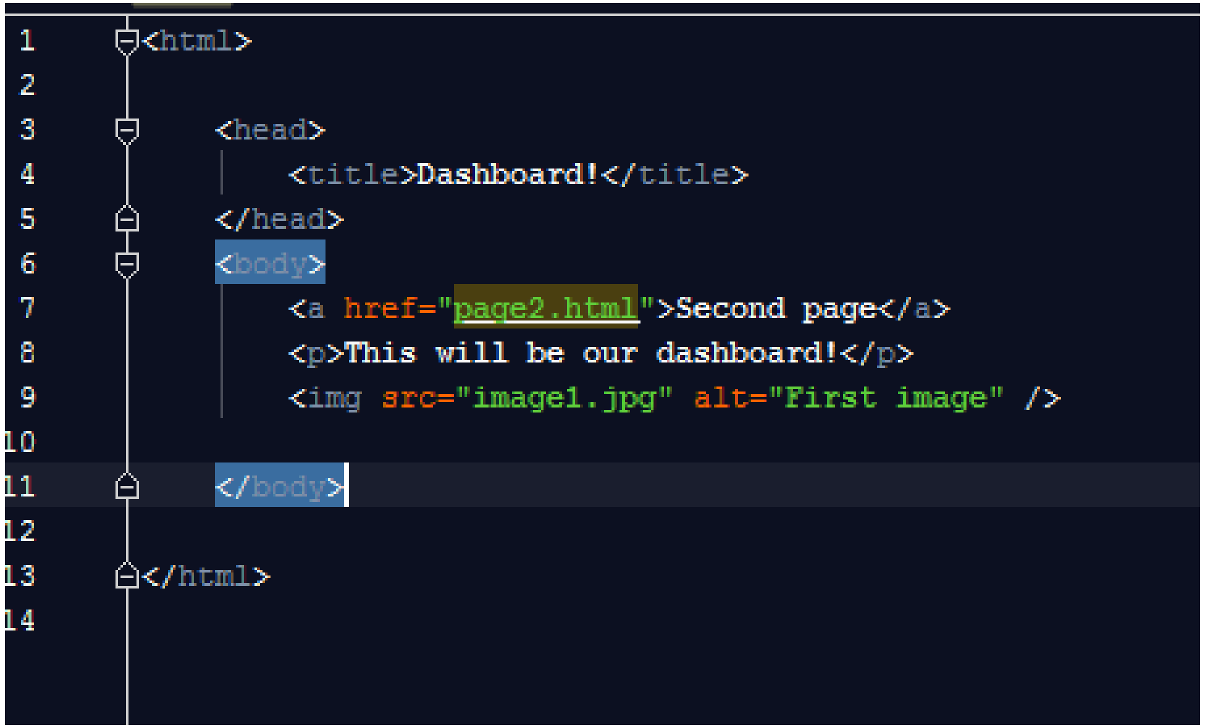
All attributes can be found on www.w3schools.com

Tags HTML

Paragraph	<p></p>
Regeleinde	<br />
Horizontale lijn	<hr />
Heading 1-6	<h1>, <h2>, … <h6>
Bold, italics, underline	<b></b>, <i></i>, <u></u>*
Bold, italics	<strong></strong>, <em></em>
Superscript	<sup>hoog</sup>
Subscript	<sub>laag</sub>
Commentaar	<!-- onzichtbaar -->
Divider	**<div></div>**

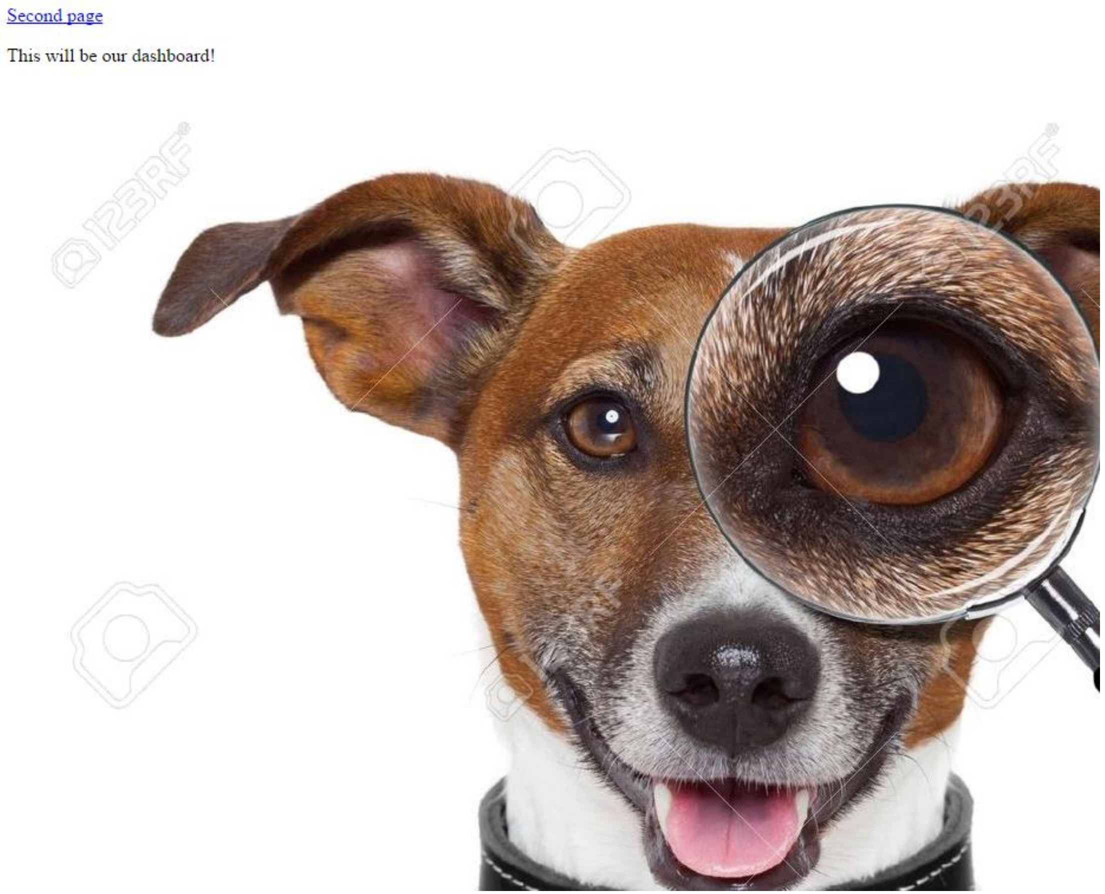
Attributes HTML

Input



```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body>
        <a href="page2.html">Second page</a>
        <p>This will be our dashboard!</p>
        <img src="image1.jpg" alt="First image" />

    </body>

</html>

```

Output

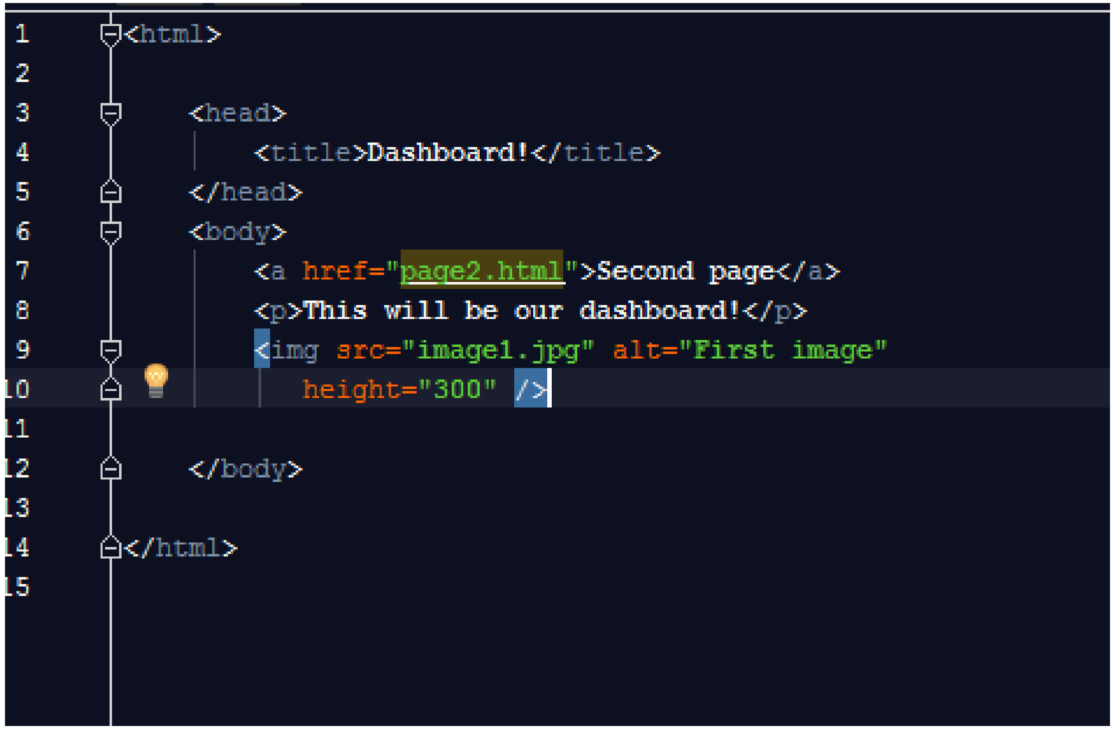


Second page

This will be our dashboard!

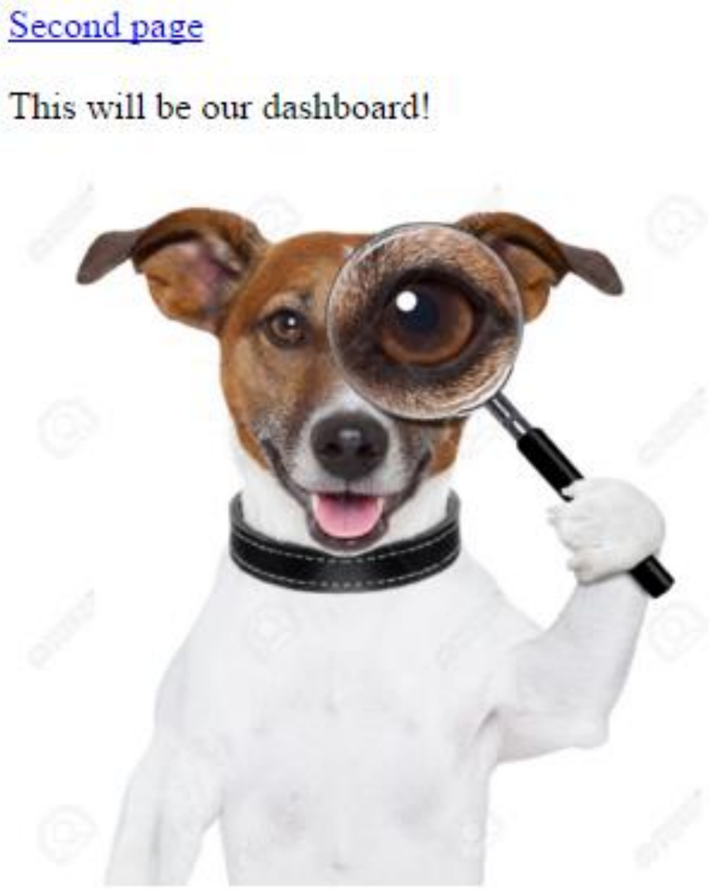
Attributes HTML

Input



```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body>
        <a href="page2.html">Second page</a>
        <p>This will be our dashboard!</p>
        <img src="image1.jpg" alt="First image"
            height="300" />

    </body>

</html>

```

Output



Second page

This will be our dashboard!

