

# PHP

---

Forelesning 12, Webprosjekt H2013

Alfred Bratterud

# Plan for dagen

- \* “Table” som beholder proporsjoner!
- \* Grupper!
- \* PHP Basics
- \* I morgen: Oblig - del 2!
- \* Neste gang: Fyrverkeri!

# Kvadratisk Table med kun ordinær CSS

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett
- \* Trikset: all padding i prosent, er relativ til foreldrelementets bredde. (Rart!)

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett
- \* Trikset: all padding i prosent, er relativ til foreldrelementets bredde. (Rart!)
- \* Man kan derfor sette bredde til å være feks. 90%, høyde 0px, og padding-bottom til...?

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett
- \* Trikset: all padding i prosent, er relativ til foreldrelementets bredde. (Rart!)
- \* Man kan derfor sette bredde til å være feks. 90%, høyde 0px, og padding-bottom til...?
  - \* 90% det og!

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett
- \* Trikset: all padding i prosent, er relativ til foreldrelementets bredde. (Rart!)
- \* Man kan derfor sette bredde til å være feks. 90%, høyde 0px, og padding-bottom til...?
  - \* 90% det og!
  - \* Eller - en hvilken som helst proporsjon

# Kvadratisk Table med kun ordinær CSS

- \* Takk til Einar Belck-Olsen som fant løsning på nett
- \* Trikset: all padding i prosent, er relativ til foreldrelementets bredde. (Rart!)
- \* Man kan derfor sette bredde til å være feks. 90%, høyde 0px, og padding-bottom til...?
  - \* 90% det og!
  - \* Eller - en hvilken som helst proporsjon
  - \* Men, andre vei går ikke (Eller?)

# Gruppearbeid!

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser
- \* I hjemmelekse, skriv ned:

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser
- \* I hjemmelekse, skriv ned:
  - \* Hva slags type nettside du helst vil lage (Spill, avis, blogg ...)

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser
- \* I hjemmelekse, skriv ned:
  - \* Hva slags type nettside du helst vil lage (Spill, avis, blogg ...)
  - \* Tre tematiske stikkord (SciFi, Big Bang Theory, Trening, Mat)

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser
- \* I hjemmelekse, skriv ned:
  - \* Hva slags type nettside du helst vil lage (Spill, avis, blogg ...)
  - \* Tre tematiske stikkord (SciFi, Big Bang Theory, Trening, Mat)
  - \* Hvilke teknologier vil du vektlegge? HTML/CSS, Javascript, JQuery, Ajax, Php?

# Gruppearbeid!

- \* Gruppene blir satt sammen basert på interesser
- \* I hjemmelekse, skriv ned:
  - \* Hva slags type nettside du helst vil lage (Spill, avis, blogg ...)
  - \* Tre tematiske stikkord (SciFi, Big Bang Theory, Trening, Mat)
  - \* Hvilke teknologier vil du vektlegge? HTML/CSS, Javascript, JQuery, Ajax, Php?
  - \* Følg med på github!

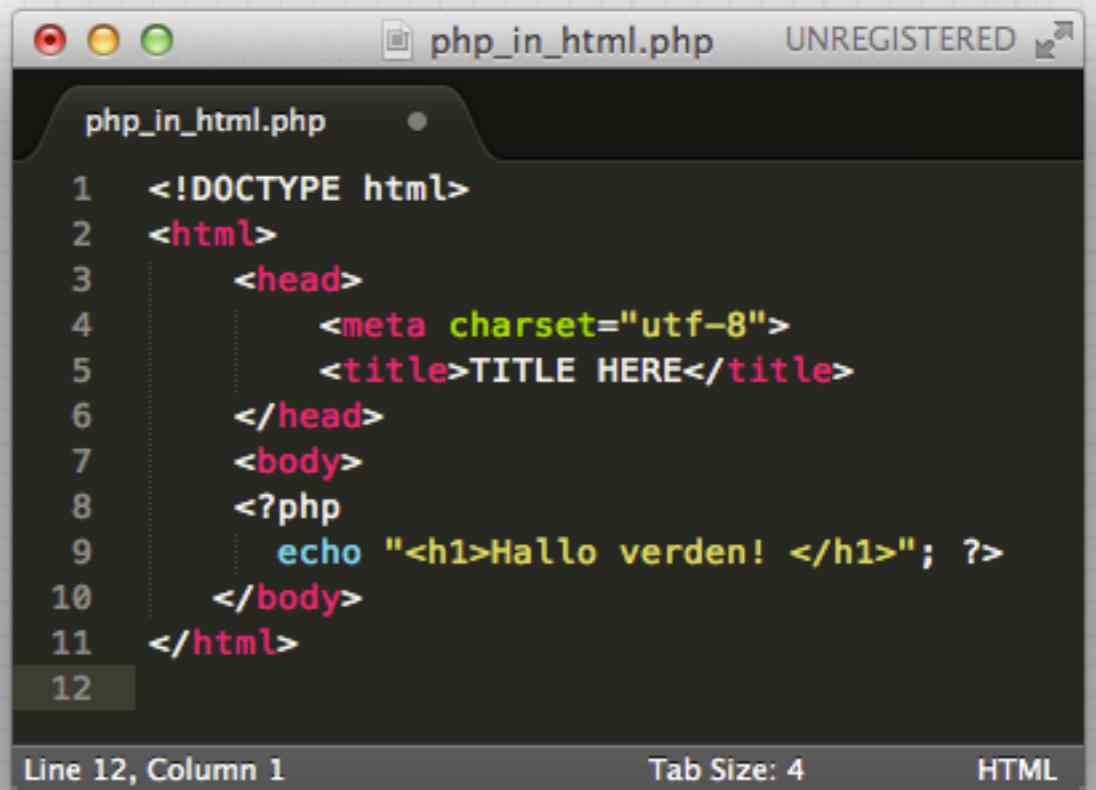
# Demo

---

`maintaining_aspect_ratio.html`

# PHP

# “Hallo Verden”



A screenshot of a code editor window titled "php\_in\_html.php". The code editor has a dark theme and shows the following PHP code:

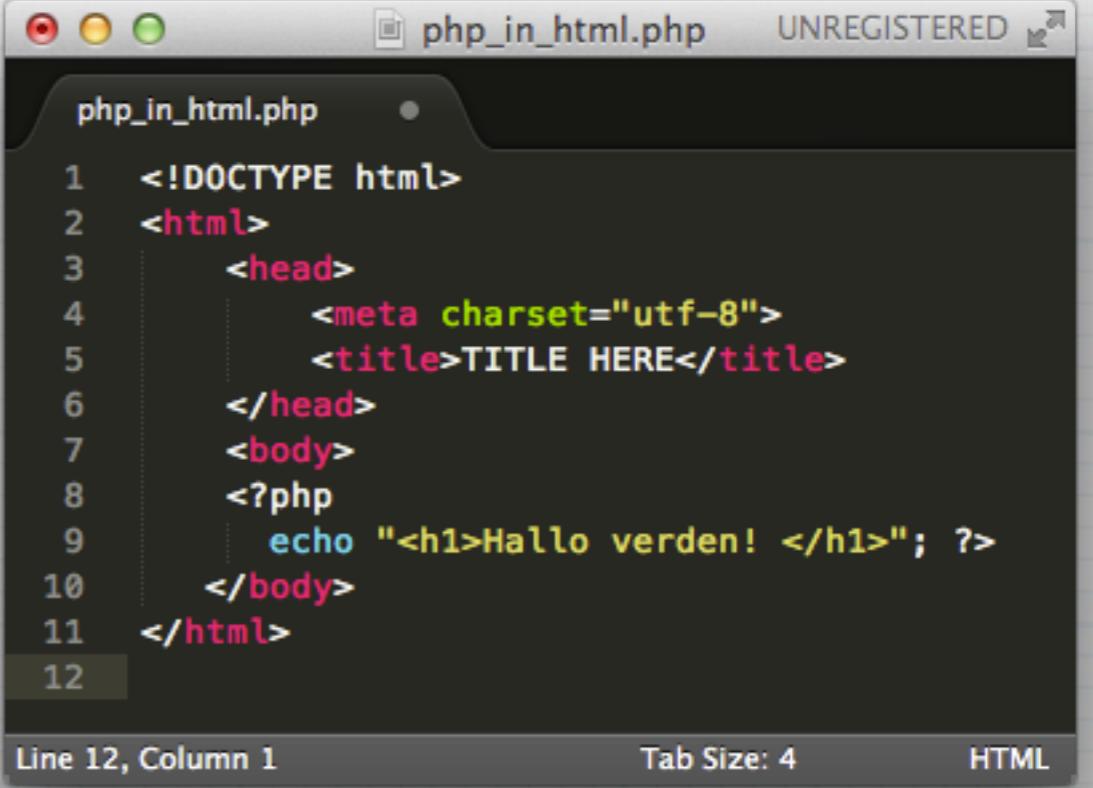
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>TITLE HERE</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1>Hallo verden! </h1>"; ?>
10    </body>
11  </html>
12
```

The status bar at the bottom of the editor window displays "Line 12, Column 1", "Tab Size: 4", and "HTML".

# PHP

# “Hallo Verden”

- \* PHP er i likhet med JS et “interpretert” “scriptspråk”



A screenshot of a code editor window titled "php\_in\_html.php". The code editor shows the following PHP script:

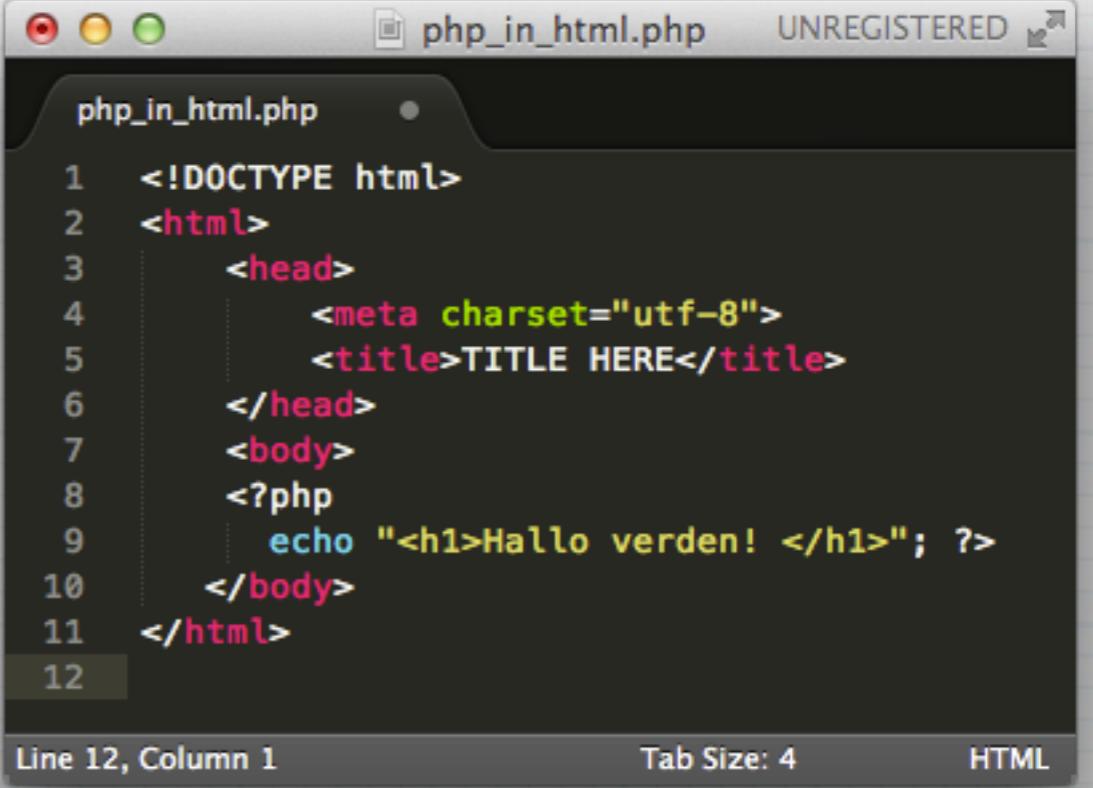
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>TITLE HERE</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1>Hallo verden! </h1>"; ?>
10    </body>
11  </html>
12
```

The code editor interface includes a toolbar at the top with icons for file operations, and status bars at the bottom indicating "Line 12, Column 1", "Tab Size: 4", and "HTML".

# PHP

# “Hallo Verden”

- \* PHP er i likhet med JS et “interpretert” “scriptspråk”
- \* Kjører på webseveren.



A screenshot of a code editor window titled "php\_in\_html.php". The code editor displays the following PHP script:

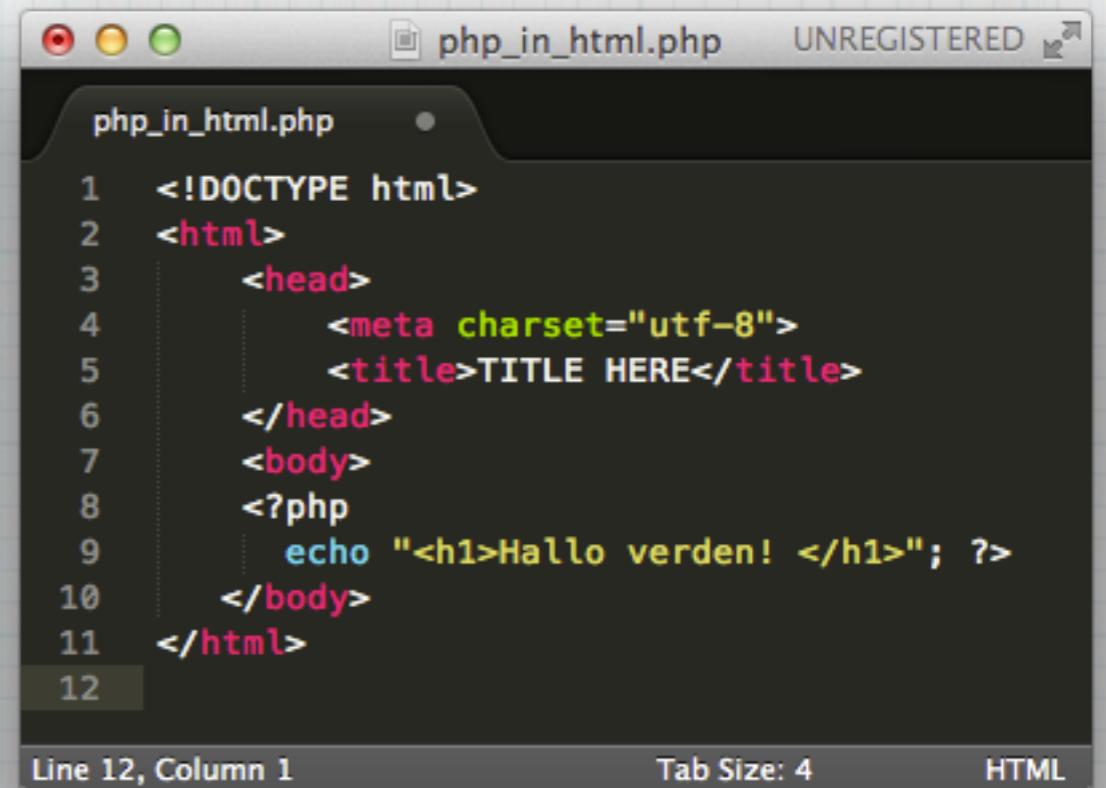
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>TITLE HERE</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1>Hallo verden! </h1>"; ?>
10    </body>
11  </html>
12
```

The code editor interface includes a toolbar at the top with icons for file operations, and status bars at the bottom indicating "Line 12, Column 1", "Tab Size: 4", and "HTML".

# PHP

# “Hallo Verden”

- \* PHP er i likhet med JS et “interpretert” “scriptspråk”
- \* Kjører på webseveren.
- \* Produserer (hvis man vil) HTML



A screenshot of a code editor window titled "php\_in\_html.php". The code is a combination of HTML and PHP:

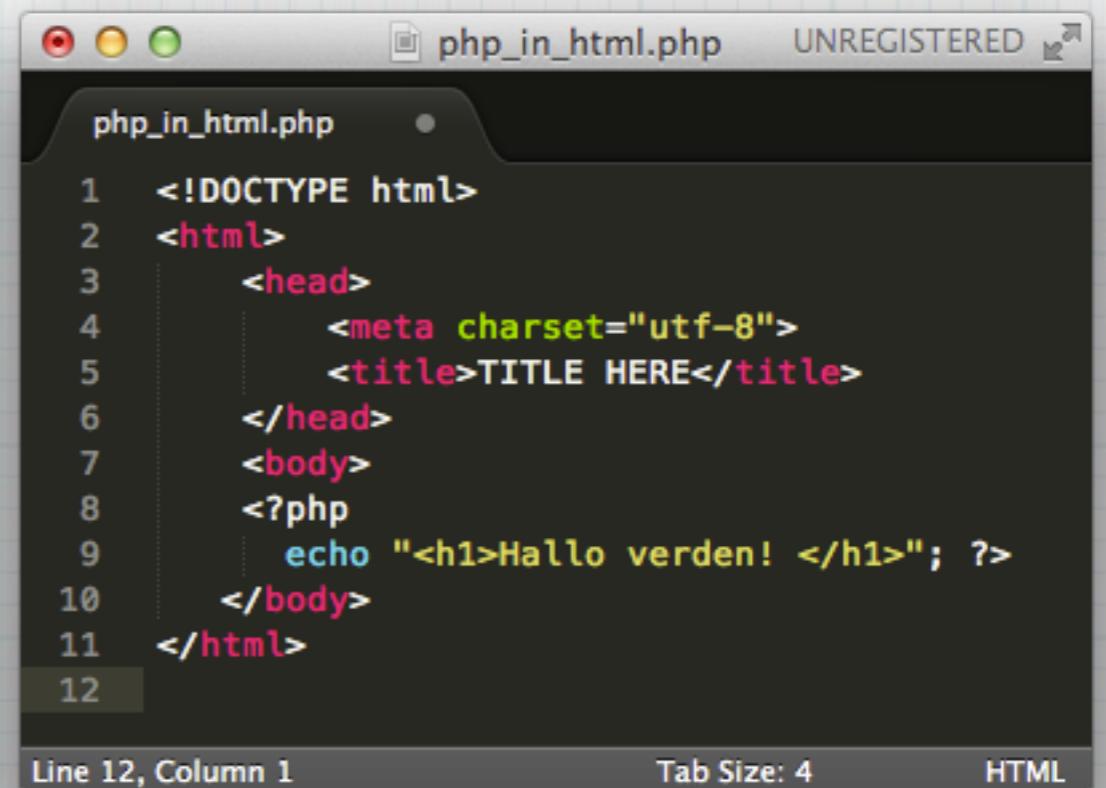
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>TITLE HERE</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1>Hallo verden! </h1>"; ?>
10    </body>
11  </html>
12
```

The code editor interface includes a tab bar with "php\_in\_html.php" and "UNREGISTERED", and status bars at the bottom showing "Line 12, Column 1", "Tab Size: 4", and "HTML".