Vuistregels

- Tags altijd netjes nesten
  - Fout: `<b><i>tekst</b></i>`
  - Goed: `<b><i>tekst</i></b>`
- Alle tags en attribuutnamen in kleine letters
- Binaire tags niet vergeten af te sluiten
- Unaire tags niet vergeten af te sluiten:
  - Fout (beetje): `<br>` Goed: `<br />`
- Fout: `<font>`, `<center>` en `<s>`
- Inspringen na een geopende tag voor leesbaarheid

CSS

Cascading **S**tyle **S**heet

- Styling of page

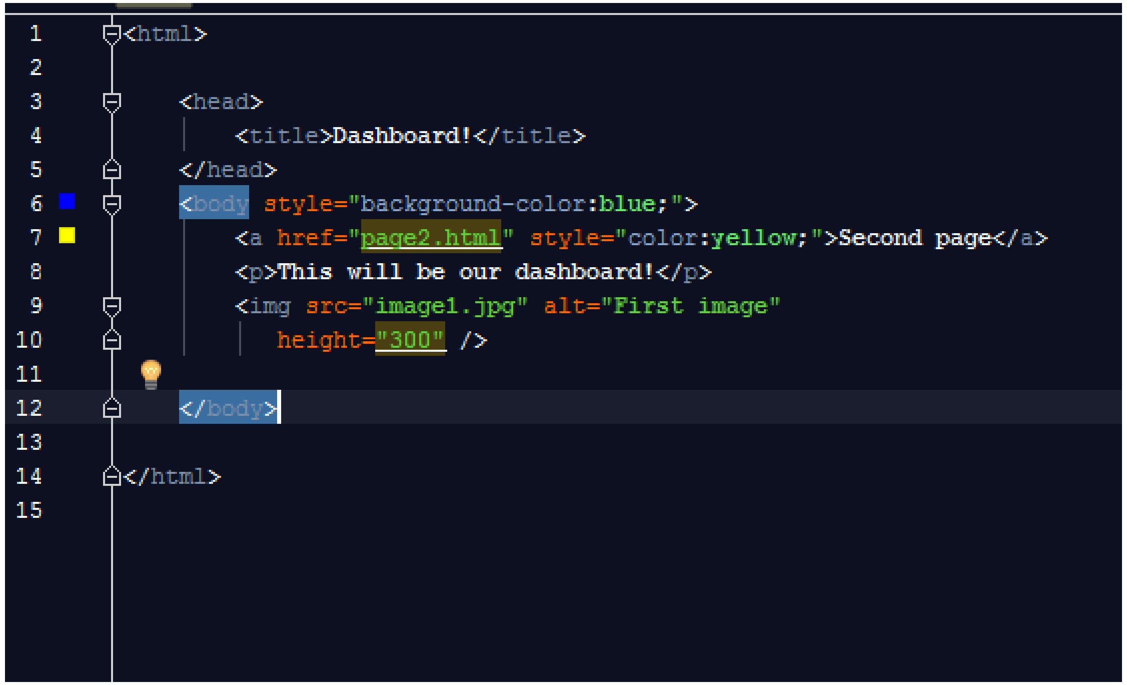
- Hoe?

1. Via het soort tag, bijv voor `p` of `img`, voor het toepassen, in css via **`tagnaam`**
2. Via het attribuut `id` van een tag, voor één bepaalde tag, in css via **`#idnaam`**
3. Via het attribuut `class` van een tag, voor meerdere deze tags, in css via **`.classnaam`**

`http://www.w3schools.com/cssref/default.asp`

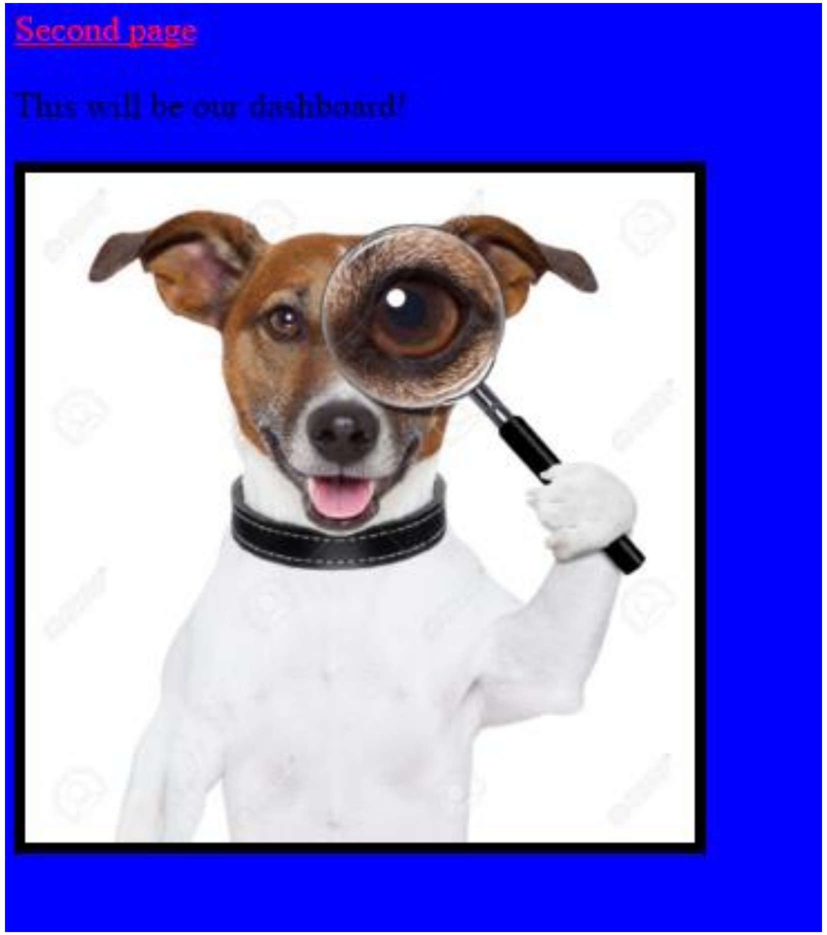
CSS example

Input



```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body style="background-color:blue;">
        <a href="page2.html" style="color:yellow;">Second page</a>
        <p>This will be our dashboard!</p>
        <img src="image1.jpg" alt="First image"
           height="300" />

    </body>

</html>
```

Output



Second page

This will be our dashboard!

Voorbeeld CSS met HTML

Style als attribuut gebruiken in tag.

HTML

```
<html>
 <body>
  <h1>Welkom</h1>
  <h2 id="tip">Tip</h2>
  <p>Dit is een normale
  tekst</p>
  <p class="mooi">Dit
  is een mooie tekst
  </p>
 </body>
</html>
```

CSS

```
body {
 background: yellow;
}
#tip {
 font-weight: bold;
 color: blue;
}
p {
 font-size: 16px;
}
.mooi {
 font-style: italic;
}
```

Id/class/div/span

- `id` bij een unieke tag
  Meerdere tags met hetzelfde `id` *mag niet*
- `class` bij opmaak-soortgelijke tags
  Meerdere tags met dezelfde `class` *mag wel*
- `class` waarden concateneren: spatie
- Combinaties zijn dus toegestaan, bijv:

```
<p id=“header" class="belangrijk mooi">Dit is een tekst</p>
```

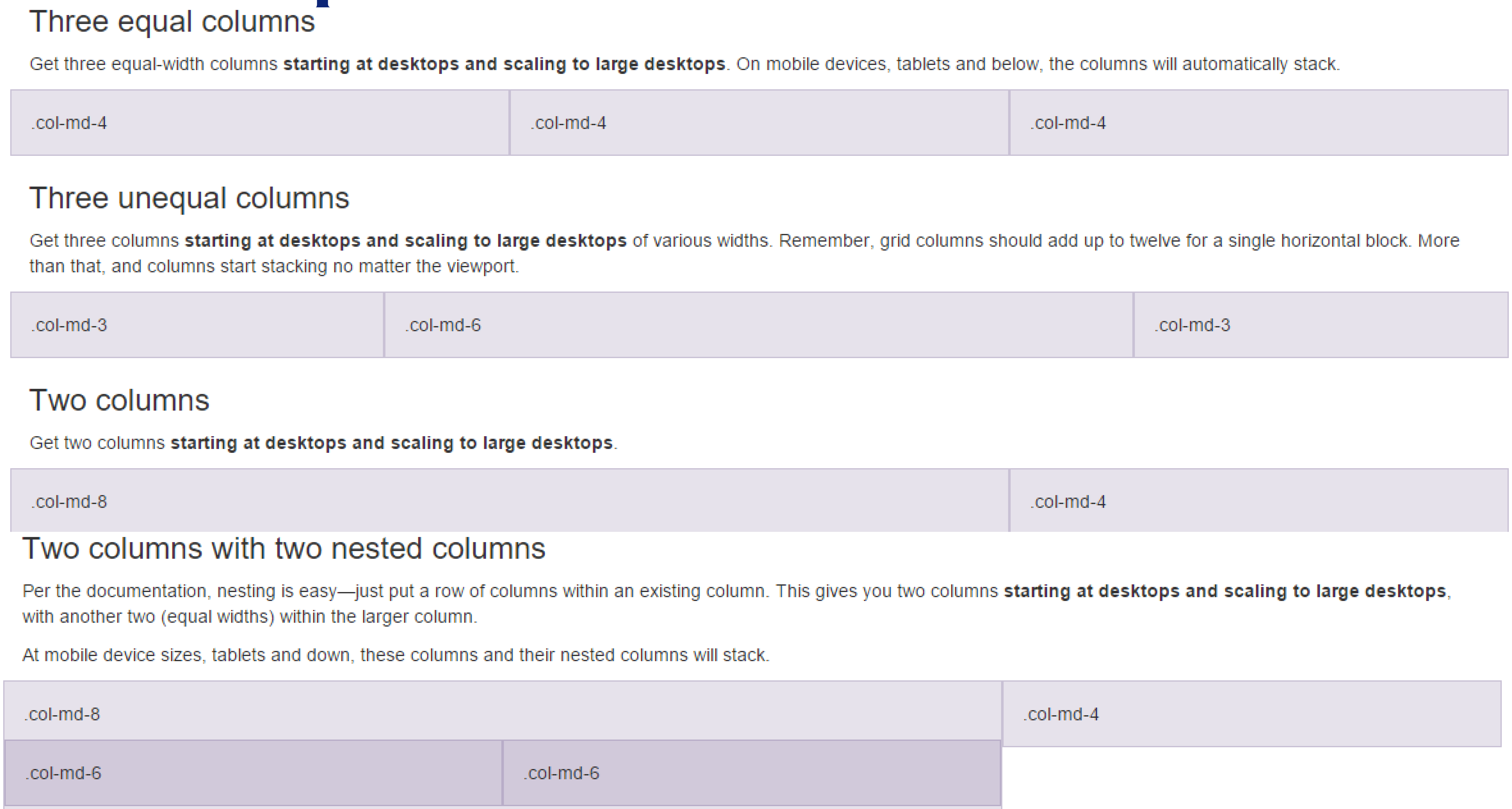
- `<div>`xx`</div>` groepeert elementen in een "box"
- `<span>`tekst`</span>` groepeert een stuk tekst

Bootstrap

- Waarom Bootstrap?
  - Kant en klare styles
  - Consistentie
  - Responsive
  - Templates

- Ideaal voor een mooi dashboard!
  - Templates beschikbaar op Google.

Bootstrap Grid



Three equal columns

Get three equal-width columns **starting at desktops and scaling to large desktops**. On mobile devices, tablets and below, the columns will automatically stack.

.col-md-4	.col-md-4	.col-md-4

Three unequal columns

Get three columns **starting at desktops and scaling to large desktops** of various widths. Remember, grid columns should add up to twelve for a single horizontal block. More than that, and columns start stacking no matter the viewport.

.col-md-3	.col-md-6	.col-md-3

Two columns

Get two columns **starting at desktops and scaling to large desktops**.

.col-md-8	.col-md-4

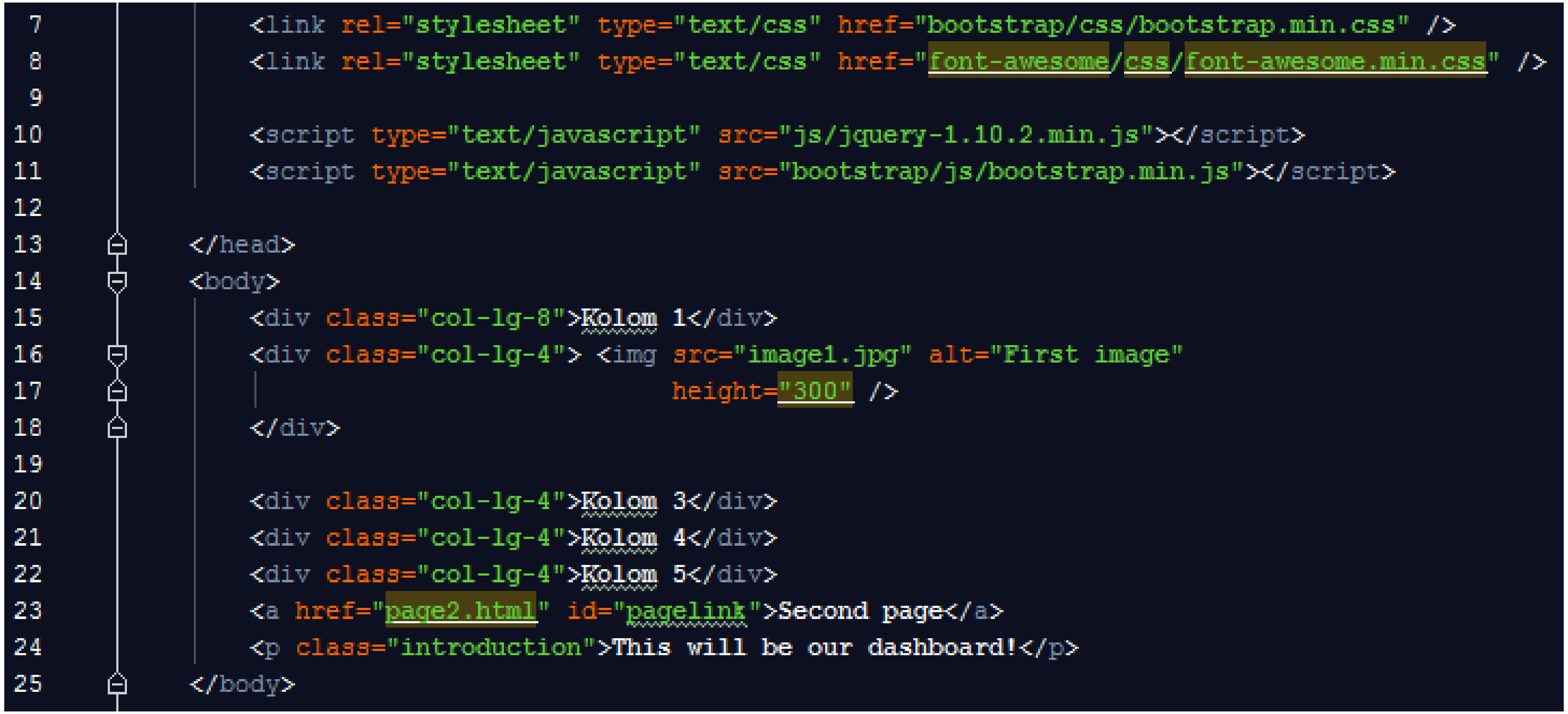
Two columns with two nested columns

Per the documentation, nesting is easy—just put a row of columns within an existing column. This gives you two columns **starting at desktops and scaling to large desktops**, with another two (equal widths) within the larger column.

At mobile device sizes, tablets and down, these columns and their nested columns will stack.

<table>
<tr><td colspan="2">.col-md-8</td><td>.col-md-4</td></tr>
<tr><td>.col-md-6</td><td>.col-md-6</td><td></td></tr>
</table>

Hoe te gebruiken?



```
    <link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="font-awesome/css/font-awesome.min.css" />

    <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
    <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>

</head>
<body>
    <div class="col-lg-8">Kolom 1</div>
    <div class="col-lg-4"> <img src="image1.jpg" alt="First image"
                                height="300" />
    </div>

    <div class="col-lg-4">Kolom 3</div>
    <div class="col-lg-4">Kolom 4</div>
    <div class="col-lg-4">Kolom 5</div>
    <a href="page2.html" id="pagelink">Second page</a>
    <p class="introduction">This will be our dashboard!</p>
</body>
```

Hoe te gebruiken?

Kolom 1

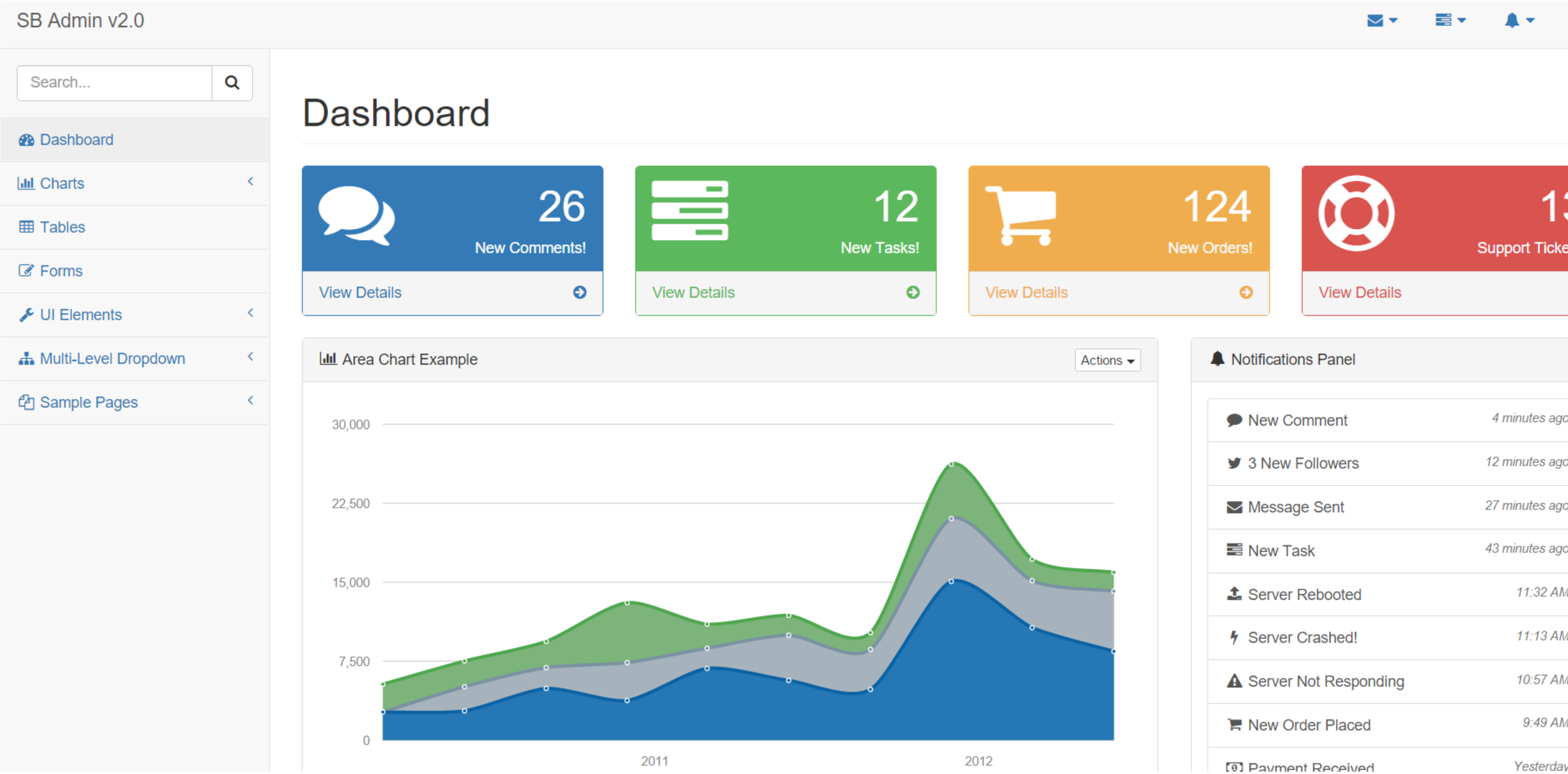
Kolom 3

Kolom 4

Kolom 5

Templates

- Meestal kant en klaar, downloaden en inladen.
- Veel voorbeelden!