# PHP

# “Hallo Verden”

- \* PHP er i likhet med JS et “interpretert” “scriptspråk”
- \* Kjører på webseveren.
- \* Produserer (hvis man vil) HTML
- \* Har gjerne lese/skrivetilgang egne steder på server



A screenshot of a code editor window titled "php\_in\_html.php". The code editor displays the following PHP script:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>TITLE HERE</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1>Hallo verden! </h1>"; ?>
10    </body>
11  </html>
```

The code editor interface includes a status bar at the bottom with "Line 12, Column 1", "Tab Size: 4", and "HTML".

# PHP

# “Hallo Verden”

The image shows three separate code editors, each with a dark theme and light-colored syntax highlighting. They are arranged vertically.

- Top Editor:** titled "html\_head.html". It contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```
- Middle Editor:** titled "php1.php". It contains the following code:

```
1 <?php
2 include "html_head.html";
3 echo "<h1>Hallo verden!</h1>";
4 include "html_foot.html";
5 ?>
6 |
```
- Bottom Editor:** titled "html\_foot.html". It contains the following code:

```
1 </body>
2 </html>
```

Each editor has a status bar at the bottom indicating the current line and column number. The middle editor also shows a "Tab Size: 4" setting.

# PHP “Hallo Verden”

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)

The image shows a Mac OS X desktop with three terminal windows open:

- html\_head.html**: Contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```
- php1.php**: Contains the following code:

```
1 <?php
2 include "html_head.html";
3 echo "<h1>Hallo verden!</h1>";
4 include "html_foot.html";
5 ?>
6 |
```
- html\_foot.html**: Contains the following code:

```
1 </body>
2 </html>
```

The bottom right corner of the middle window shows "Tab Size: 4".

# PHP “Hallo Verden”

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)
- \* Eller skrive PHP med “echo” eller include av HTML-biter

The image shows three separate windows from a Mac OS X desktop environment, each displaying a code editor with syntax highlighting. The top window is titled 'html\_head.html' and contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```

The middle window is titled 'php1.php' and contains the following PHP code:

```
1 <?php
2 include "html_head.html";
3 echo "<h1>Hallo verden!</h1>";
4 include "html_foot.html";
5 ?>
6 |
```

The bottom window is titled 'html\_foot.html' and contains the following HTML code:

```
1 </body>
2 </html>
```

Each window has its own status bar at the bottom indicating the current line and column number.

# PHP “Hallo Verden”

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)
- \* Eller skrive PHP med “echo” eller include av HTML-biter
- \* Det viktige er at det er gyldig HTML som blir sendt til browser.

The screenshot shows three separate code editors side-by-side:

- html\_head.html:** Contains the opening HTML document structure: DOCTYPE, html, head, meta charset="utf-8", and title "PHP 1".
- php1.php:** Contains PHP code that includes the "html\_head.html" file and outputs the string "<h1>Hallo verden!</h1>".
- html\_foot.html:** Contains the closing HTML document structure: body and html tags.

# PHP “Hallo Verden”

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)
- \* Eller skrive PHP med “echo” eller include av HTML-biter
- \* Det viktige er at det er gyldig HTML som blir sendt til browser.
- \* Man slipper å kopiere HTML for undersider

The screenshot shows three code editors side-by-side:

- html\_head.html:** Contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```
- php1.php:** Contains the following code:

```
1 <?php
2   include "html_head.html";
3   echo "<h1>Hallo verden!</h1>";
4   include "html_foot.html";
5 ?>
```
- html\_foot.html:** Contains the following code:

```
1 </body>
2 </html>
```

The bottom right editor has a tab size of 4.

# PHP

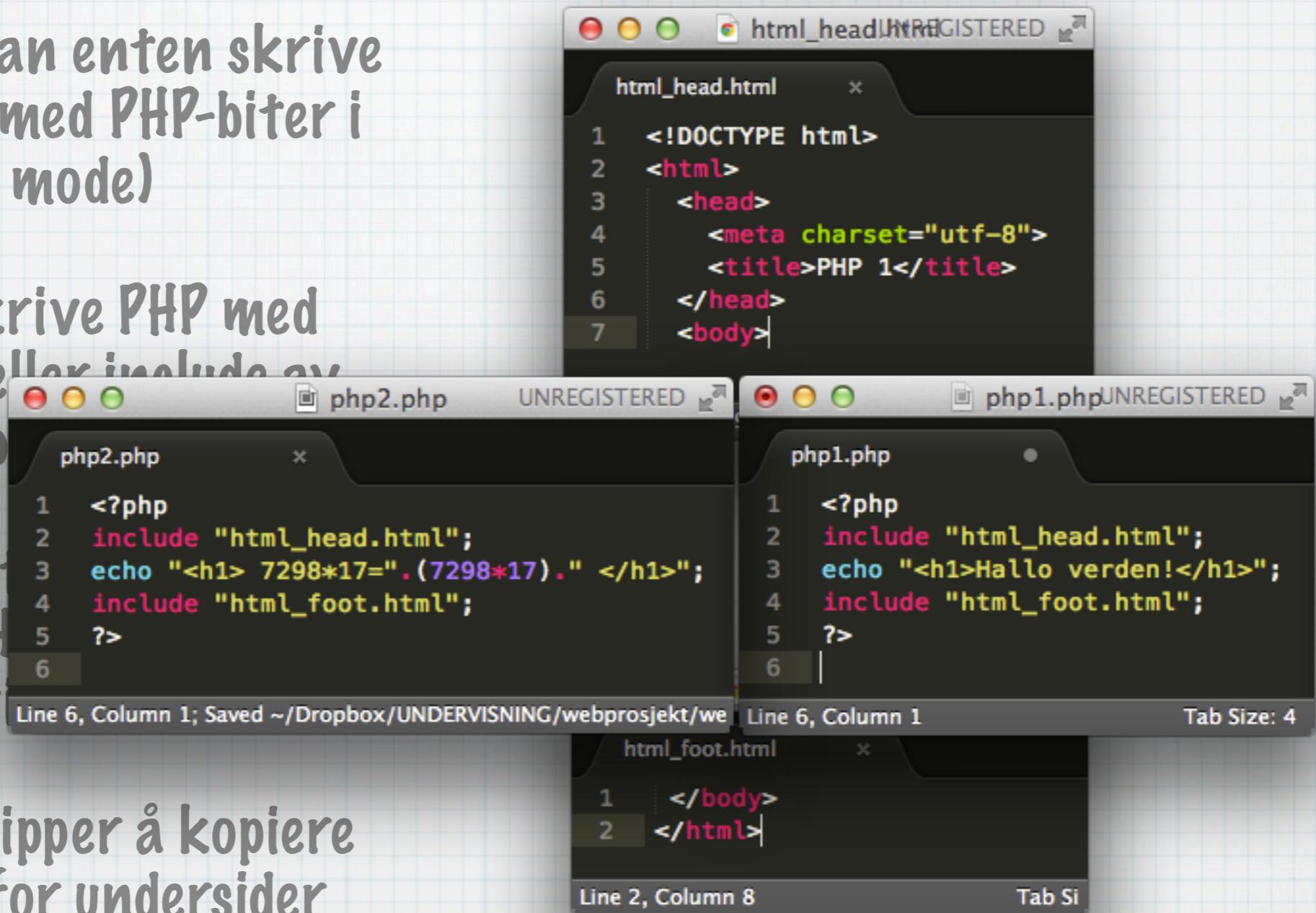
# “Hallo Verden”

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)

- \* Eller skrive PHP med “echo” eller include av HTML-biter

- \* Det viktigste er at det er gyldig HTML da man sendt til en browser

- \* Man slipper å kopiere HTML for undersider



```
html_head.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```

```
php1.php
1 <?php
2   include "html_head.html";
3   echo "<h1> 7298*17=".(7298*17)."</h1>";
4   include "html_foot.html";
5 ?>
```

```
php2.php
1 <?php
2   include "html_head.html";
3   echo "<h1>Hallo verden!</h1>";
4   include "html_foot.html";
5 ?>
```

```
html_foot.html
1 </body>
2 </html>
```

# PHP

# “Hallo Verden”

Deles!

- \* Man kan enten skrive HTML, med PHP-biter i (Mixed mode)

- \* Eller skrive PHP med “echo” eller include av HTML-biter

- \* Det viktigste er at det er gyldig HTML da man sendt til en browser

- \* Man slipper å kopiere HTML for undersider

```
html_head.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>PHP 1</title>
6   </head>
7   <body>
```

```
php1.php
1 <?php
2   include "html_head.html";
3   echo "<h1> 7298*17=".(7298*17)."</h1>";
4   include "html_foot.html";
5 ?>
```

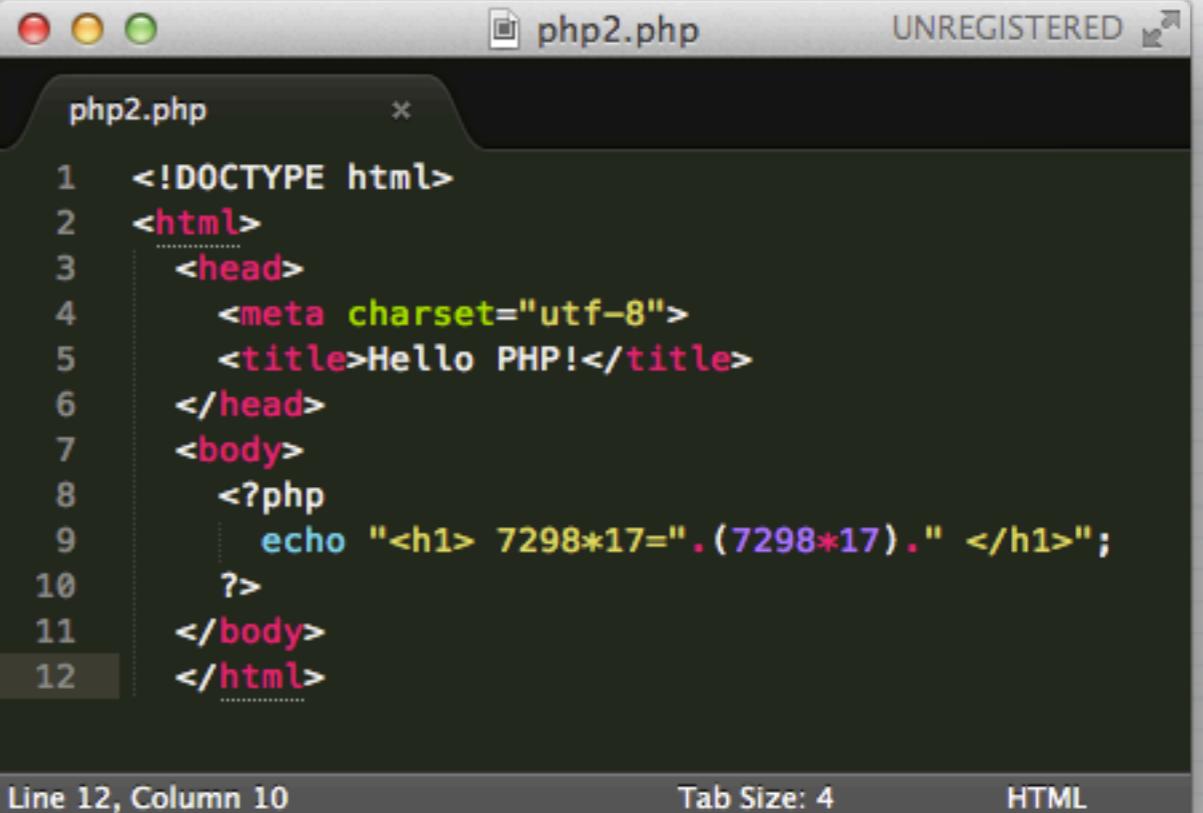
  

```
php2.php
1 <?php
2   include "html_head.html";
3   echo "<h1>Hallo verden!</h1>";
4   include "html_foot.html";
5 ?>
```

```
html_foot.html
1 </body>
2 </html>
```

# “Trigge” PHP



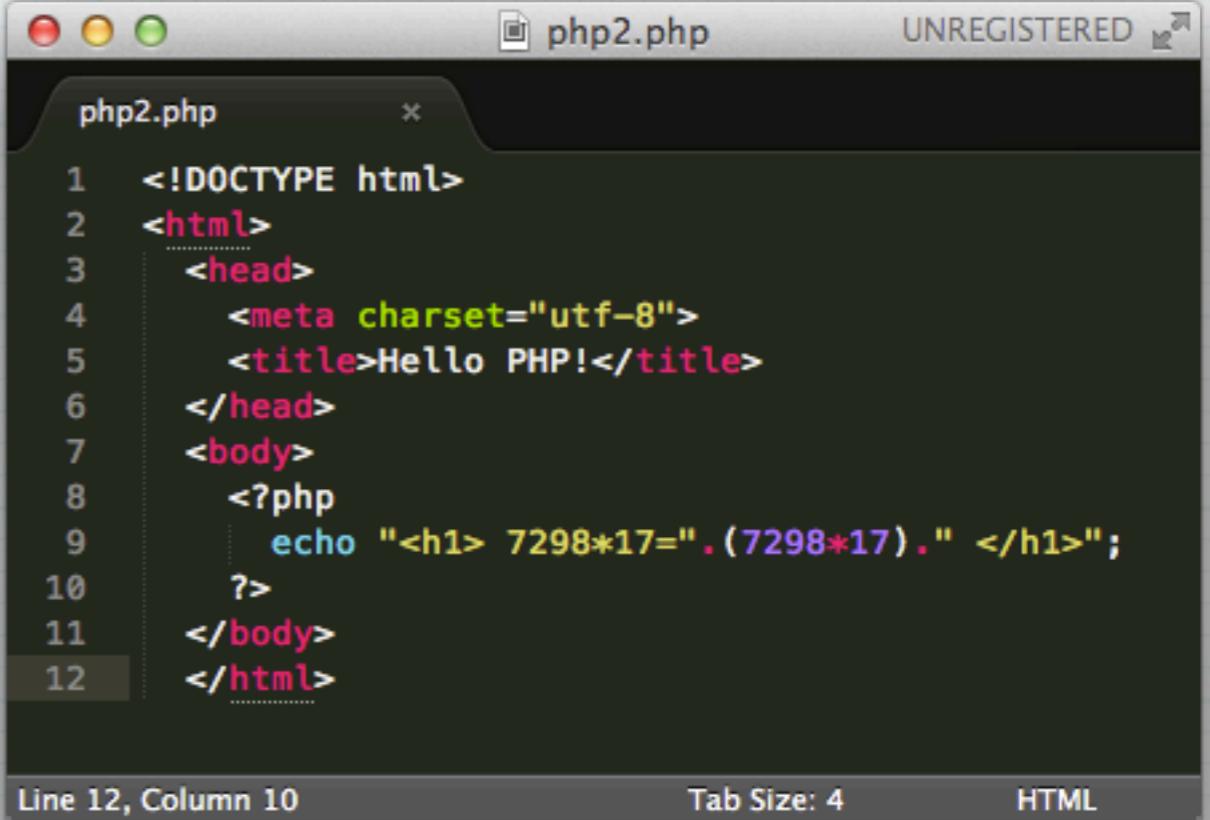
A screenshot of a Mac OS X TextEdit window titled "php2.php". The window shows a PHP script with syntax highlighting. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Hello PHP!</title>
6   </head>
7   <body>
8     <?php
9       echo "<h1> 7298*17=".(7298*17)."</h1>";
10    ?>
11  </body>
12 </html>
```

The status bar at the bottom of the window indicates "Line 12, Column 10", "Tab Size: 4", and "HTML".

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?

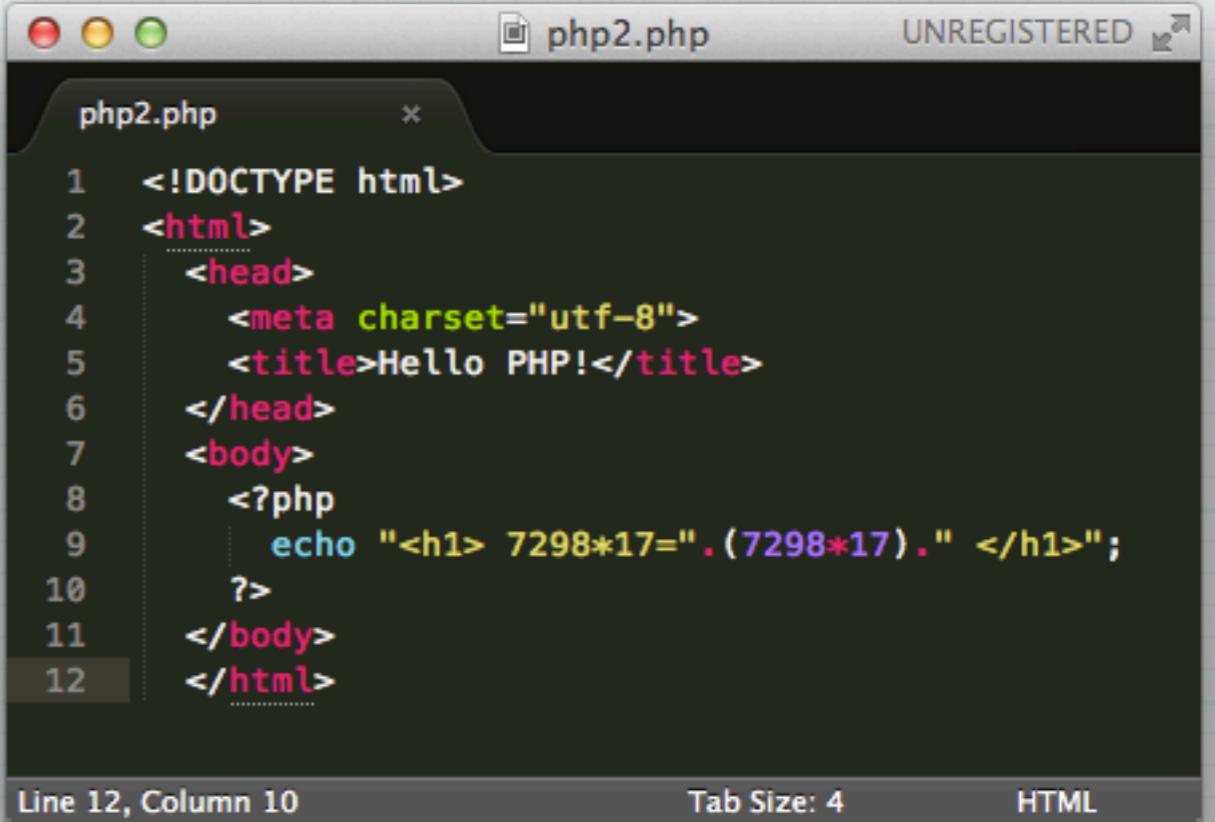