Aanpassen CSS Bootstrap

- Eigen style.css later dan Bootstrap CSS implementeren, dan is style.css laatste waar naar gekeken word.

```
<link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="style.css" />
```

PHP

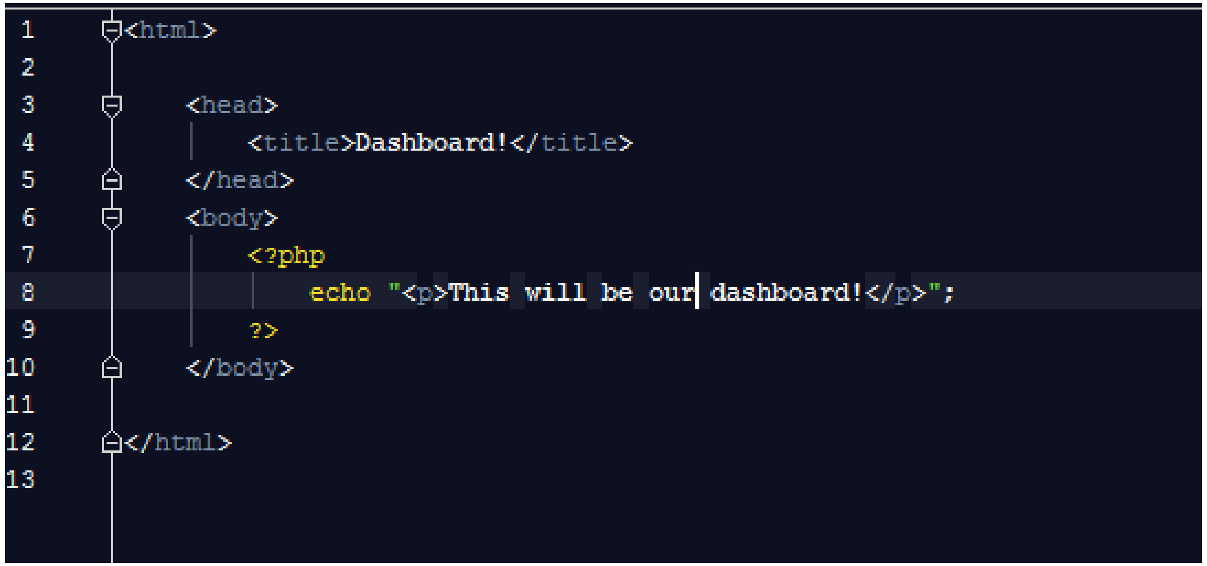
- **H**ypertext **P**reprocessor
- Programmeertaal
- Server-side

Voor de opdracht relevant vanwege:

- Communicatie met de database

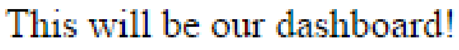
Simple PHP Page

Input



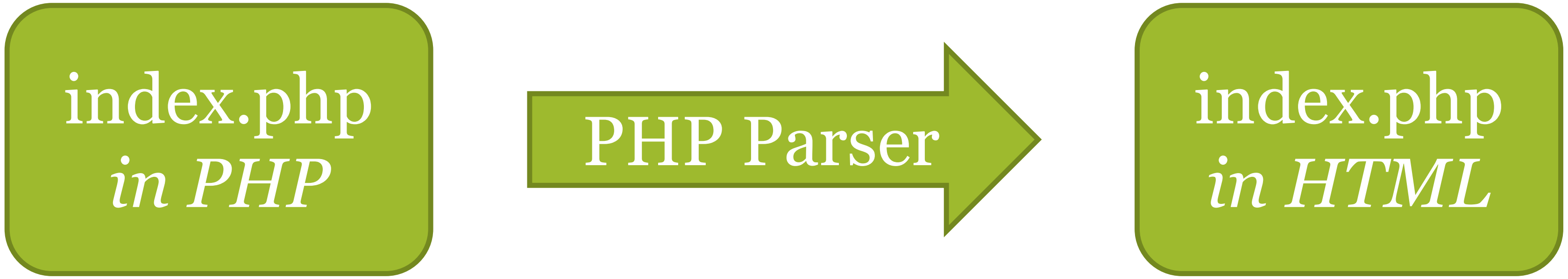
```
<html>

    <head>
        <title>Dashboard!</title>
    </head>
    <body>
        <?php
            echo "<p>This will be our dashboard!</p>";
        ?>
    </body>

</html>

```

Output



This will be our dashboard!

Hoe werkt PHP?



index.php
in PHP

PHP Parser

index.php
in HTML

```
<html>
  <body>
    <?
    echo '<p>Hoi!</p>';
    ?>
  </body>
</html>
```

```
<html>
  <body>
    <p>Hoi!</p>
  </body>
</html>
```

Variabelen

- Automatische datatypes:
  - Integers `$var1 = 4;`
  - Doubles/Floats `$var2 = 4.223161584;`
  - Characters/strings `$var3 = "8";`
- Automatische typecasting:
  `echo $var1 + $var3;` geeft (meestal) `12`
- Variabelen in strings: `$var = 'Tekst';`
  - `echo "Dit is $var";` geeft `Dit is Tekst`
  - `echo 'Dit is $var';` geeft `Dit is $var`

PHP commando’s

• Output	`echo`
• Output concatenatie	`.`
• Einde statement	`;`

• Operatoren	`+   -   /   *   %   =`
	`++   --   +=   -=`

• Logische operatoren	`AND   OR   XOR`
• Vergelijkende operatoren	`==   !=   <=  >=  <  >`
• Commentaar	`//` of `/* iets */`

PHP commando’s

- Vergelijkende statements
  - `if()`
  - `else()`
  - `switch()`
- Loops
  - `while()`
  - `for()`
  - `foreach()`

Arrays

- Traditioneel:

  ```
  $things = array(13.37, "Volvo", 4);
  ```

  Nu bestaan `$things[0]`, `$things[1]` & `$things[2]`, maar zomaar `$things[19493] = 4;` mag ook

- Associatief:

  ```
  $things = array("Audi"=>2.8, "Opel"=>1.3);
  ```

  `$things["Audi"]` bevat `2.8`, etc.

- Doorlopen:

  ```
  foreach($array as $key => $value)
    echo $key . " bevat " . $value . "\n";
  ```

- Multidimensionaal kan ook

Functies

- Geen type bij functie-declaratie
- Return-statements zijn optioneel
- Parameters zijn optioneel
- Scope is uitsluitend lokaal
- Gebruik van globals moet expliciet worden aangegeven

-
  ```
  function doSomething($para = 10) {
      global $altijd;
      $para *= 2;
      $iets = $altijd + 4 + $para;
      return $iets;
  }
  ```
- Mag aangeroepen worden met `doSomething();` of `doSomething(20);`

$_GET[“”]

- Stel je hebt een pagina http://liacs.leidenuniv.nl/index.php?message=Bonjour&name=Thomas

- Nu kan je met $_GET[“message”] de waarde bonjour verkrijgen. Hiermee kan je dus communiceren via de link, en bijvoorbeeld gerichter zoeken!

index.php?message=bonjour

```
<html>
  <body>
    <?
    echo '<p>’ .
$_GET[‘message’] . ‘ ’
. $_GET[‘name’] .
‘</p>';
    ?>
  </body>
</html>
```

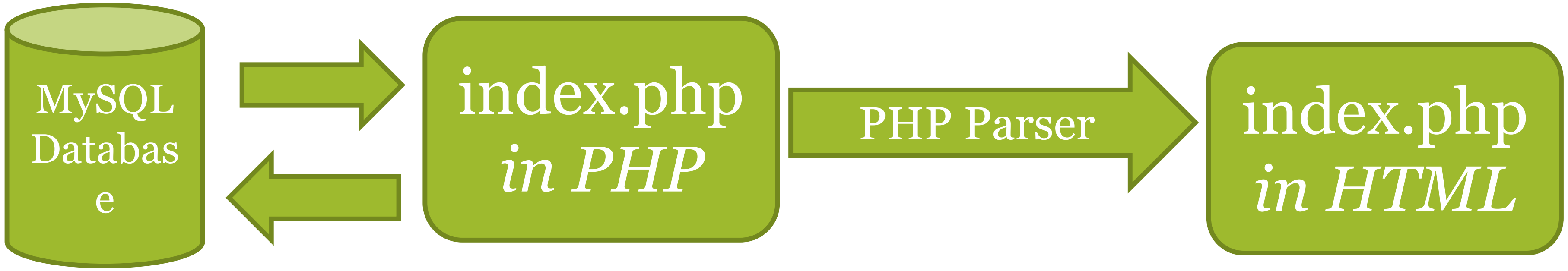
```
<html>
  <body>
 <p>Bonjour
Thomas!</p>
  </body>
</html>
```

Pauze?

MySQL

- SQL = Structured Query Language
- MySQL is een gratis variant
- **Database** bevat tabellen
- **Tabel** bevat rijen en kolommen
- **Kolommen** hebben een datatype
- **Rijen** zijn de instanties
- **Query**: verzoek aan database

MySQL



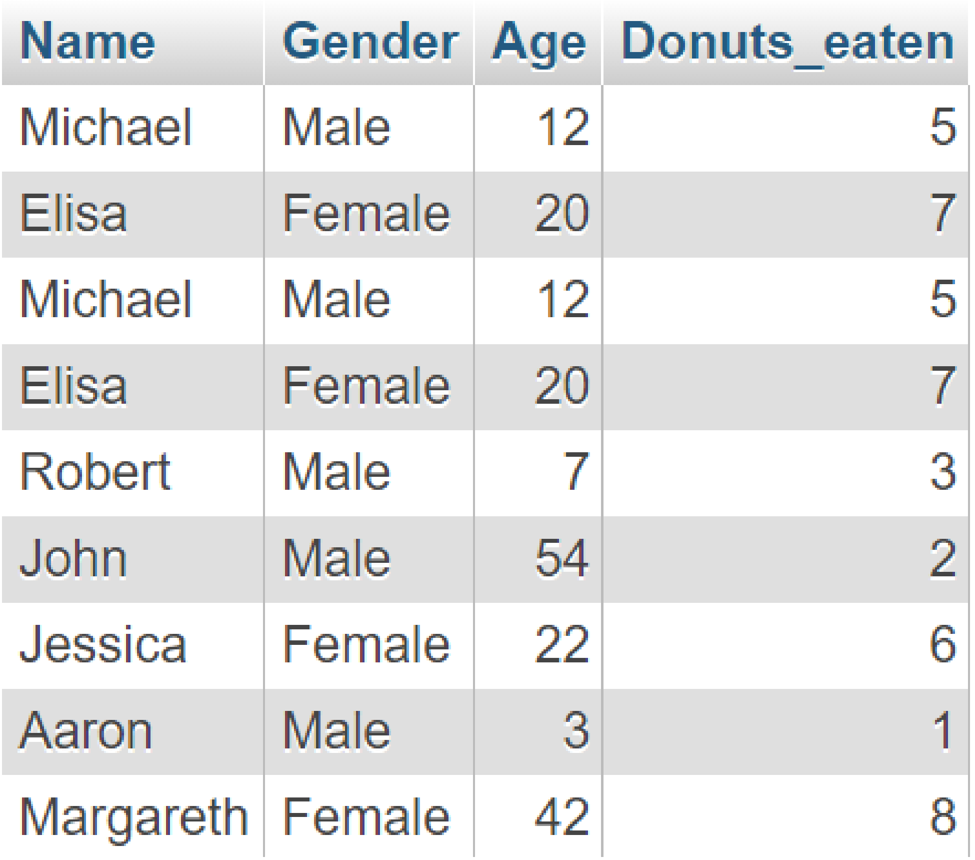
```
<html>
<body>
<?
$sql = "SELECT welcome FROM
texttable";
$res = mysql_query($sql);
$rec = mysql_fetch_array($res);
echo '<p>' . $rec['welcome'] . '</p>';
?>
</body>
</html>
```

```
<html>
  <body>
    <p>Hoi!</p>
  </body>
</html>
```

MySQL verbinding

```
<?php
 $servername = “mysql.liacs.nl";
 $username = "username";
 $password = "password";
 $dbname = “DB";

 // Maken van verbinding
 $conn = new mysqli($servername, $username, $password, $dbname);
 // Check connection
 if ($conn->connect_error) {
     die("Connection failed: " . $conn->connect_error);
 }

/* HIER ROEP JE QUERIES AAN E.D. */

$conn->close()// Afsluiten verbinding

?>
```