```
php2.php
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello PHP!</title>
6      </head>
7      <body>
8          <?php
9              echo "<h1> 7298*17=".(7298*17)."</h1>";
10         ?>
11     </body>
12 </html>
```

Line 12, Column 10      Tab Size: 4      HTML

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?
- \* Hva “Er” PHP?

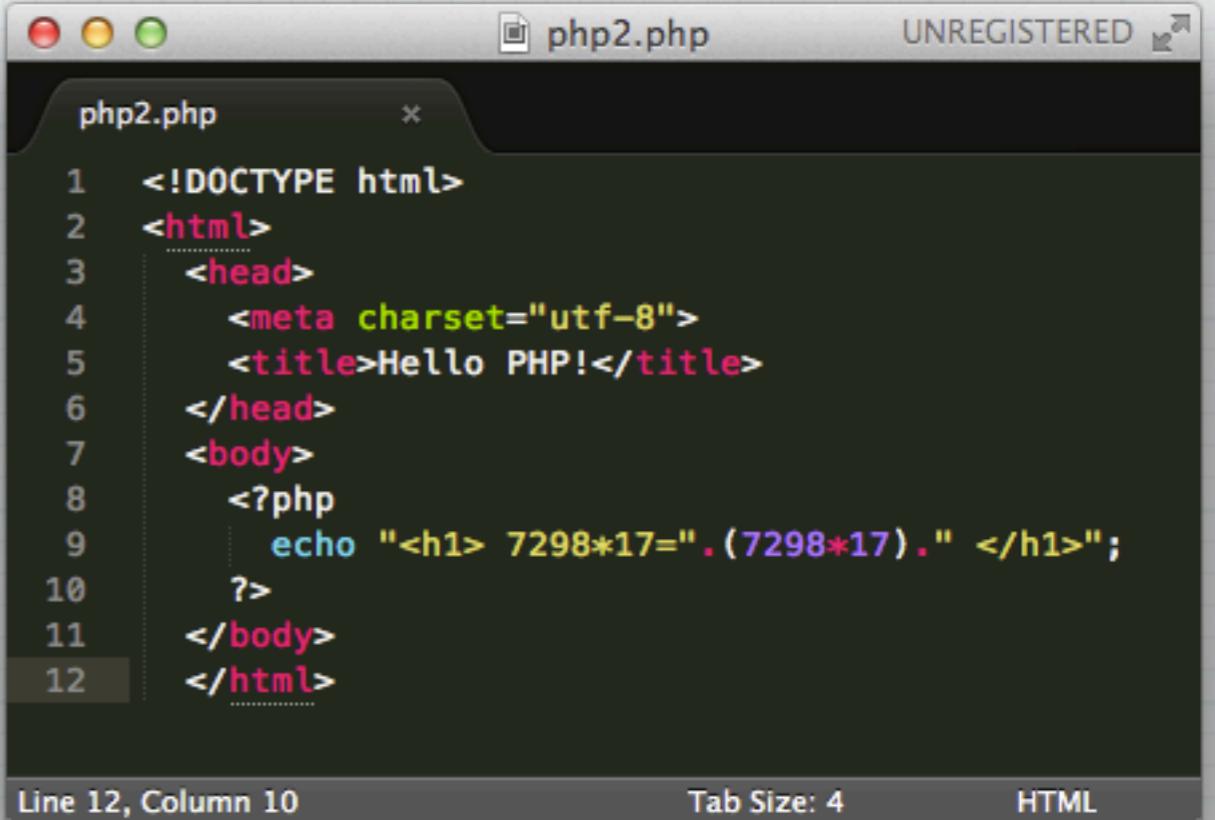


```
php2.php
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello PHP!</title>
6      </head>
7      <body>
8          <?php
9              echo "<h1> 7298*17=".(7298*17)."</h1>";
10         ?>
11     </body>
12 </html>
```

Line 12, Column 10      Tab Size: 4      HTML

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?
- \* Hva “Er” PHP?
  - \* Et program

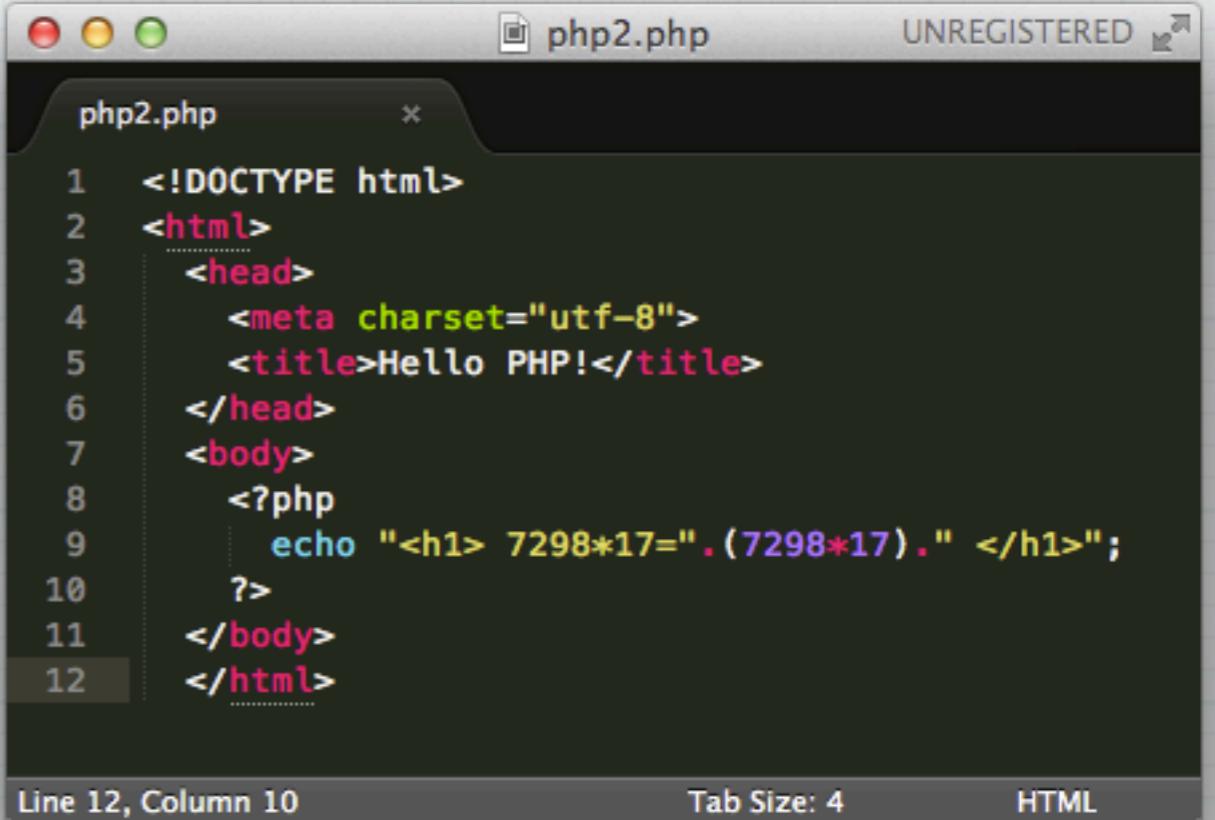


```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello PHP!</title>
6      </head>
7      <body>
8          <?php
9              echo "<h1> 7298*17=".(7298*17)."</h1>";
10         ?>
11     </body>
12 </html>
```

Line 12, Column 10      Tab Size: 4      HTML

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?
- \* Hva “Er” PHP?
  - \* Et program
  - \* Kan kalles via terminal,  
eller av webserver



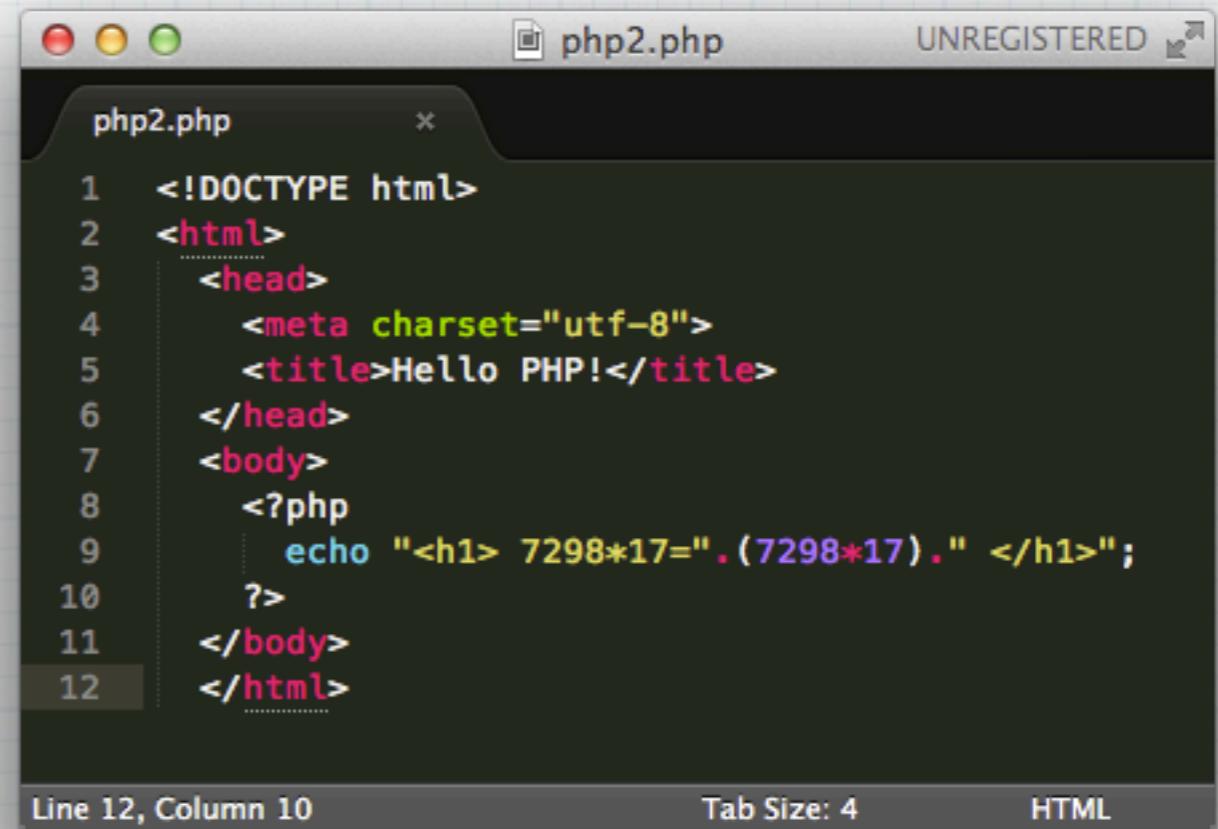
A screenshot of a code editor window titled "php2.php". The window shows a dark-themed interface with a tab bar at the top labeled "php2.php" and "UNREGISTERED". The main area contains the following PHP code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello PHP!</title>
6      </head>
7      <body>
8          <?php
9              echo "<h1> 7298*17=".(7298*17)."</h1>";
10         ?>
11     </body>
12 </html>
```

The status bar at the bottom of the editor shows "Line 12, Column 10", "Tab Size: 4", and "HTML".

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?
- \* Hva “Er” PHP?
  - \* Et program
  - \* Kan kalles via terminal,  
eller av webserver
- \* Må være installert på  
server, sammen med  
webserver (Feks. Apache)

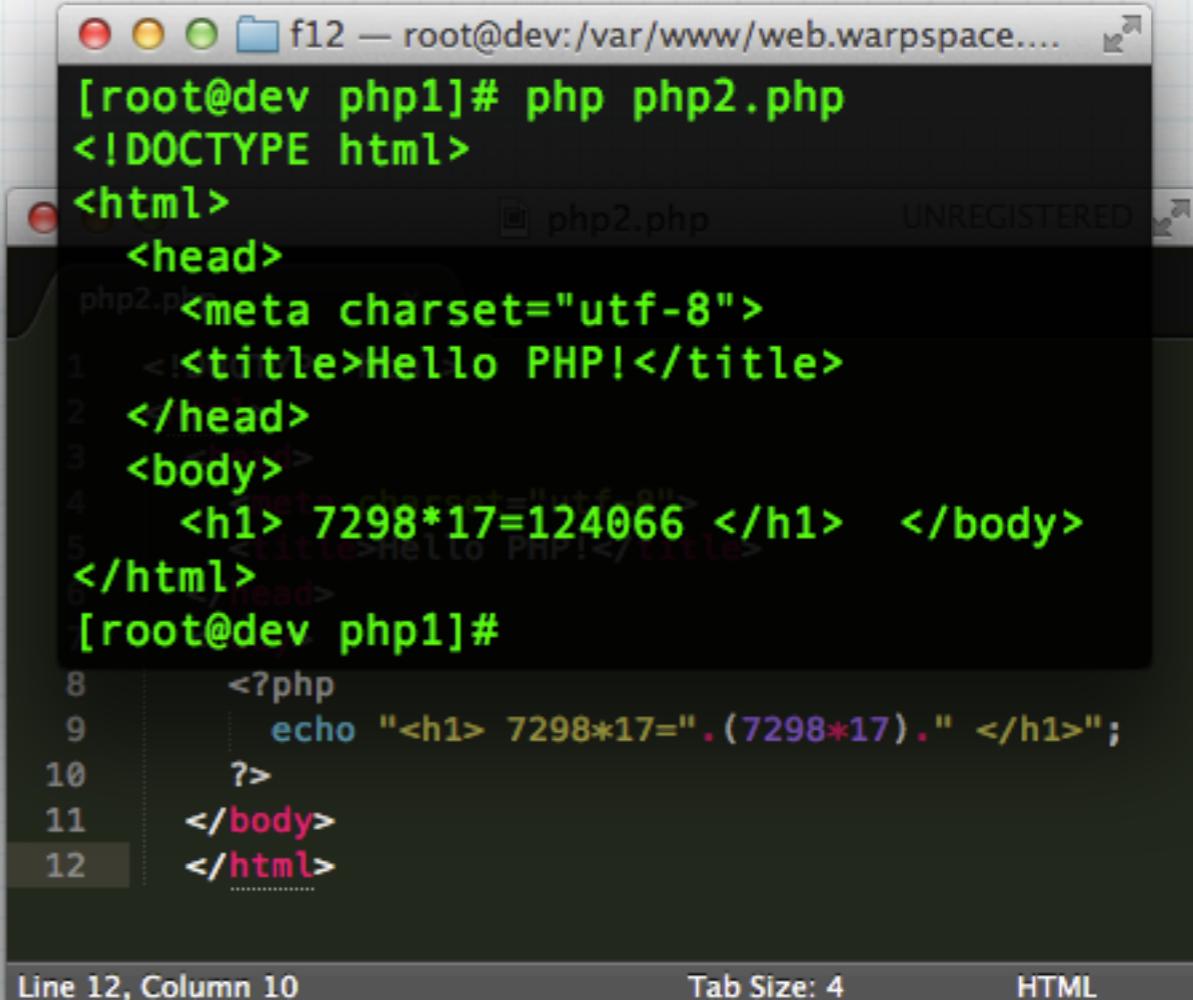


A screenshot of a code editor window titled "php2.php". The window shows a PHP script with syntax highlighting. The code includes HTML tags like <html>, <head>, <title>, and <body>, along with PHP code within <?php> tags. The editor interface includes a toolbar at the top, a status bar at the bottom with "Line 12, Column 10", "Tab Size: 4", and "HTML" tabs.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello PHP!</title>
6      </head>
7      <body>
8          <?php
9              echo "<h1> 7298*17=". (7298*17). "</h1>";
10         ?>
11     </body>
12 </html>
```

# “Trigge” PHP

- \* <?php .... ?>  
likner HTML. Er det?
- \* Hva “Er” PHP?
  - \* Et program
  - \* Kan kalles via terminal,  
eller av webserver
- \* Må være installert på  
server, sammen med  
webserver (Feks. Apache)



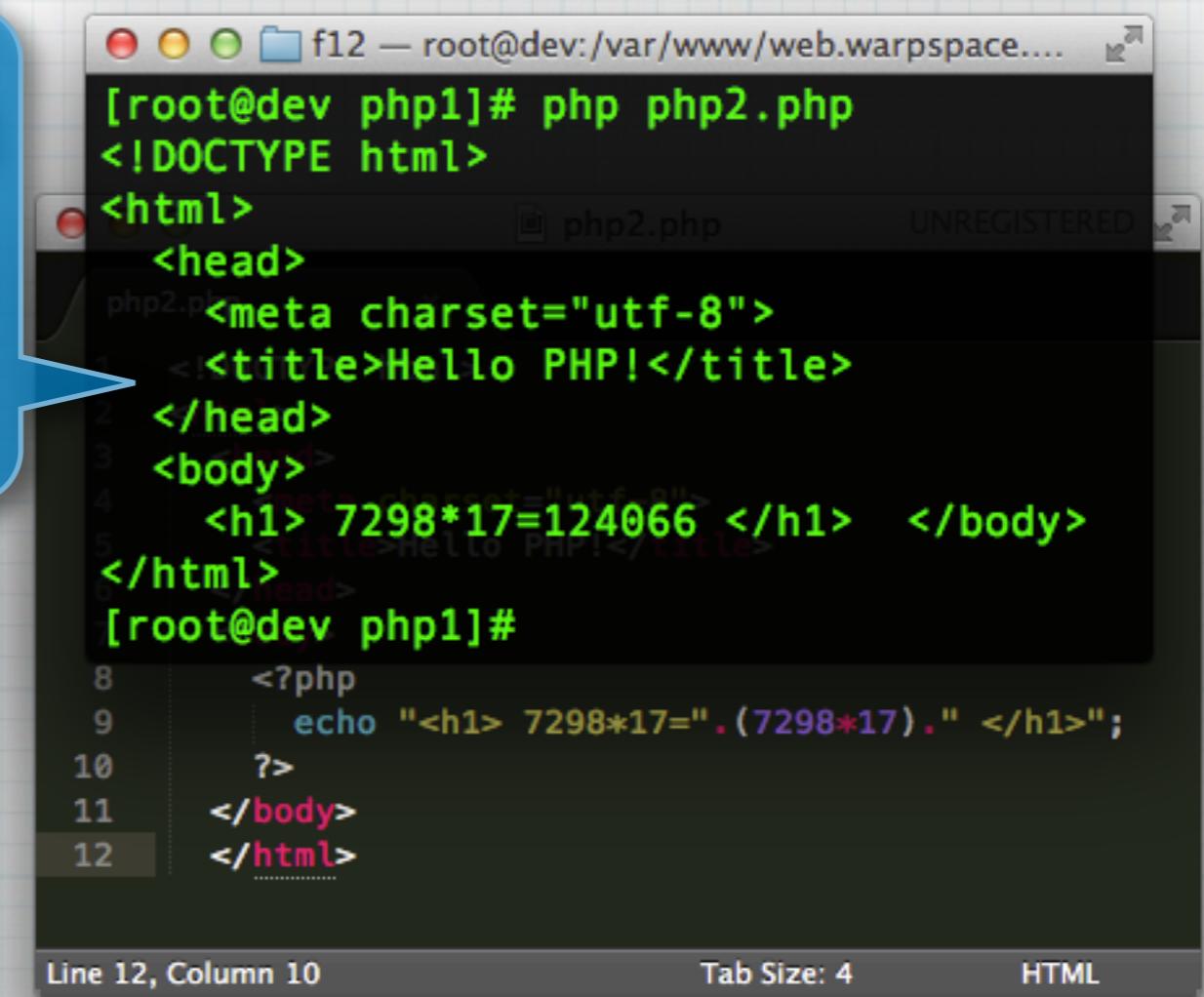
A screenshot of a terminal window titled "f12 — root@dev:/var/www/web.warpspace....". The window displays the following command and its output:

```
[root@dev php1]# php php2.php
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello PHP!</title>
  </head>
  <body>
    <h1> 7298*17=124066 </h1>
  </body>
</html>
[root@dev php1]#
8      <?php
9      echo "<h1> 7298*17=".(7298*17)."</h1>";
10     ?>
11   </body>
12   </html>
```

The terminal shows line numbers 1 through 12. The output of the PHP code is displayed in green, showing the result of the multiplication  $7298 \times 17 = 124066$ . The status bar at the bottom of the terminal window indicates "Line 12, Column 10", "Tab Size: 4", and "HTML".

# “Trigge” PHP

- \* <?php ... ?>  
likner HTML. Funksjon  
PHP spiser fila og genererer HTML
- \* Hva “Er” PHP?
- \* Et program
- \* Kan kalles via terminal, eller av webserver
- \* Må være installert på server, sammen med webserver (Feks. Apache)



A screenshot of a terminal window titled "f12 — root@dev:/var/www/web.warpsspace....". The window shows the command "php php2.php" being run, which outputs the following HTML code:

```
[root@dev php1]# php php2.php
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello PHP!</title>
  </head>
  <body>
    <h1> 7298*17=124066 </h1>
  </body>
</html>
[root@dev php1]#
8   <?php
9     echo "<h1> 7298*17=".(7298*17)."</h1>";
10    ?>
11  </body>
12  </html>
```

The terminal also displays the file path "php2.php" and status "UNREGISTERED". At the bottom, it shows "Line 12, Column 10", "Tab Size: 4", and "HTML".

# “Trigge” PHP

- \* <?php ... ?> likner HTML. Funksjon
- \* PHP spiser fila og genererer HTML
- \* Hva “Er” PHP?
- \* Et program
- \* Kan kalles via terminal, eller av webserver
- \* Må være installert på server, sammen med webserver (Feks. Apache)

The image shows a terminal window and a web browser window. The terminal window (f12 — root@dev:/var/www/web.warpspace.org) displays the command `php php2.php` and its output:

```
[root@dev php1]# php php2.php
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello PHP!</title>
  </head>
  <body>
    <h1> 7298*17=124066 </h1>
  </body>
</html>
[root@dev php1]#
```

The browser window (Hello PHP!) shows the generated HTML output:

7298\*17=124066

# “Trigge” PHP

- \* <?php ... ?> likner HTML. Først til PHP spiser fila og genererer HTML
- \* Hva “Er” PHP?
- \* Et program
- \* Apache sender HTML over HTTP til browser
- \* Må være installert på server, sammen med webserver (Apache)

The screenshot shows a terminal window titled 'f12 — root@dev:/var/www/web.warpspace....' and a web browser window.

**Terminal:**

```
[root@dev php1]# php php2.php
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello PHP!</title>
  </head>
  <body>
    <h1> 7298*17=124066 </h1>
  </body>
</html>
[root@dev php1]#
```

**Browser:**

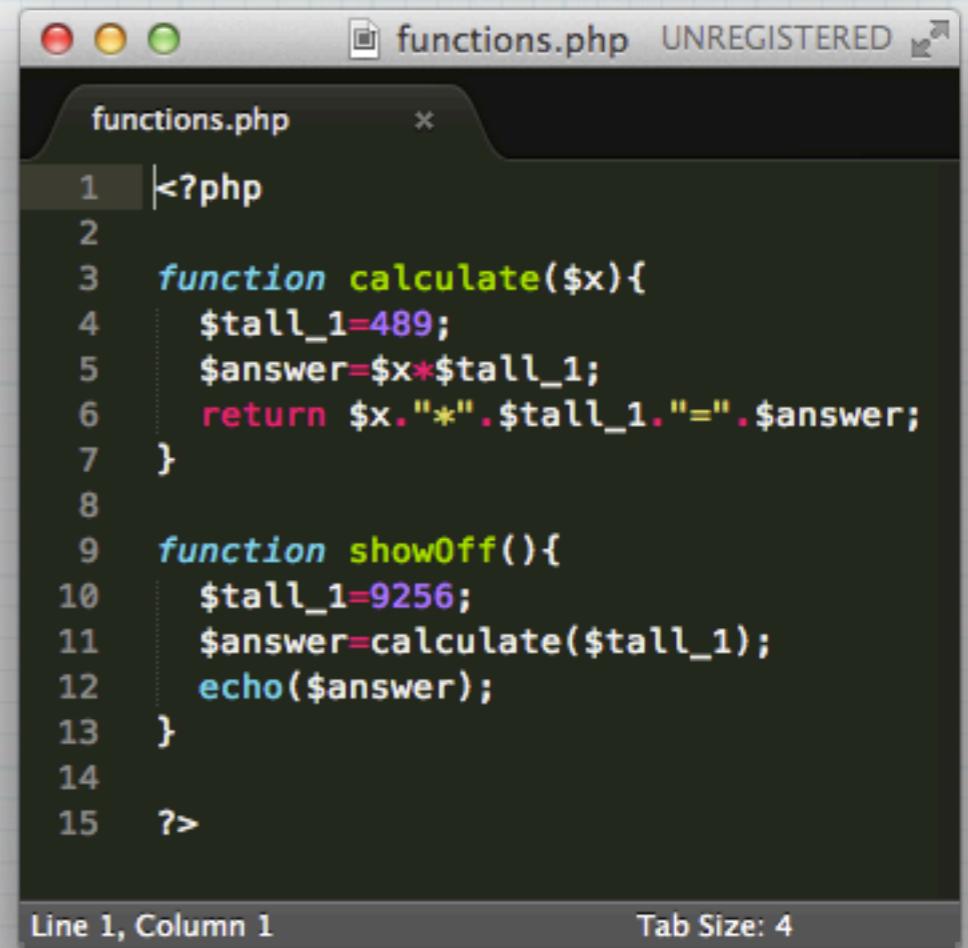
The browser window title is 'Hello PHP!' and the URL is 'web.warpspace.org/php1/'. The page content displays the calculation **7298\*17=124066**.

# Demo

---

phpl.php  
php2.php

# PHP-basics: Funksjoner



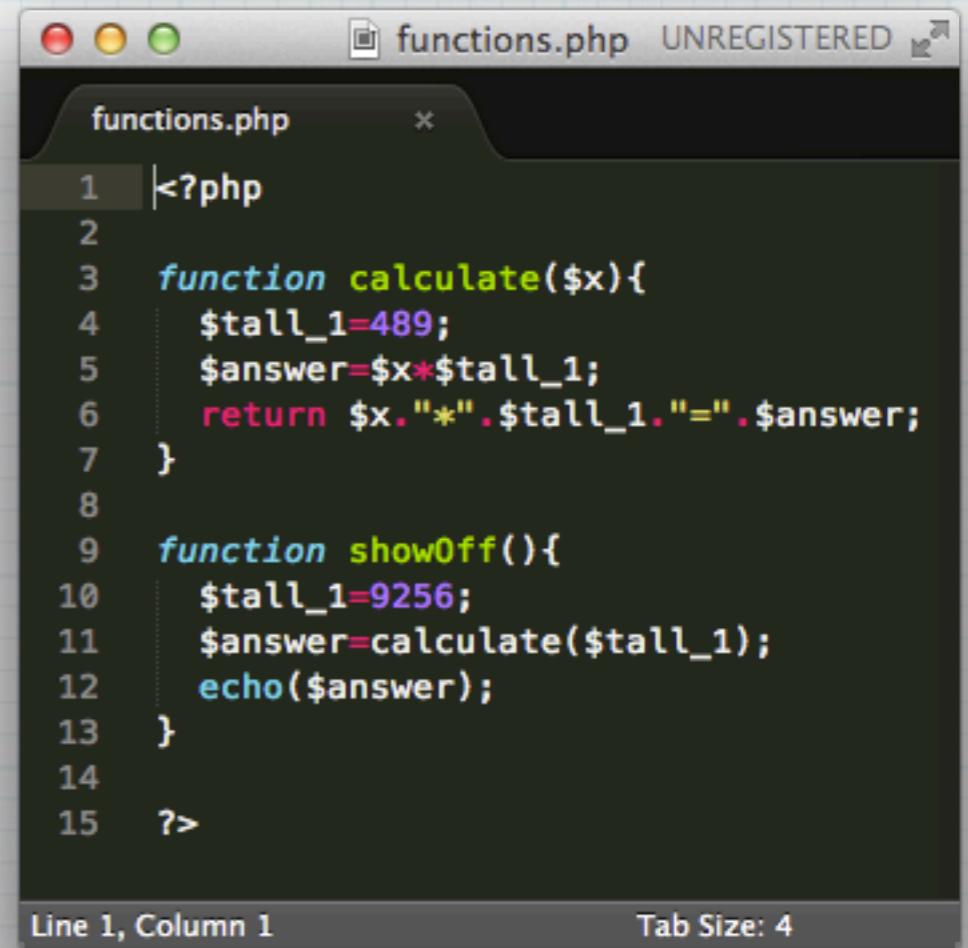
A screenshot of a code editor window titled "functions.php UNREGISTERED". The code editor displays the following PHP script:

```
1 <?php
2
3 function calculate($x){
4     $tall_1=489;
5     $answer=$x*$tall_1;
6     return $x."*".$tall_1."=". $answer;
7 }
8
9 function showOff(){
10    $tall_1=9256;
11    $answer=calculate($tall_1);
12    echo($answer);
13 }
14
15 ?>
```

The status bar at the bottom of the editor shows "Line 1, Column 1" and "Tab Size: 4".

# PHP-basics: Funksjoner

- \* Hva er en funksjon?



A screenshot of a code editor window titled "functions.php UNREGISTERED". The code is written in PHP and defines two functions: "calculate" and "showOff". The "calculate" function takes a parameter \$x, multiplies it by a constant \$tall\_1 (set to 489), and returns the result. The "showOff" function sets a constant \$tall\_1 to 9256, calls the calculate function, and then echoes the result. The code is numbered from 1 to 15.

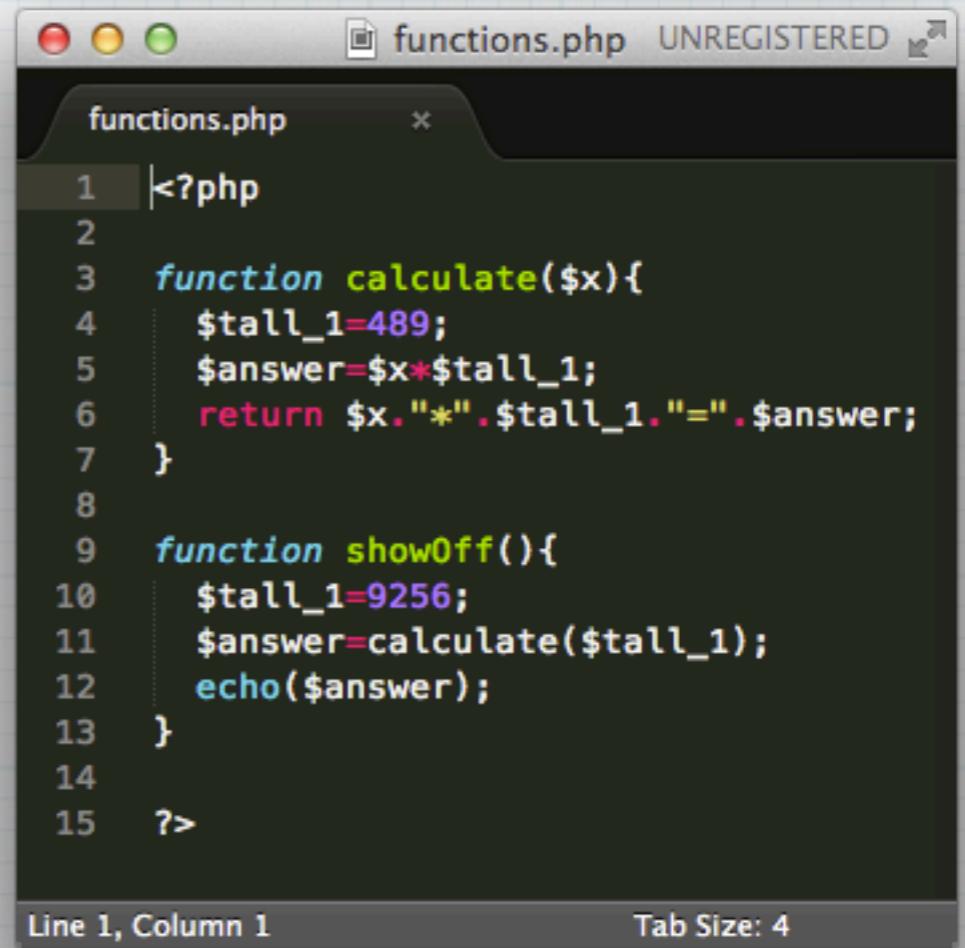
```
<?php
function calculate($x){
    $tall_1=489;
    $answer=$x*$tall_1;
    return $x."*".$tall_1."=". $answer;
}

function showOff(){
    $tall_1=9256;
    $answer=calculate($tall_1);
    echo($answer);
}