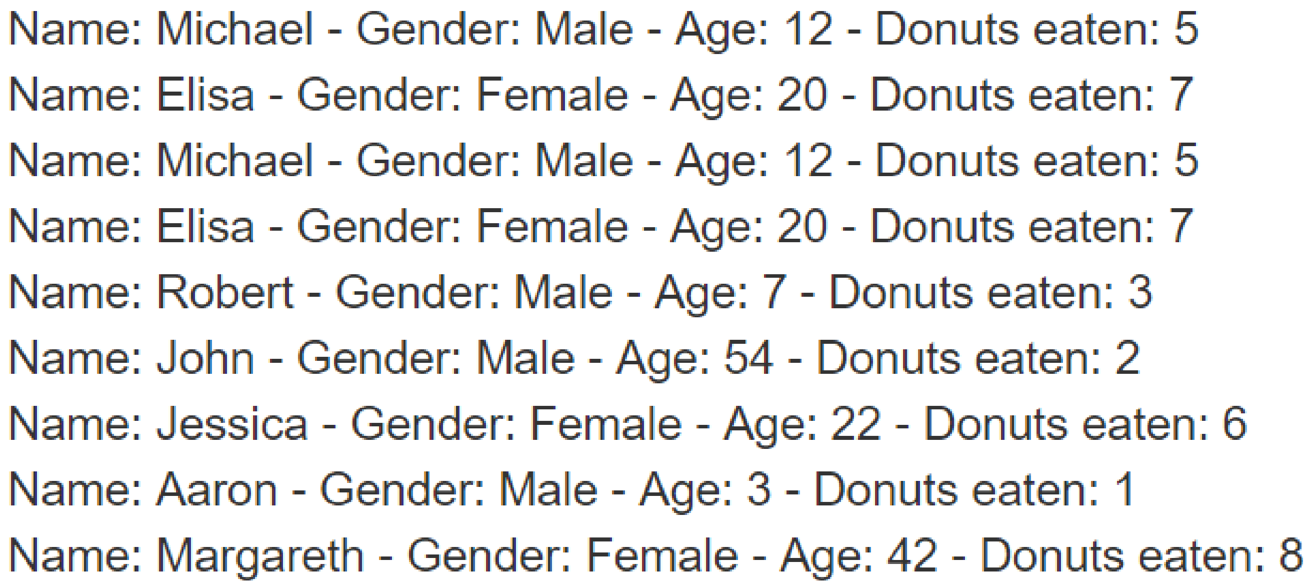
Tabel: Donuts



Name	Gender	Age	Donuts_eaten
Michael	Male	12	5
Elisa	Female	20	7
Michael	Male	12	5
Elisa	Female	20	7
Robert	Male	7	3
John	Male	54	2
Jessica	Female	22	6
Aaron	Male	3	1
Margareth	Female	42	8

Queries in PHP

```
$sql = "SELECT * FROM bipm";

$result = $conn->query($sql);


if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "Name: " . $row["Name"] . " –

          Gender: " .          $row["Gender"] . " - Age: " .
          $row["Age"] . " -          Donuts eaten: " .
          $row["Donuts_eaten"] . “

        <br />";
    }

} else {
    echo “0 results";

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



```
Name: Michael - Gender: Male - Age: 12 - Donuts eaten: 5
Name: Elisa - Gender: Female - Age: 20 - Donuts eaten: 7
Name: Michael - Gender: Male - Age: 12 - Donuts eaten: 5
Name: Elisa - Gender: Female - Age: 20 - Donuts eaten: 7
Name: Robert - Gender: Male - Age: 7 - Donuts eaten: 3
Name: John - Gender: Male - Age: 54 - Donuts eaten: 2
Name: Jessica - Gender: Female - Age: 22 - Donuts eaten: 6
Name: Aaron - Gender: Male - Age: 3 - Donuts eaten: 1
Name: Margareth - Gender: Female - Age: 42 - Donuts eaten: 8
```

jQuery (Javascript library)

- Snel, klein en bevat veel features
- Maakt veel functies stuk simpeler

- Gebruikt Javascript

Variabelen

- Datatypes:
  - Integers `var number = 4;`
  - Doubles/Floats `var number = 4.223161584;`
  - Characters/strings `var text = “hello";`
- Variabelen in strings: `var text = 'Tekst';`
  - `echo "Dit is “ + $var;` geeft `Dit is Tekst`

Javascript commando’s

Output	console.log
Output concatenatie	+
Einde statement	;
Operatoren	+ - / * % = ++ -- += -=
Logische operatoren	AND OR XOR
Vergelijkende operatoren	== != <= >= < >
Commentaar	// of /* iets */
Console zien	F12

Aanroepen

- Download jQuery: https://jquery.com/download/ en zet in eigen map

- In je <head>:
  - <script type="**text/javascript**" src="**js/jquery-x.y.z.min.js**"></script>
  - X, y, z zijn afhankelijk van de versie. Bijv. 10.2.1.

- Nu kan je alle functies van jQuery gebruiken!

jQuery beginnen

- Nieuwe javascript file aanmaken met javascript erin:

```
<script type="text/javascript" src="js/eigenpagina.js"></script>
```

```
$(document).ready(function() {
  console.log('Ready!');
});
```

Maar wat kan je allemaal doen?

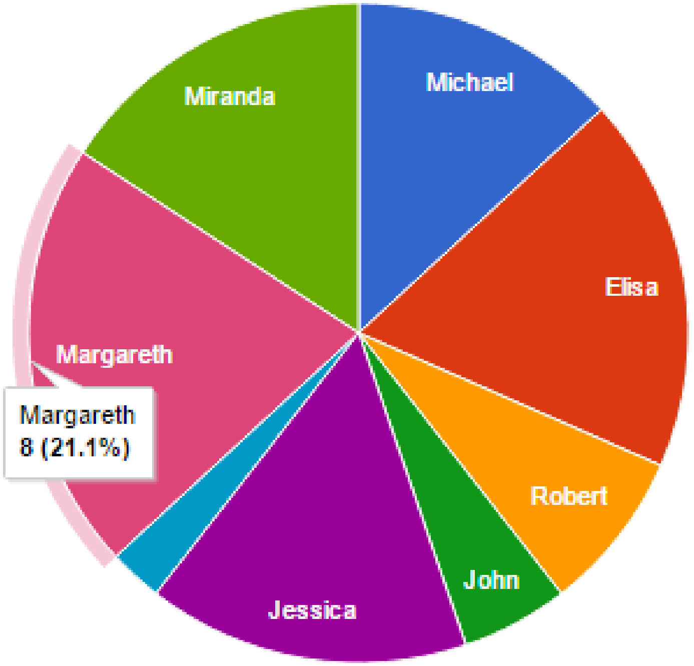
Demo’s

- https://jqueryui.com/demos/

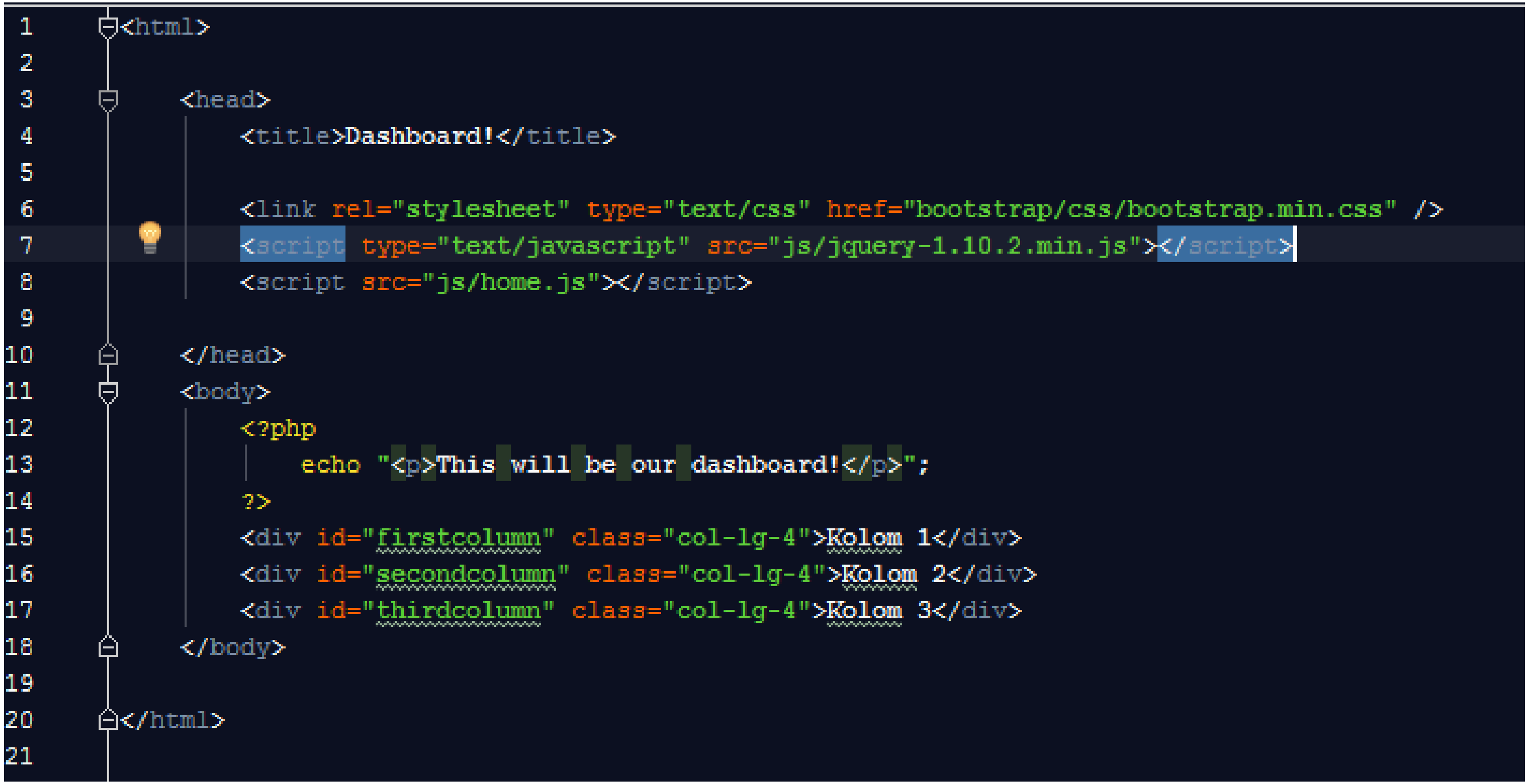
Visualisatie

- Verschillende mogelijkheden:
  - Chart.js
  - Google Charts
  - Maar nog heel veel **meer**

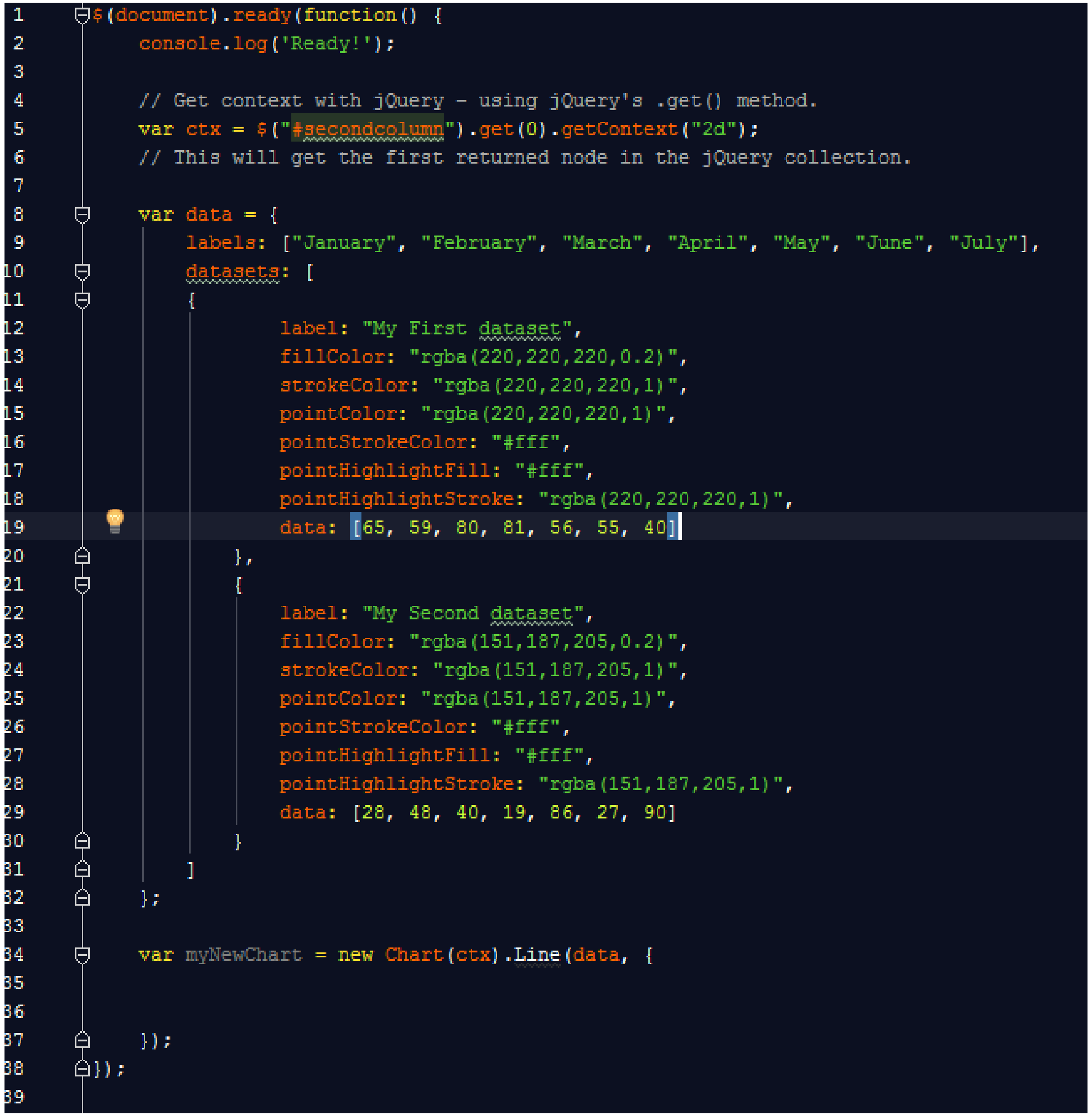
- Hoe goed gebruiken?
  - Gebruik de **demo's en voorbeelden**
  - Lees documentatie!



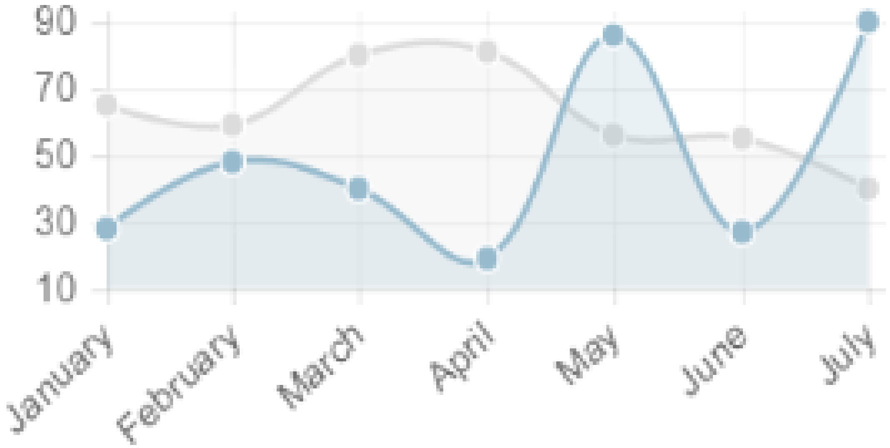
Voorbeeld

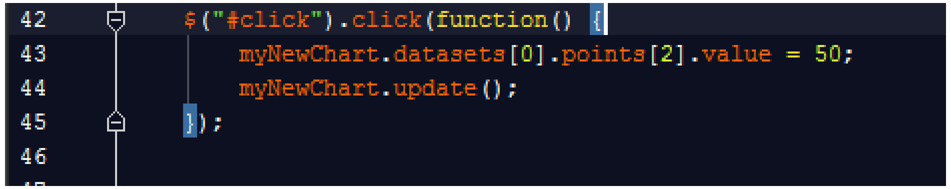


```
<html>

    <head>
        <title>Dashboard!</title>

        <link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css" />
        <script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
        <script src="js/home.js"></script>

    </head>
    <body>
        <?php
            echo "<p>This will be our dashboard!</p>";
        ?>
        <div id="firstcolumn" class="col-lg-4">Kolom 1</div>
        <div id="secondcolumn" class="col-lg-4">Kolom 2</div>
        <div id="thirdcolumn" class="col-lg-4">Kolom 3</div>
    </body>

</html>

```

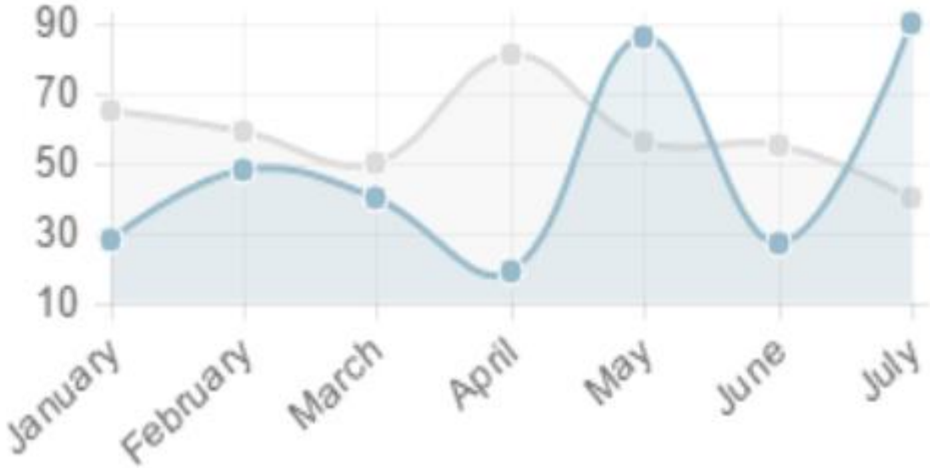


```
$(document).ready(function() {
    console.log('Ready!');

    // Get context with jQuery - using jQuery's .get() method.
    var ctx = $("#secondcolumn").get(0).getContext("2d");
    // This will get the first returned node in the jQuery collection.

    var data = {
        labels: ["January", "February", "March", "April", "May", "June", "July"],
        datasets: [
        {
                label: "My First dataset",
                fillColor: "rgba(220,220,220,0.2)",
                strokeColor: "rgba(220,220,220,1)",
                pointColor: "rgba(220,220,220,1)",
                pointStrokeColor: "#fff",
                pointHighlightFill: "#fff",
                pointHighlightStroke: "rgba(220,220,220,1)",
                data: [65, 59, 80, 81, 56, 55, 40]
            },
            {
                label: "My Second dataset",
                fillColor: "rgba(151,187,205,0.2)",
                strokeColor: "rgba(151,187,205,1)",
                pointColor: "rgba(151,187,205,1)",
                pointStrokeColor: "#fff",
                pointHighlightFill: "#fff",
                pointHighlightStroke: "rgba(151,187,205,1)",
                data: [28, 48, 40, 19, 86, 27, 90]
            }
        ]
    };