?>
```

# PHP-basics: Funksjoner

- \* Hva er en funksjon?
- \* Rent matematisk:  
“(Black) box”-maskineri,  
med kun input og output



A screenshot of a code editor window titled "functions.php UNREGISTERED". The code is written in PHP and defines two functions: "calculate" and "showOff". The "calculate" function takes a single argument \$x, multiplies it by a constant \$tall\_1 (set to 489), and returns the result. The "showOff" function also uses \$tall\_1 (set to 9256) and calls the "calculate" function, then prints the result using echo. The code is numbered from 1 to 15.

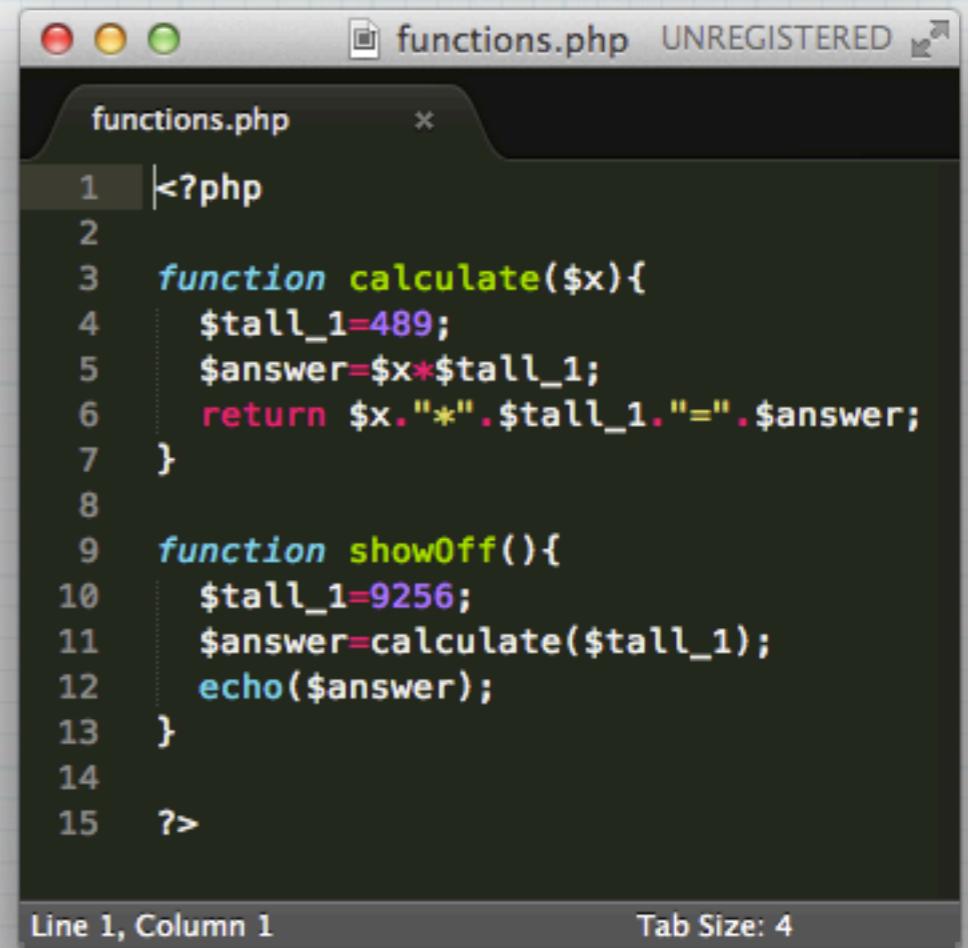
```
<?php
function calculate($x){
    $tall_1=489;
    $answer=$x*$tall_1;
    return $x."*".$tall_1."=". $answer;
}

function showOff(){
    $tall_1=9256;
    $answer=calculate($tall_1);
    echo($answer);
}

?>
```

# PHP-basics: Funksjoner

- \* Hva er en funksjon?
- \* Rent matematisk:  
“(Black) box”-maskineri,  
med kun input og output
- \* I programmering; ofte  
også “side-effects” -  
endrer på noe (minne,  
filer, piksler etc.)



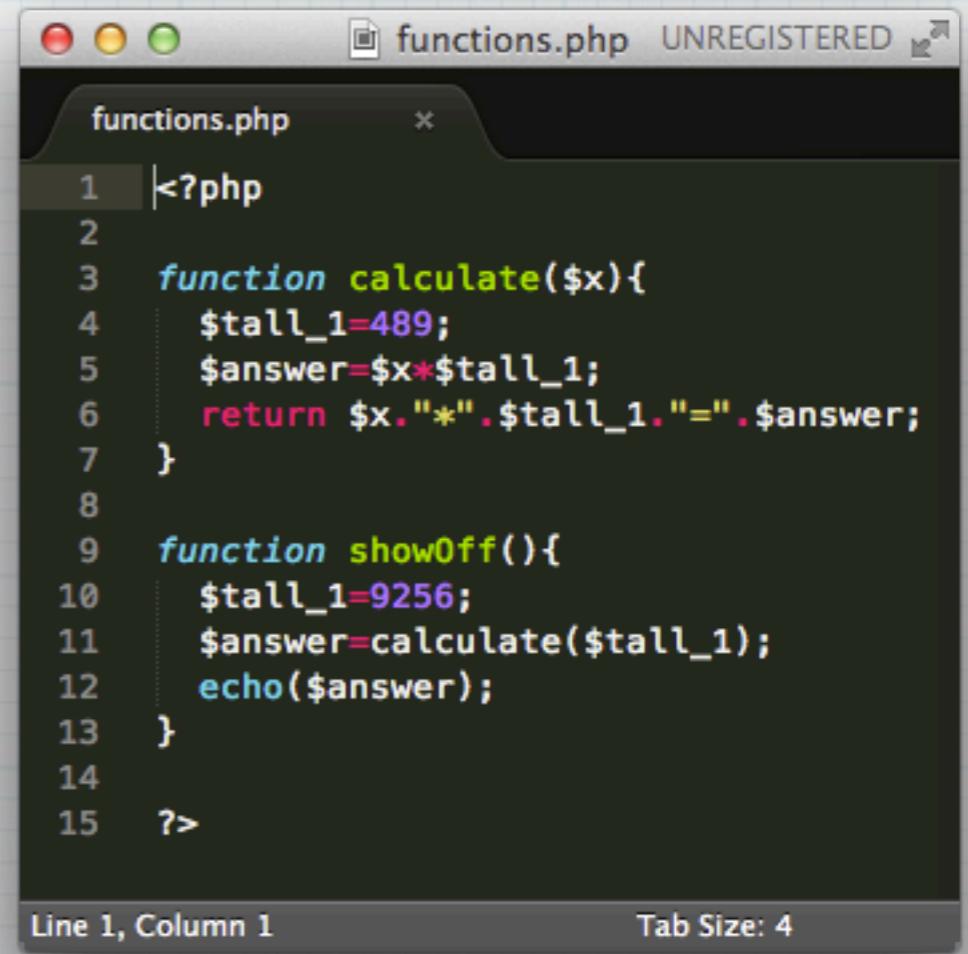
A screenshot of a Mac OS X TextEdit window titled "functions.php UNREGISTERED". The window contains the following PHP code:

```
1 <?php
2
3 function calculate($x){
4     $tall_1=489;
5     $answer=$x*$tall_1;
6     return $x."*".$tall_1."=". $answer;
7 }
8
9 function showOff(){
10    $tall_1=9256;
11    $answer=calculate($tall_1);
12    echo($answer);
13 }
14
15 ?>
```

The status bar at the bottom of the window shows "Line 1, Column 1" and "Tab Size: 4".

# PHP-basics: Funksjoner

- \* Hva er en funksjon?
  - \* Rent matematisk:  
“(Black) box”-maskineri,  
med kun input og output
  - \* I programmering; ofte  
også “side-effects” -  
endrer på noe (minne,  
filer, piksler etc.)
  - \* PHP har maaaaaange  
Innebygde (php.net)



A screenshot of a Mac OS X TextEdit window titled "functions.php UNREGISTERED". The window contains the following PHP code:

```
<?php
function calculate($x){
    $tall_1=489;
    $answer=$x*$tall_1;
    return $x."*".$tall_1."=". $answer;
}

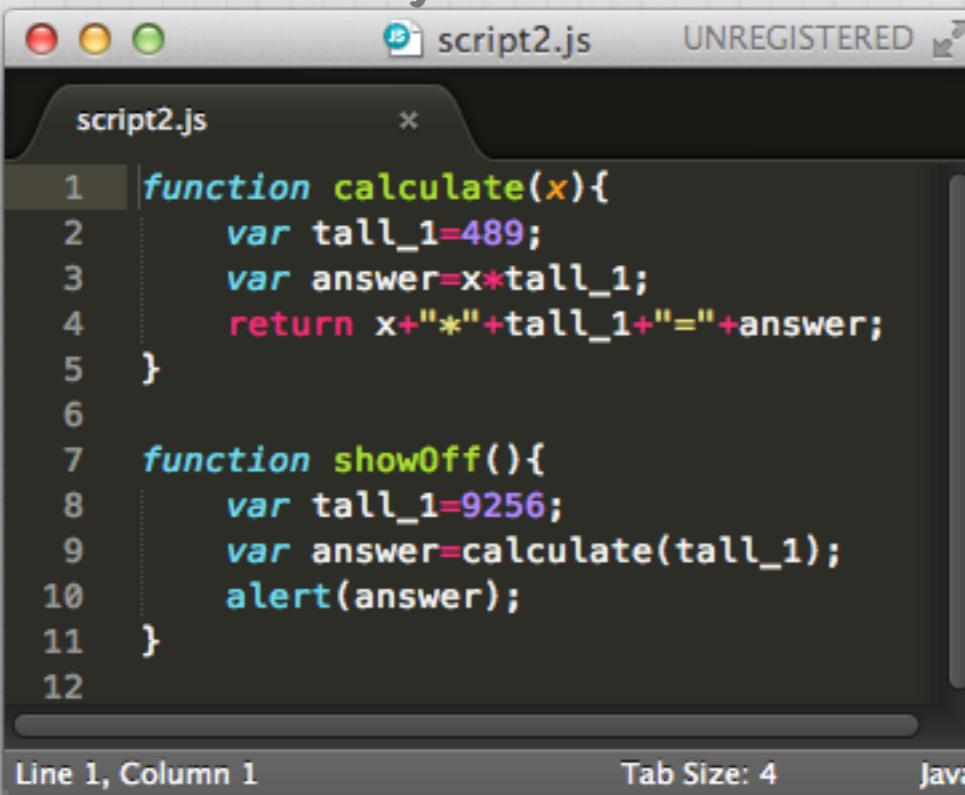
function showOff(){
    $tall_1=9256;
    $answer=calculate($tall_1);
    echo($answer);
}

?>
```

The status bar at the bottom shows "Line 1, Column 1" and "Tab Size: 4".

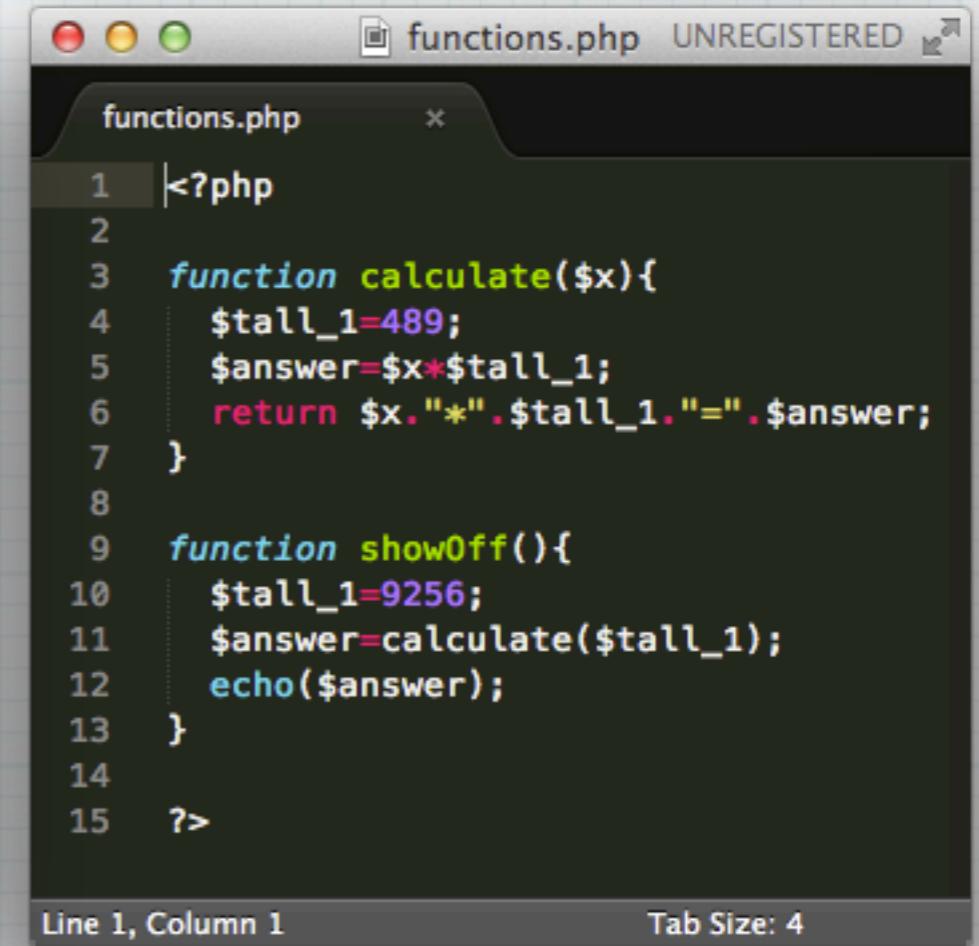
# PHP-basics: Funksjoner

- \* Hva er en funksjon?



```
script2.js
UNREGISTERED
function calculate(x){
    var tall_1=489;
    var answer=x*tall_1;
    return x+"*"+tall_1+"=".answer;
}

function showOff(){
    var tall_1=9256;
    var answer=calculate(tall_1);
    alert(answer);
}
```



```
functions.php
<?php

function calculate($x){
    $tall_1=489;
    $answer=$x*$tall_1;
    return $x."*".$tall_1."=".$answer;
}

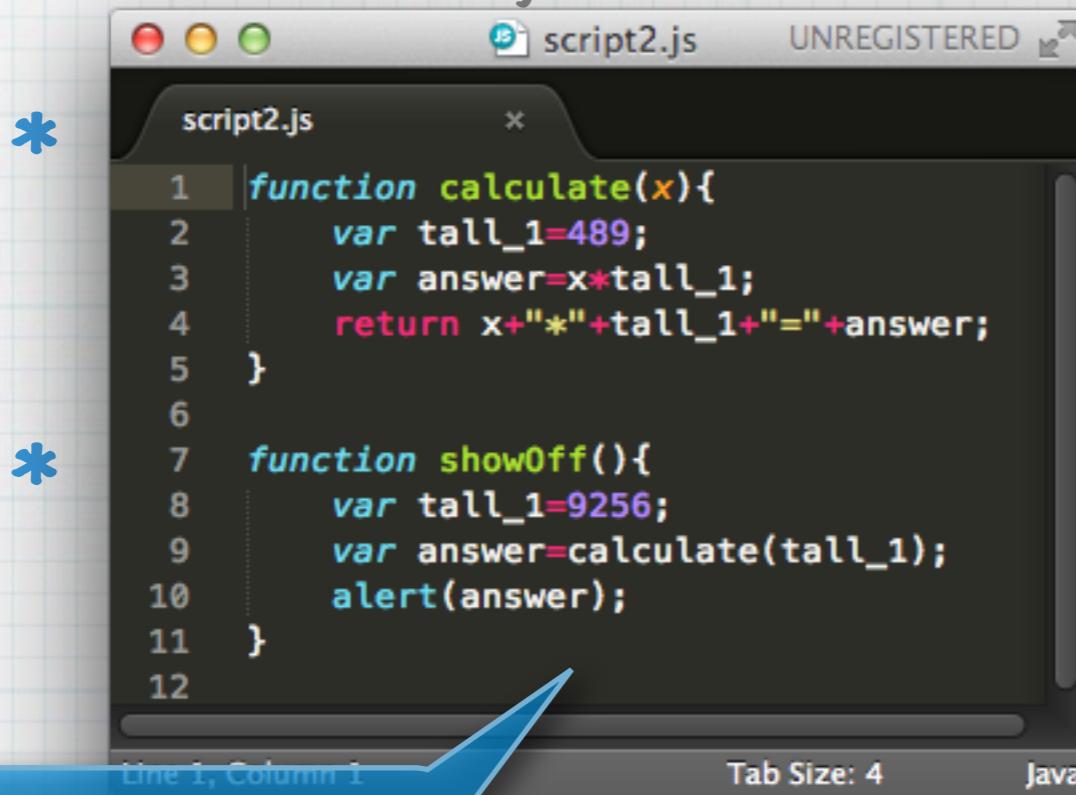
function showOff(){
    $tall_1=9256;
    $answer=calculate($tall_1);
    echo($answer);
}