    var myNewChart = new Chart(ctx).Line(data, {


    });
});
```



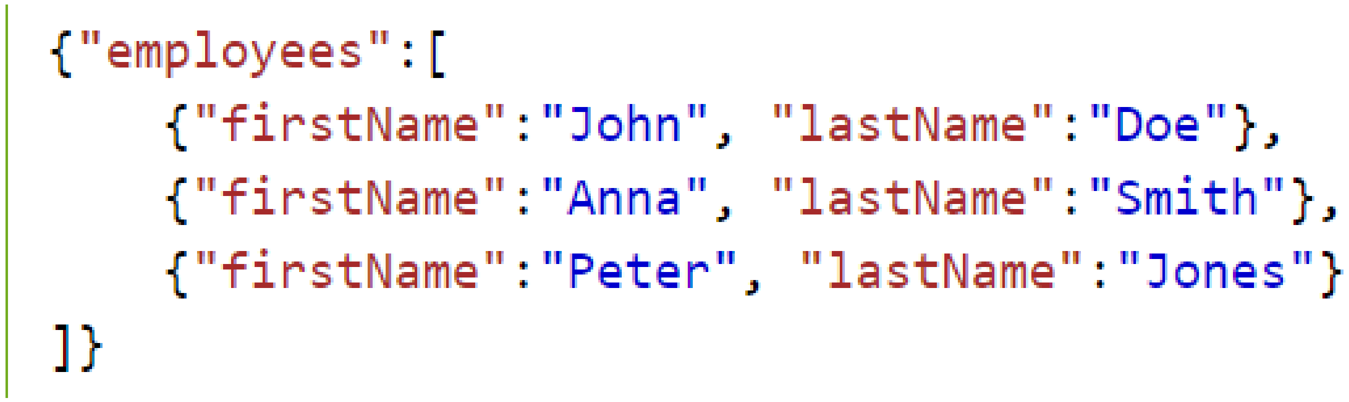


```
    $("#click").click(function() {
        myNewChart.datasets[0].points[2].value = 50;
        myNewChart.update();
    });
```



Maar hoe data inladen met PHP en jQuery?

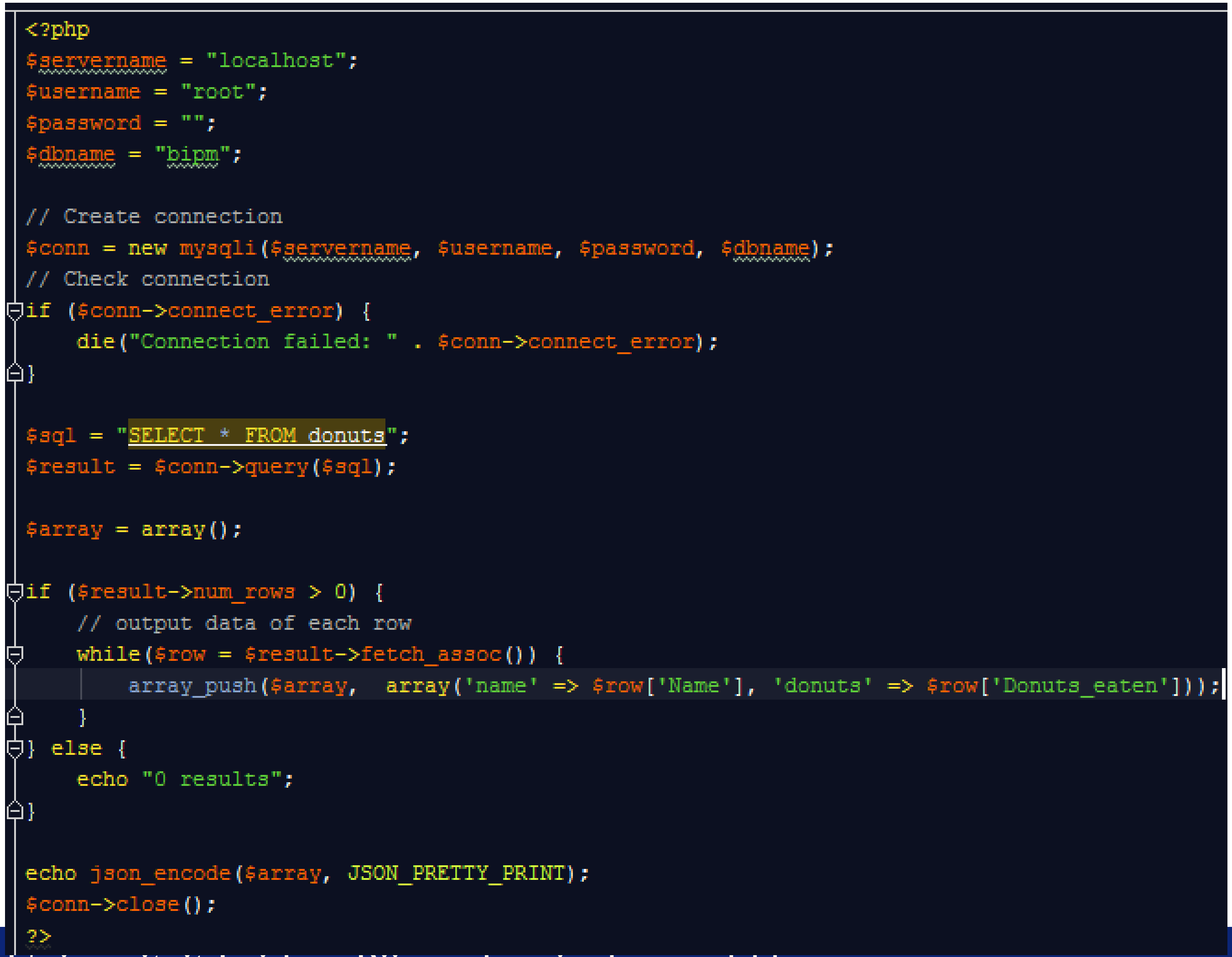
- JSON: **J**ava**S**cript **O**bject **N**otation



```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

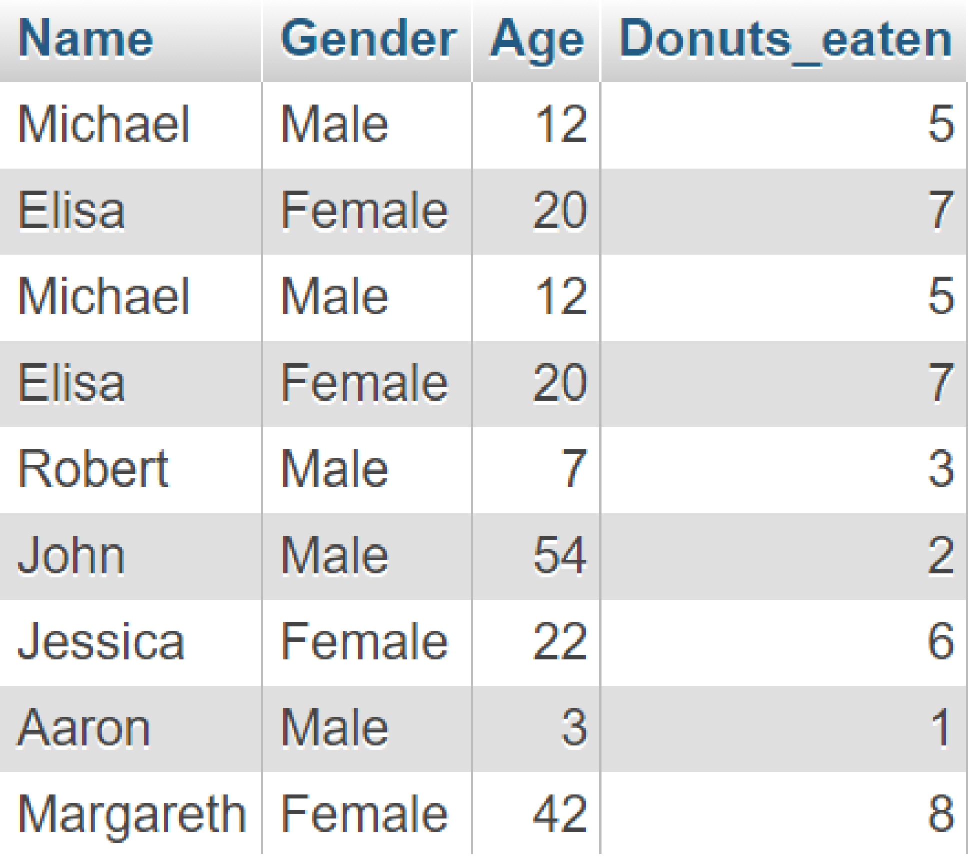
Hoe?

getDonuts.php



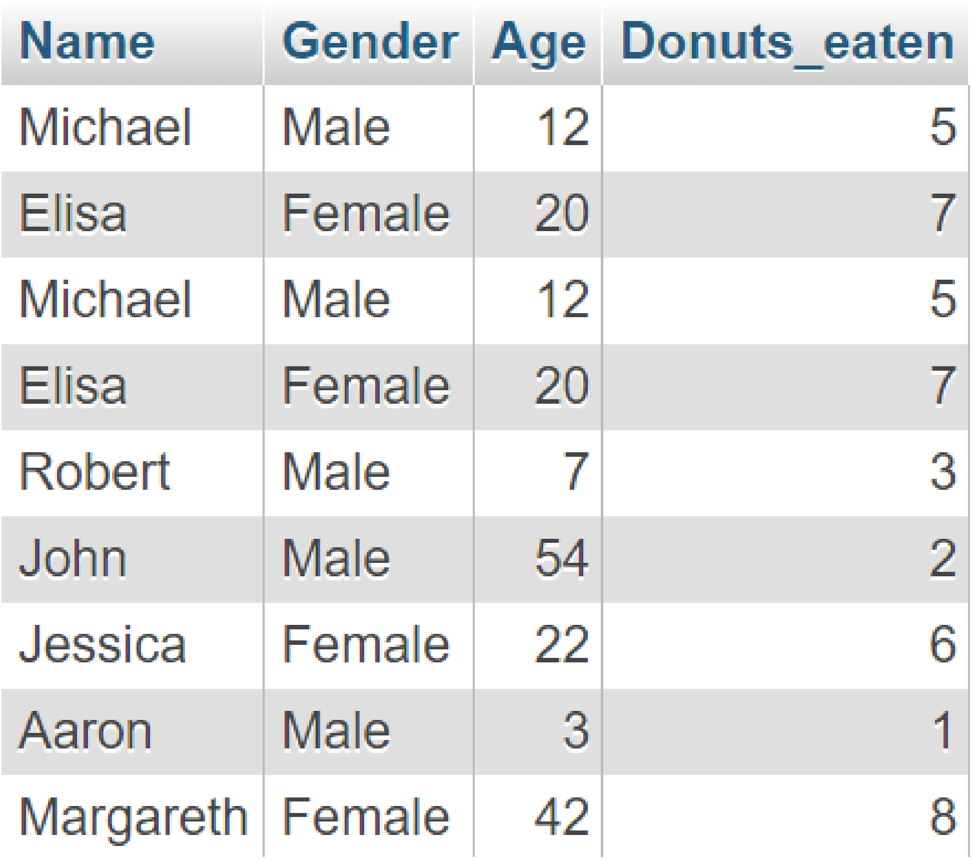
```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "bipm";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM donuts";
$result = $conn->query($sql);

$array = array();

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        array_push($array,  array('name' => $row['Name'], 'donuts' => $row['Donuts_eaten']));
    }
} else {
    echo "0 results";
}

echo json_encode($array, JSON_PRETTY_PRINT);
$conn->close();
?>
```

Database



Name	Gender	Age	Donuts_eaten
Michael	Male	12	5
Elisa	Female	20	7
Michael	Male	12	5
Elisa	Female	20	7
Robert	Male	7	3
John	Male	54	2
Jessica	Female	22	6
Aaron	Male	3	1
Margareth	Female	42	8

SQL to JSON



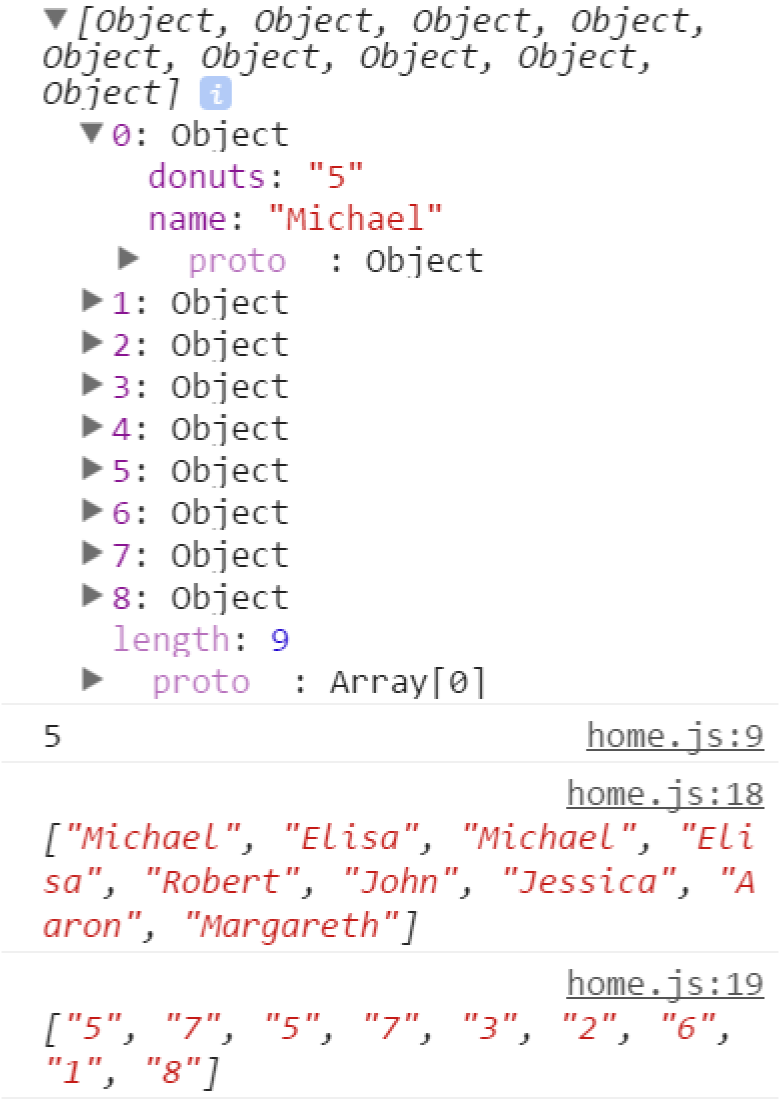
Name	Gender	Age	Donuts_eaten
Michael	Male	12	5
Elisa	Female	20	7
Michael	Male	12	5
Elisa	Female	20	7
Robert	Male	7	3
John	Male	54	2
Jessica	Female	22	6
Aaron	Male	3	1
Margareth	Female	42	8

getDonuts.php

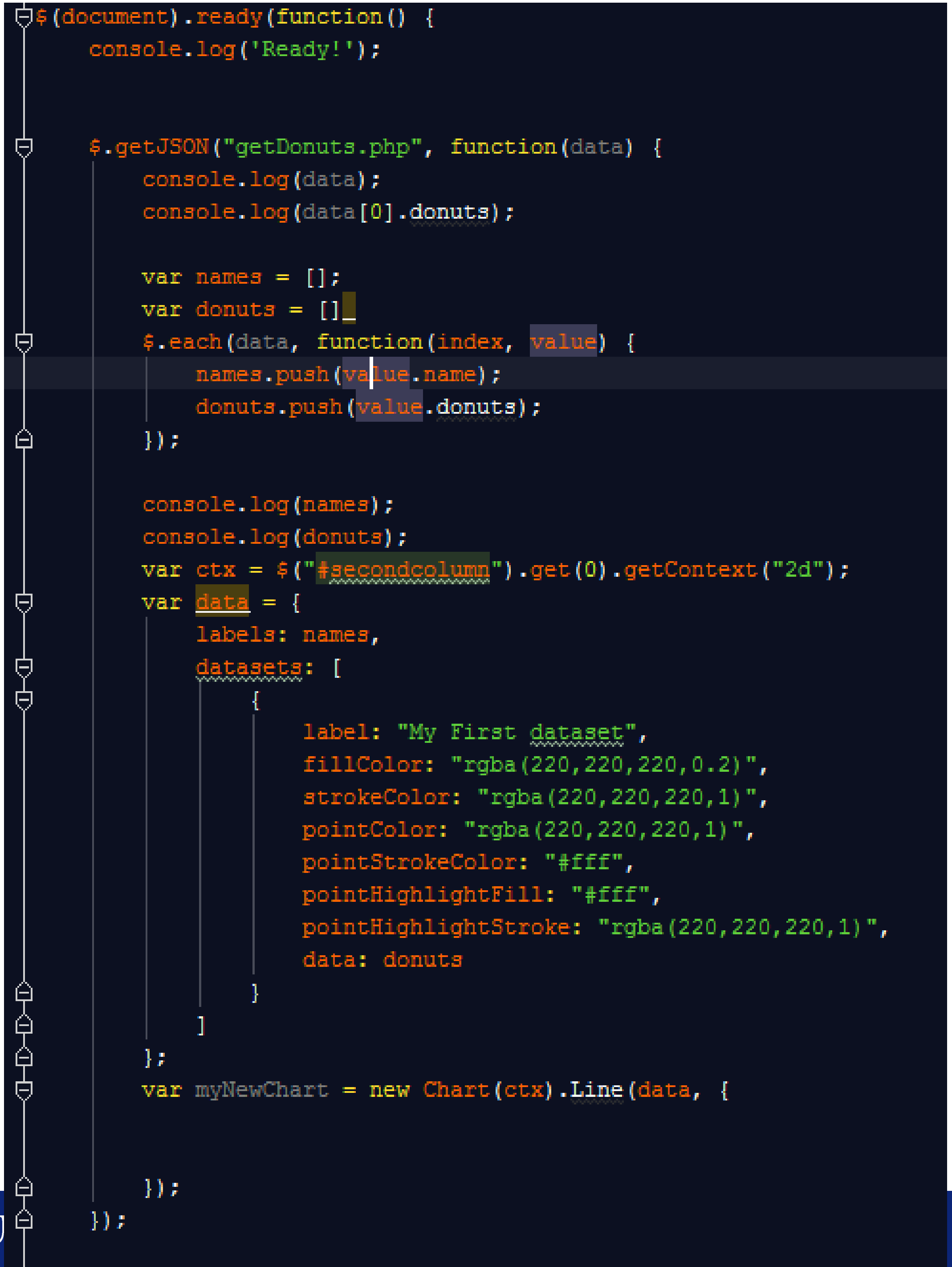
```
[
		{ "name": "Michael", "donuts": "5" },
		{ "name": "Elisa", "donuts": "7" },
		{ "name": "Michael", "donuts": "5" },
		{ "name": "Elisa", "donuts": "7" },
		{ "name": "Robert", "donuts": "3" },
		{ "name": "John", "donuts": "2" },
		{ "name": "Jessica", "donuts": "6" },
		{ "name": "Aaron", "donuts": "1" },
		{ "name": "Margareth", "donuts": "8" }
]
```

Voorbeeld

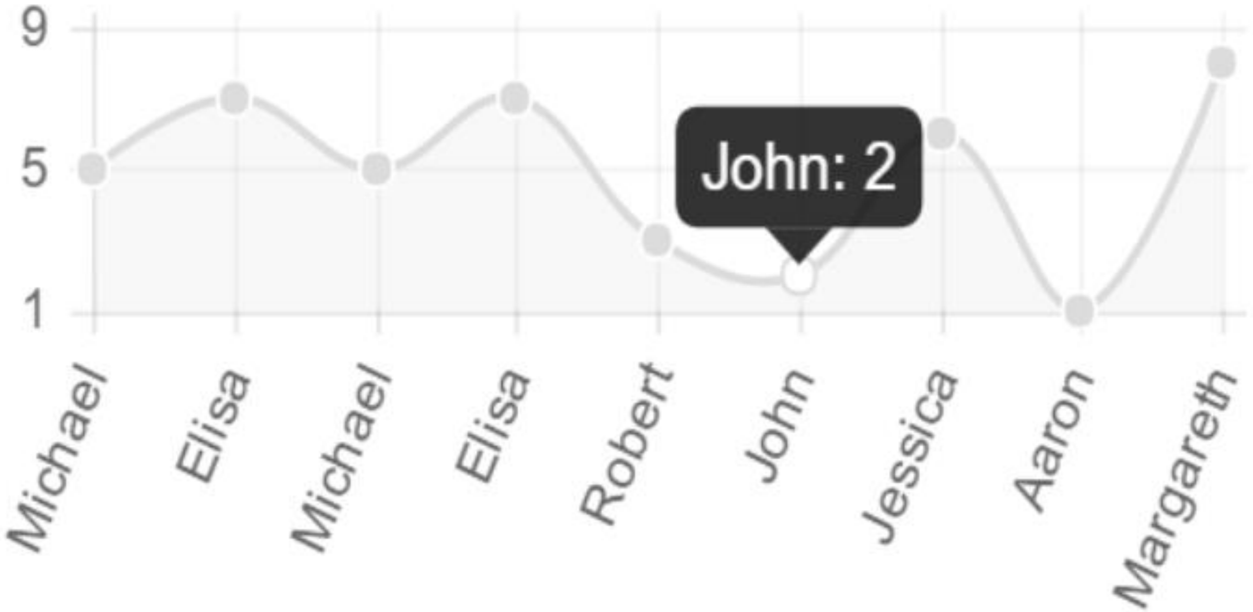
```
$.getJSON(“getDonuts.php", function(data)
    console.log(data);
    console.log(data[0].donuts);


    var names = [];

    var donuts = [];
    $.each(data, function(index, value) {
        names.push(value.name);
        donuts.push(value.donuts);
    });

    console.log(names);

    console.log(donuts);

});
```



```
▼[Object, Object, Object, Object,
Object, Object, Object, Object,
Object] i
  ▼0: Object
     donuts: "5"
     name: "Michael"
   ▶ proto  : Object
  ▶1: Object
  ▶2: Object
  ▶3: Object
  ▶4: Object
  ▶5: Object
  ▶6: Object
  ▶7: Object
  ▶8: Object
   length: 9
  ▶ proto  : Array[0]
5                                  home.js:9
                                  home.js:18
["Michael", "Elisa", "Michael", "Eli
sa", "Robert", "John", "Jessica", "A
aron", "Margareth"]
                                  home.js:19
["5", "7", "5", "7", "3", "2", "6",
"1", "8"]
```



```
$(document).ready(function() {
    console.log('Ready!');


    $.getJSON("getDonuts.php", function(data) {
        console.log(data);
        console.log(data[0].donuts);

        var names = [];
        var donuts = []
        $.each(data, function(index, value) {
            names.push(value.name);
            donuts.push(value.donuts);
        });

        console.log(names);
        console.log(donuts);
        var ctx = $("#secondcolumn").get(0).getContext("2d");
        var data = {
            labels: names,
            datasets: [
                {
                    label: "My First dataset",
                    fillColor: "rgba(220,220,220,0.2)",
                    strokeColor: "rgba(220,220,220,1)",
                    pointColor: "rgba(220,220,220,1)",
                    pointStrokeColor: "#fff",
                    pointHighlightFill: "#fff",
                    pointHighlightStroke: "rgba(220,220,220,1)",
                    data: donuts
                }
            ]
        };
        var myNewChart = new Chart(ctx).Line(data, {


        });
    });
```



Tips

- Gebruik Bootstrap
  - Bootstrap heeft niet alleen makkelijke grids, ook hebben zij makkelijke andere tools (e.g. ’date picker’) http://getbootstrap.com/components/
- Gebruik jQuery
  - Je kan je website interactief maken door te meten of er geclickt wordt op een bepaalde knop.
  - Je kan data vullen door JSON te gebruiken
- Gebruik $_GET[] van PHP
  - Je kan hiermee misschien wel specifieker zoeken door variabelen mee te geven.
- Zoek verder dan alleen chart.js, Google Charts, en D3.js!

Handige links

- **HTML/CSS:** http://www.w3schools.com/
- **Bootstrap:** http://getbootstrap.com/getting-started/
- **PHP:** http://www.w3schools.com/php/php_mysql_select.asp
- **jQuery:** http://www.w3schools.com/jquery/jquery_examples.asp
- **getJSON:** http://api.jquery.com/jquery.getjson/

Google is your best friend

Wolkom! Google Translate now speaks Frisian

Webtechnology crash course

Universiteit
Leiden