?>
```

- \* PHP har maaaaange  
Innebygde (php.net)

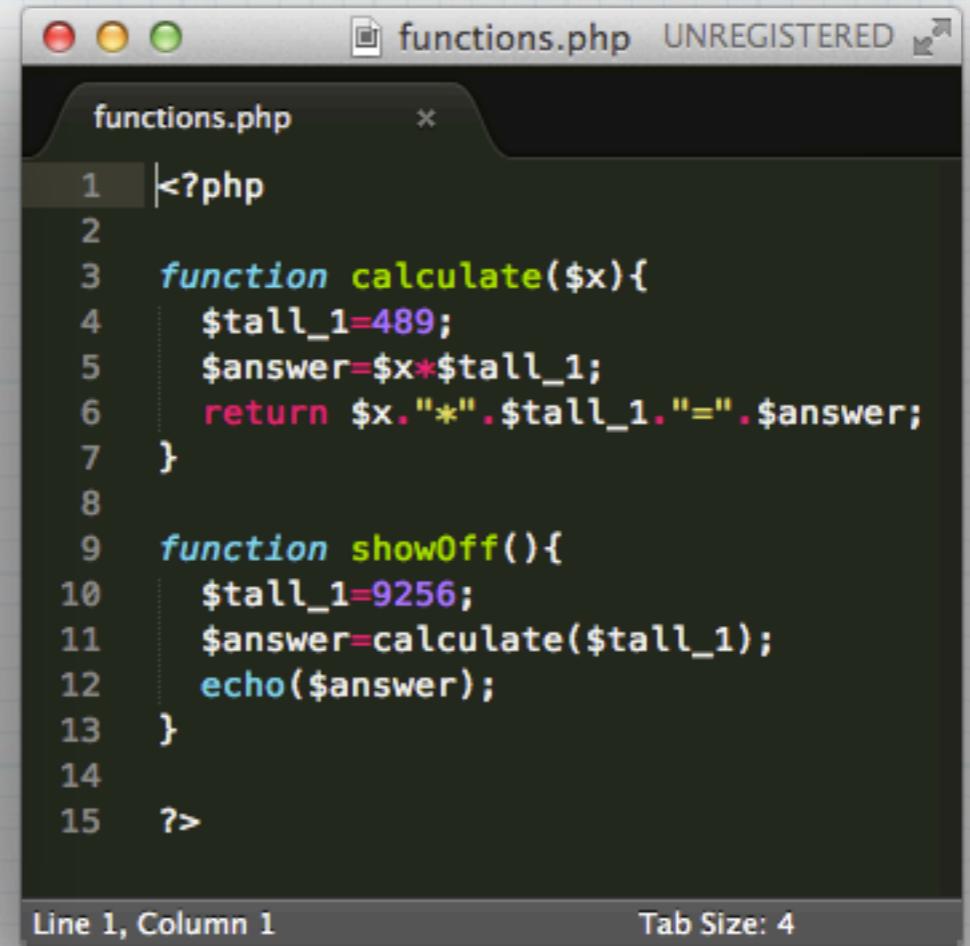
# PHP-basics: Funksjoner

## \* Hva er en funksjon?



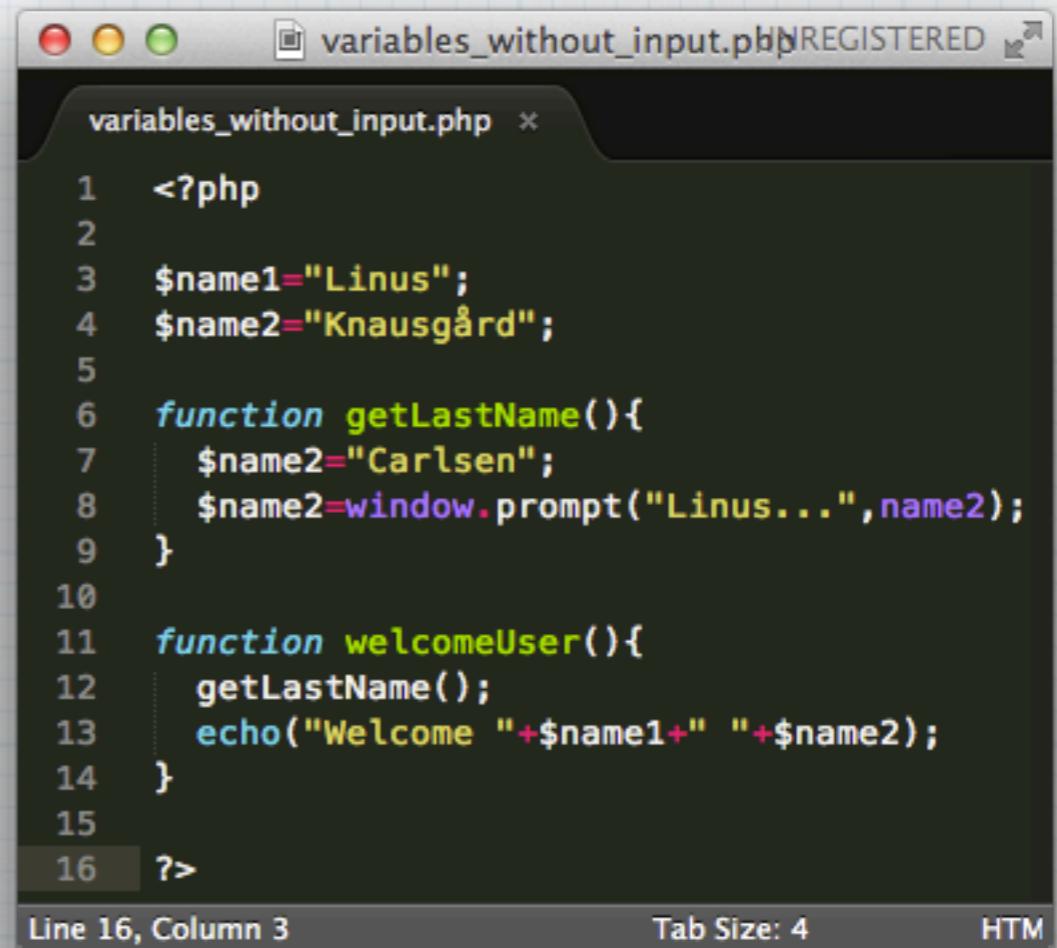
```
script2.js *  
*  
1 function calculate(x){  
2     var tall_1=489;  
3     var answer=x*tall_1;  
4     return x+"*"+tall_1+"=".answer;  
5 }  
6  
*  
7 function showOff(){  
8     var tall_1=9256;  
9     var answer=calculate(tall_1);  
10    alert(answer);  
11 }  
12  
Line 1, Column 1 Tab Size: 4 Java
```

Ganske likt  
javascript!  
(Så langt...)



```
functions.php UNREGISTERED  
functions.php *  
1 <?php  
2  
3 function calculate($x){  
4     $tall_1=489;  
5     $answer=$x*$tall_1;  
6     return $x."*".$tall_1."=".$answer;  
7 }  
8  
9 function showOff(){  
10    $tall_1=9256;  
11    $answer=calculate($tall_1);  
12    echo($answer);  
13 }  
14  
15 ?>  
Line 1, Column 1 Tab Size: 4
```

# PHP Basics: Variabler



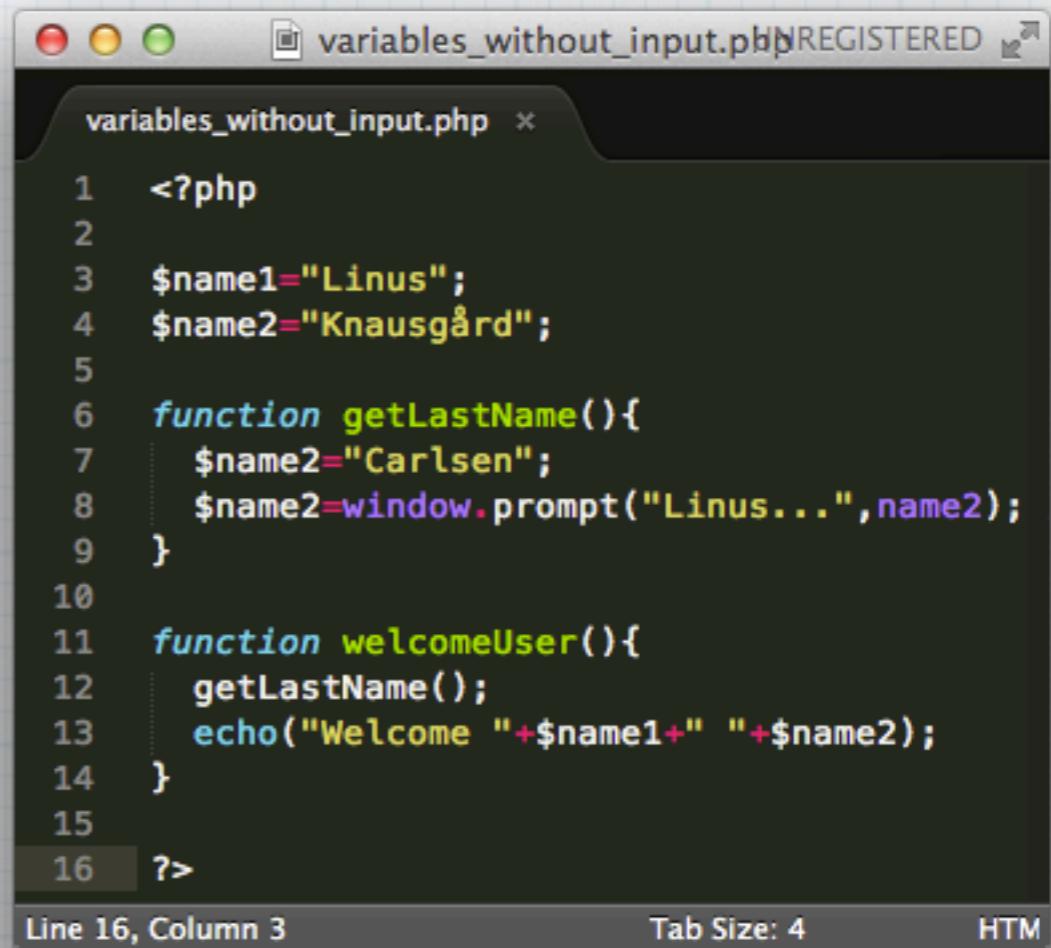
A screenshot of a code editor window titled "variables\_without\_input.php UNREGISTERED". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

The status bar at the bottom shows "Line 16, Column 3", "Tab Size: 4", and "HTM".

# PHP Basics: Variabler

## \* Hva er en variabel?



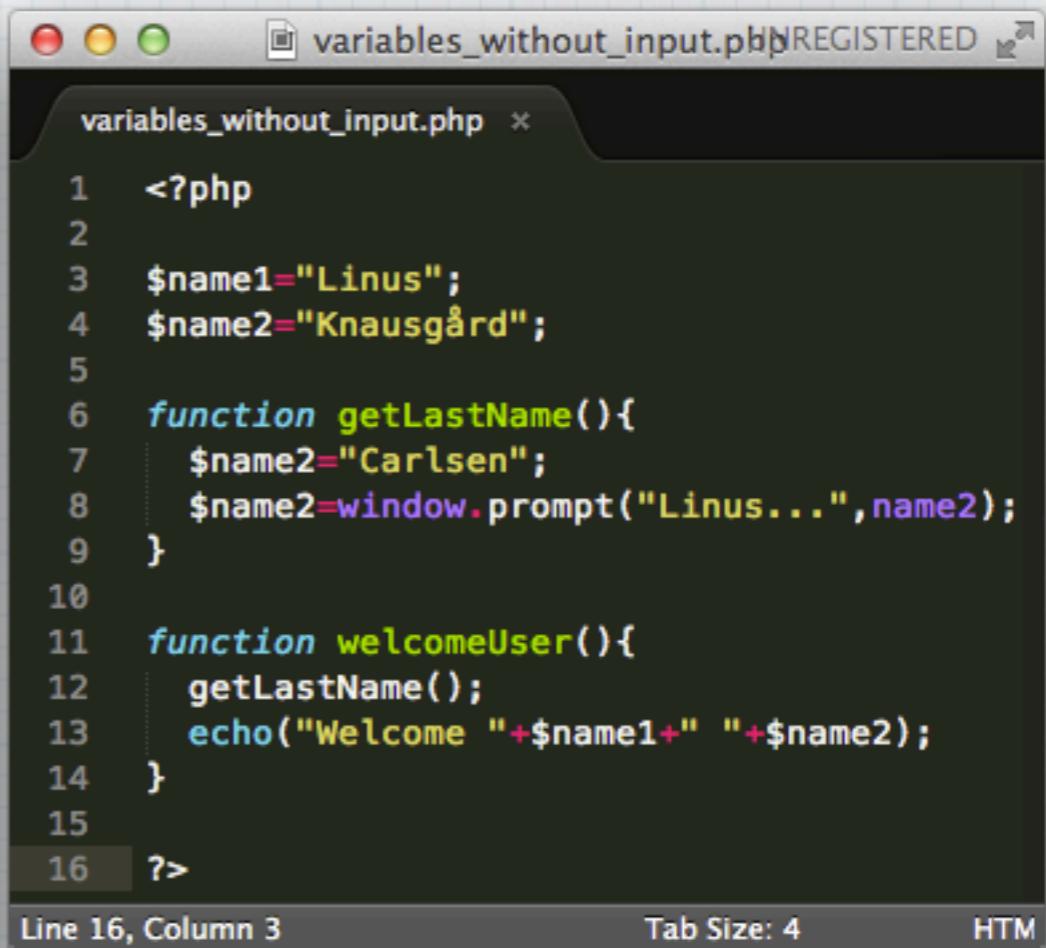
A screenshot of a code editor window titled "variables\_without\_input.php UNREGISTERED". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

The status bar at the bottom shows "Line 16, Column 3", "Tab Size: 4", and "HTM".

# PHP Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn



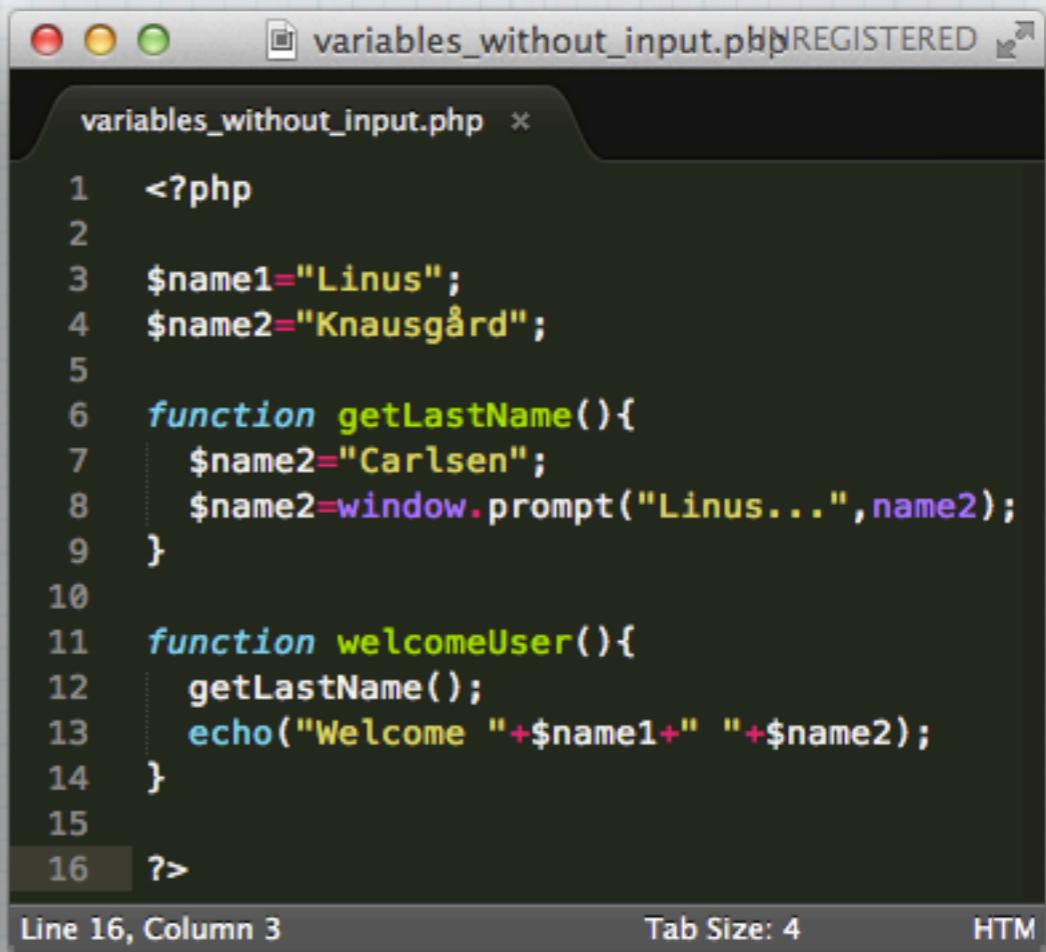
A screenshot of a code editor window titled "variables\_without\_input.php". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

The code editor shows line 16, column 3. The status bar at the bottom right indicates "Tab Size: 4" and "HTM".

# PHP Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn
- \* Finnes i ulike typer; tall, tekststrenger ++



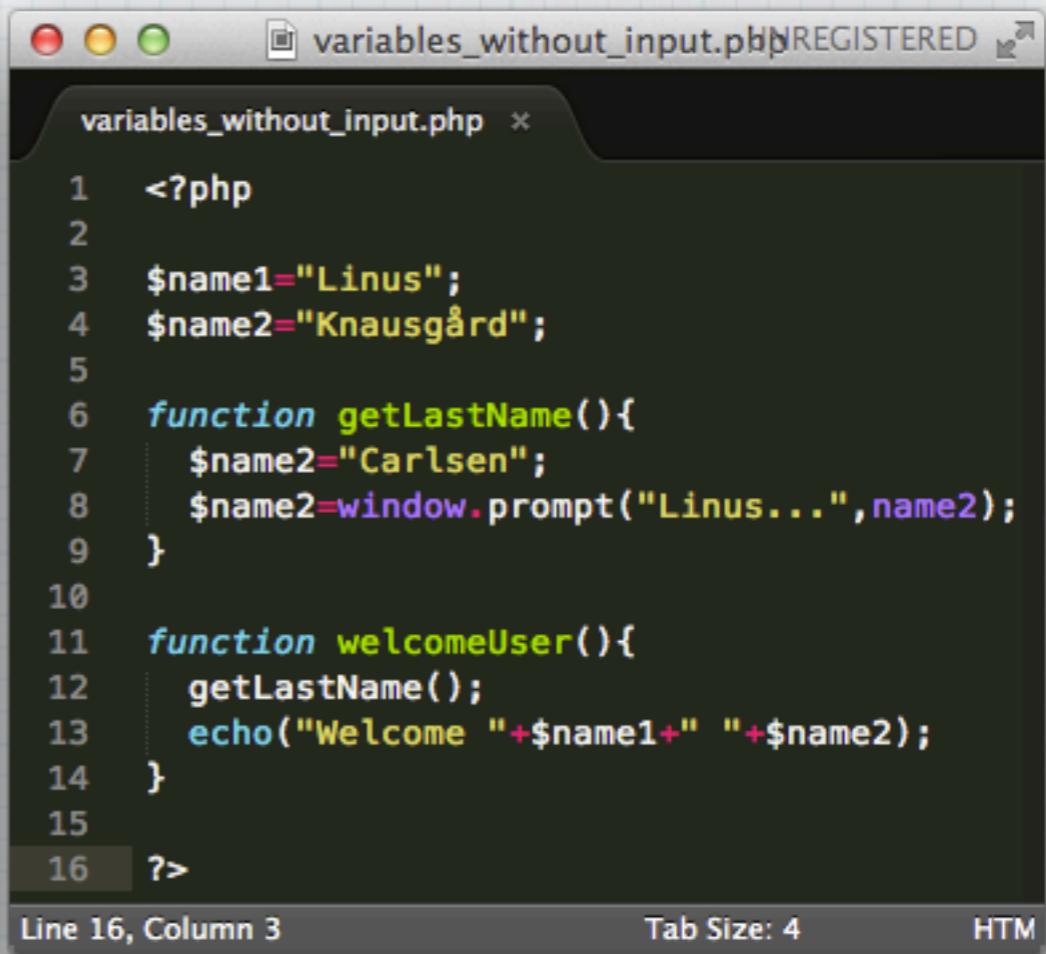
A screenshot of a code editor window titled "variables\_without\_input.php UNREGISTERED". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

The status bar at the bottom shows "Line 16, Column 3", "Tab Size: 4", and "HTM".

# PHP Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn
- \* Finnes i ulike typer; tall, tekststrenger ++
- \* Kan endres - men har et "skop".



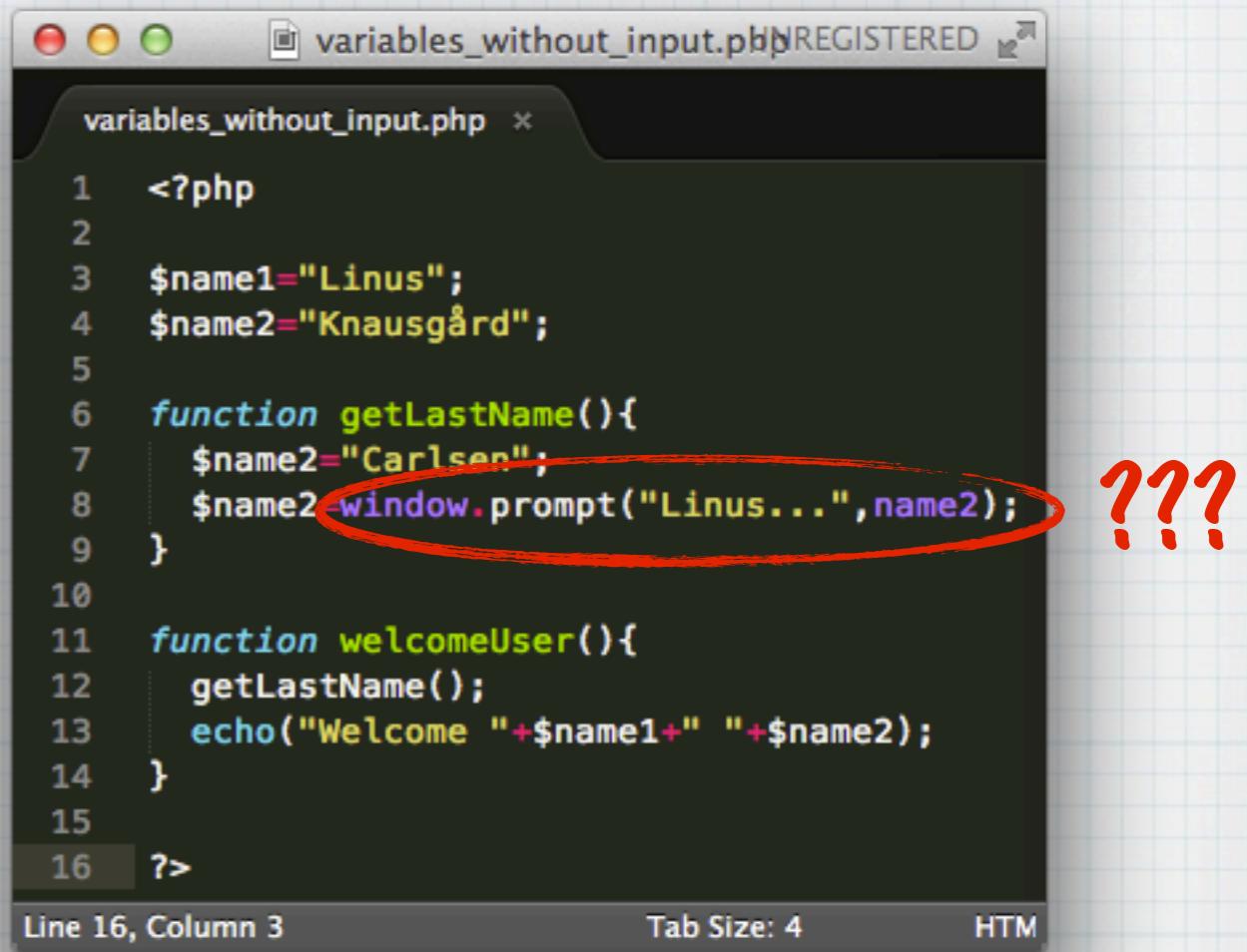
A screenshot of a code editor window titled "variables\_without\_input.php UNREGISTERED". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

The code editor shows line 16, column 3. The status bar indicates "Tab Size: 4" and "HTM".

# PHP Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn
- \* Finnes i ulike typer; tall, tekststrenger ++
- \* Kan endres - men har et "skop".



The screenshot shows a code editor window titled "variables\_without\_input.php UNREGISTERED". The code is as follows:

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2=window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

A red oval highlights the line of code: `$name2=window.prompt("Linus...",name2);`. To the right of the oval is a large red question mark (???) indicating a potential issue or security concern.

# PHP Basics: Variabler

- \* Hva er en variabel?
- \* Data lagret i minnet, med et navn
- \* Finnes i ulike typer; tall, tekststrenger ++
- \* Kan endres - men har et "skop".

PHP Kjører på server.  
Input fra bruker må hentes via nett.

The screenshot shows a code editor with a dark theme. A red oval highlights the line of code where a user input is being retrieved using `window.prompt`. The code is as follows:

```
variables_without_input.php
<?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2=window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```

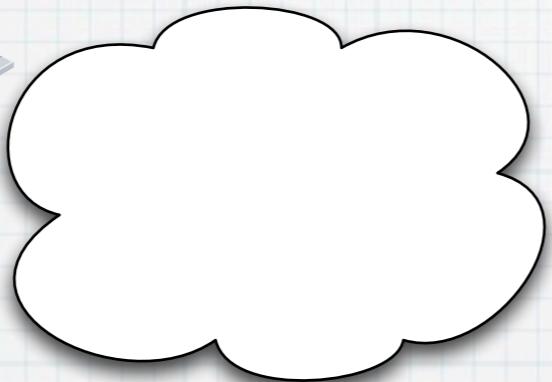
Line 16, Column 3      Tab Size: 4      HTM

???

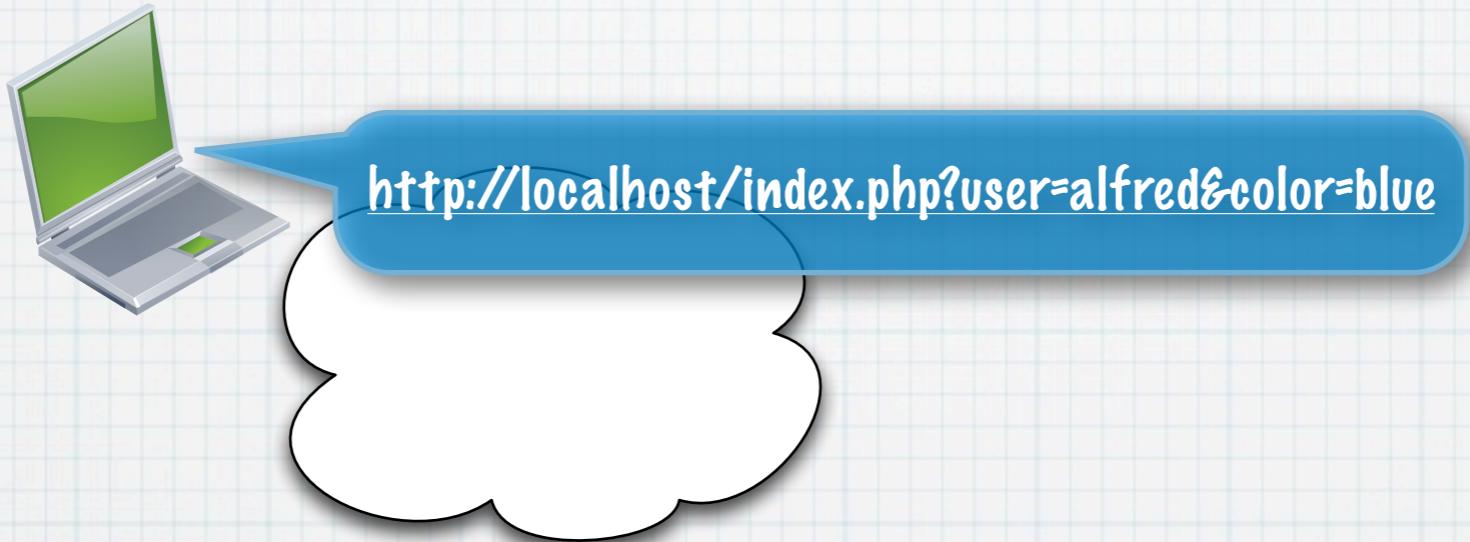
# PHP Basics: Variabler

- \* Hva er en variabel?
  - \* Data lagret i minnet, med et navn
  - \* Finnes i ulike typer; tall, tekst, objekter, osv.
  - \* Kan endres - men har et "skop".
- PHP Kjører på server.  
Input fra bruker må hentes via nett.
- Betyr det at PHP må koble seg til vår PC?
- 
- ```
variables_without_input.php
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     $name2="Carlsen";
8     $name2>window.prompt("Linus...",name2);
9 }
10
11 function welcomeUser(){
12     getLastName();
13     echo("Welcome "+$name1+" "+$name2);
14 }
15
16 ?>
```
- Line 16, Column 3
- Tab Size: 4
- HTM

# Input til PHP: HTTP request-string



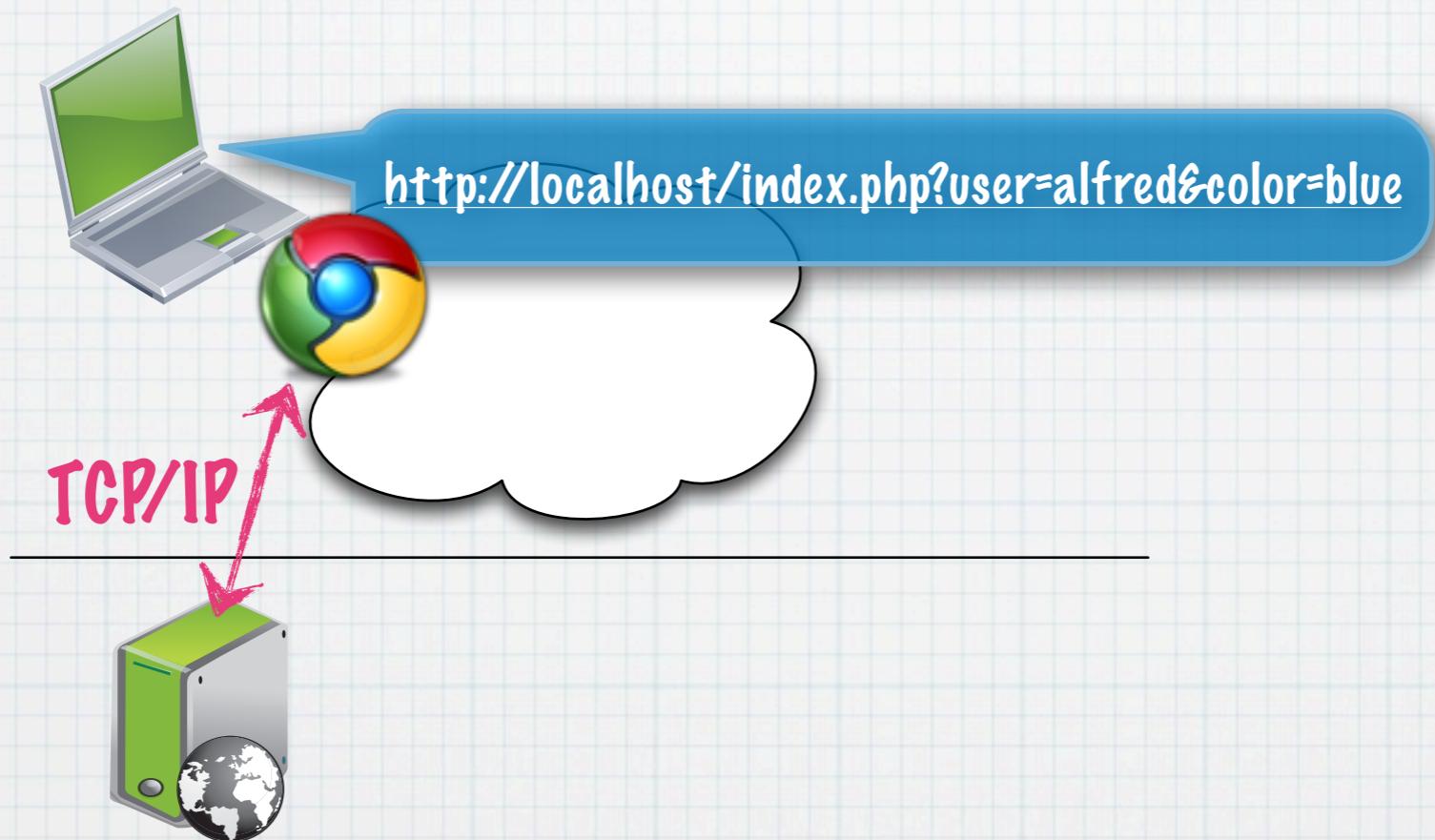
# Input til PHP: HTTP request-string



# Input til PHP: HTTP request-string



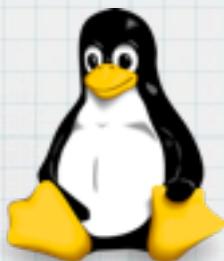
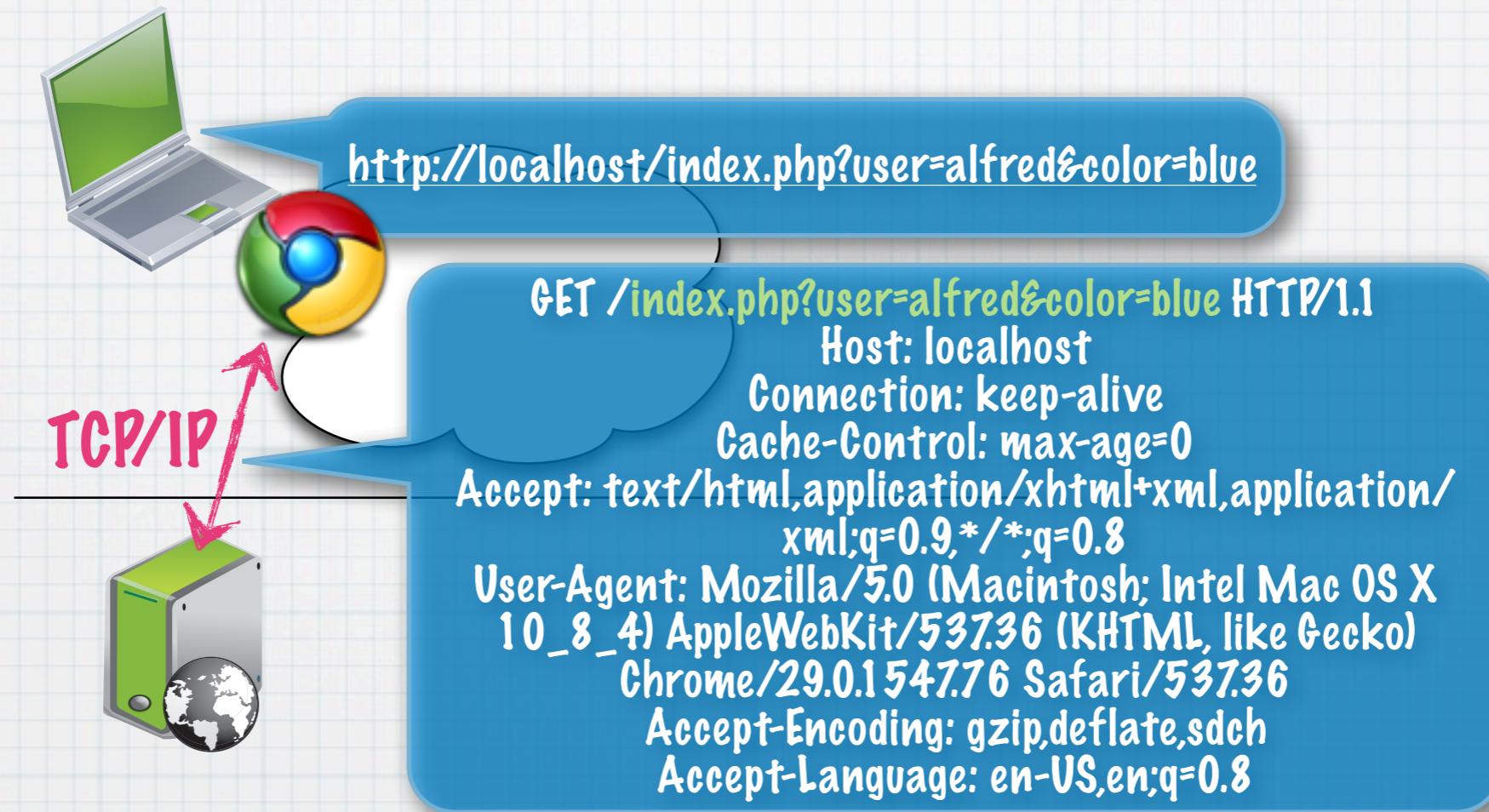
# Input til PHP: HTTP request-string



# Input til PHP: HTTP request-string



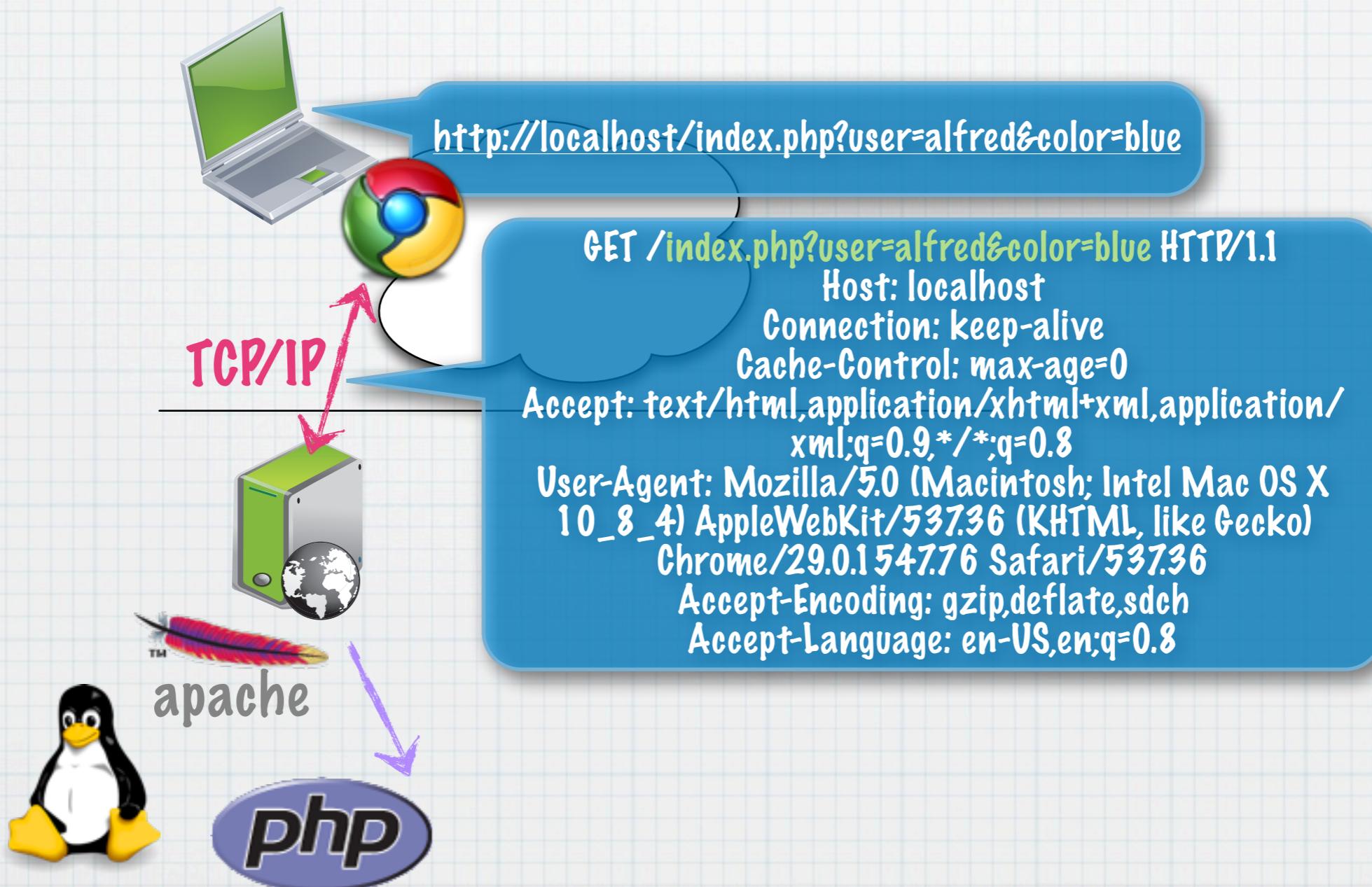
# Input til PHP: HTTP request-string



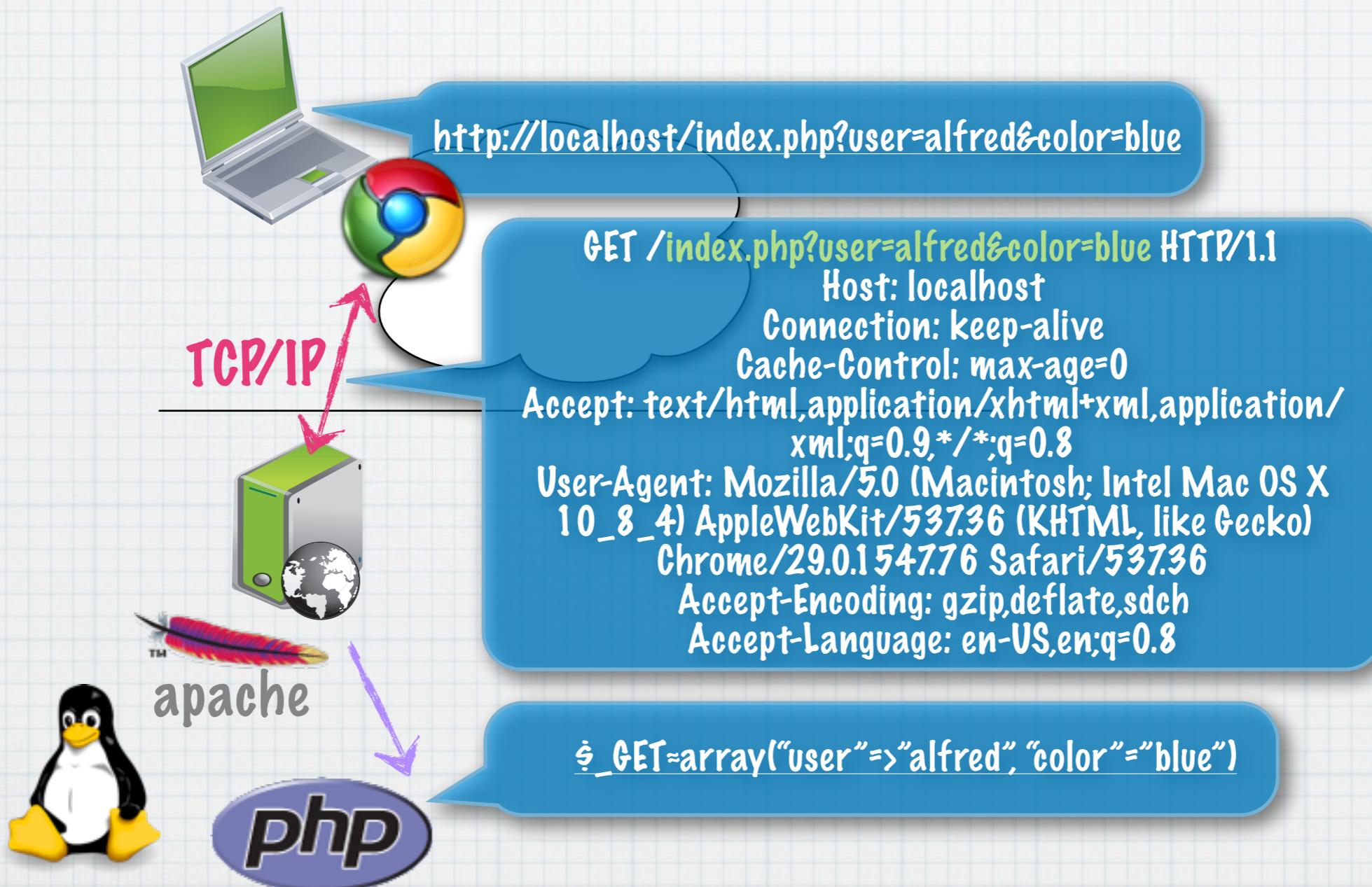
# Input til PHP: HTTP request-string



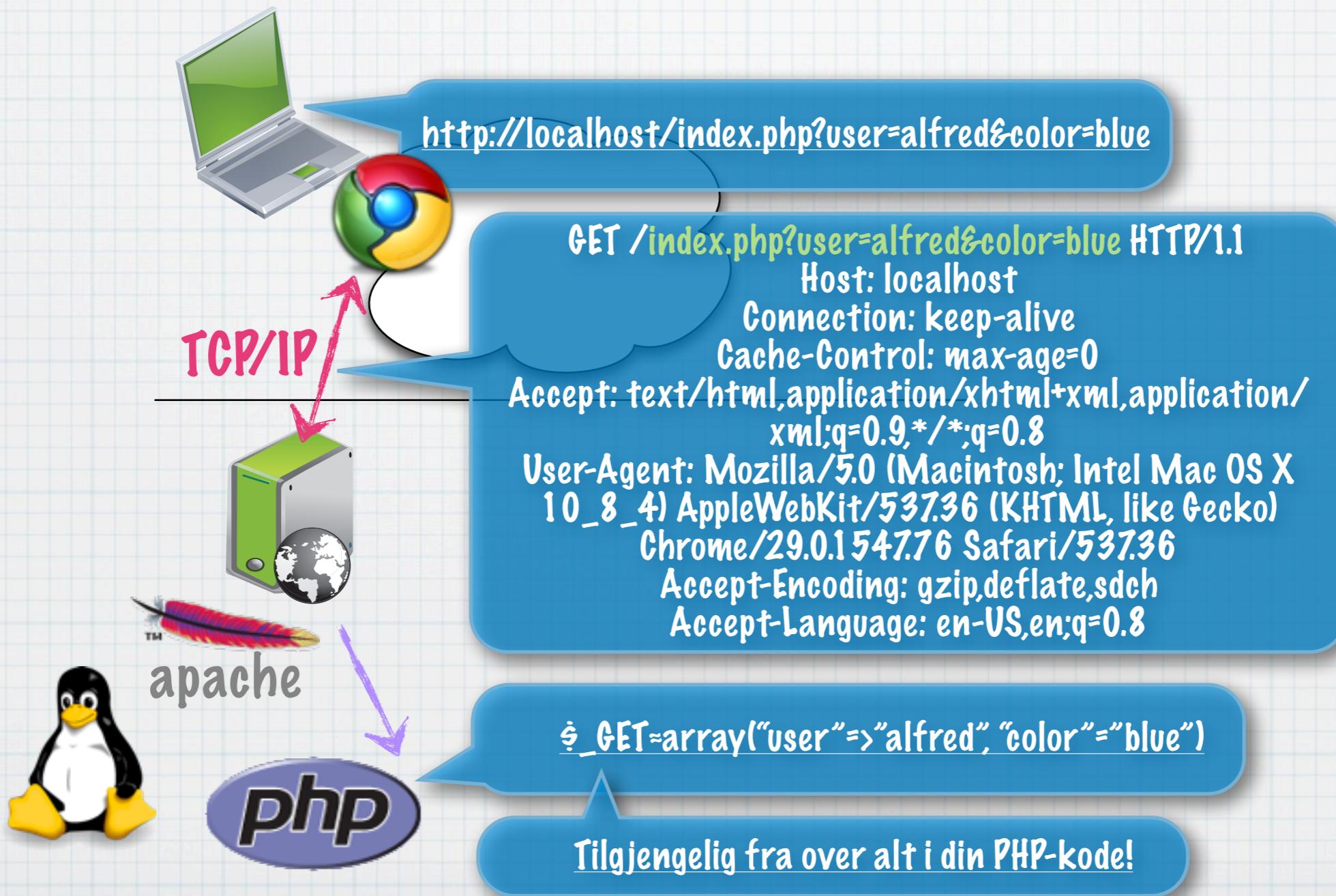
# Input til PHP: HTTP request-string



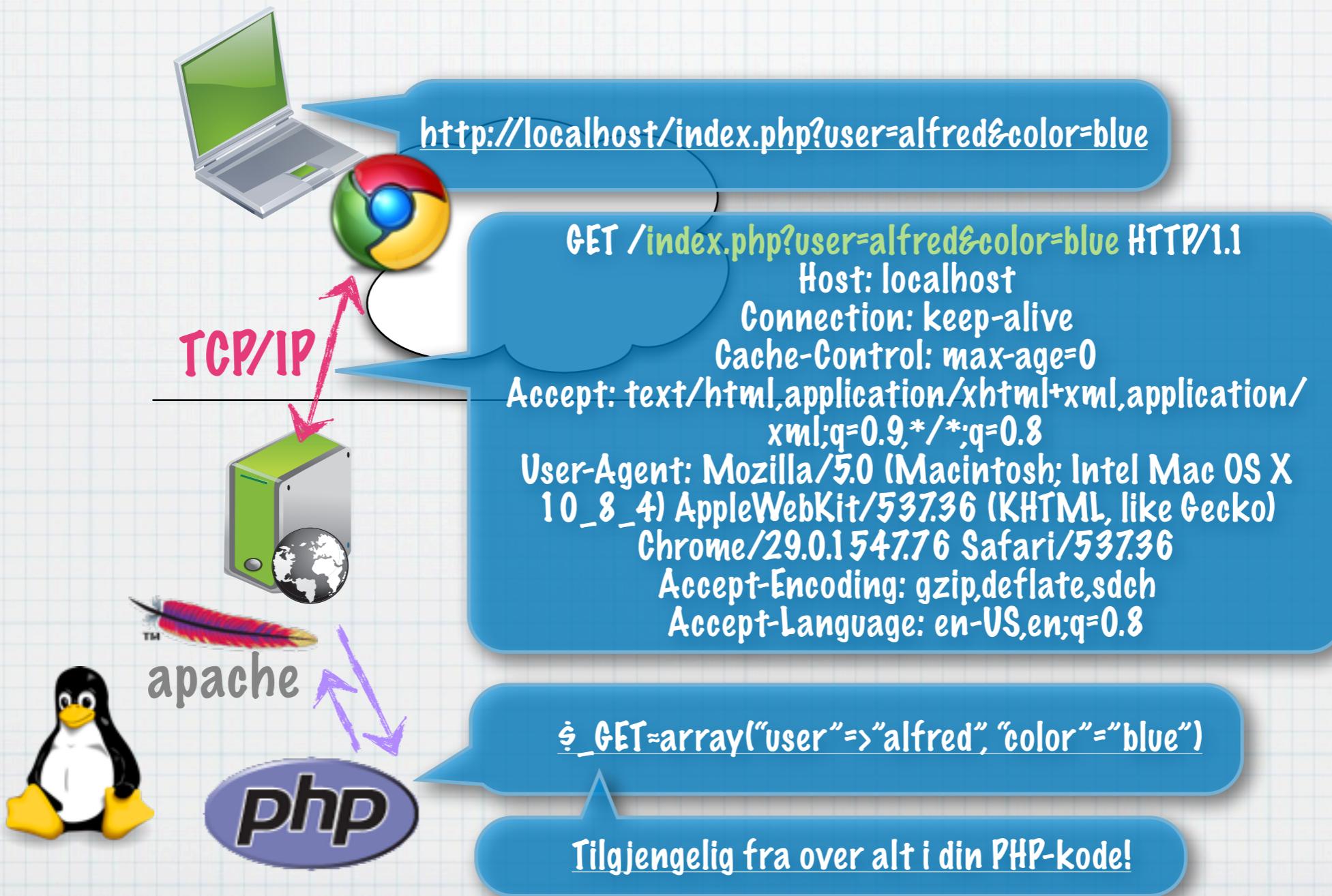
# Input til PHP: HTTP request-string



# Input til PHP: HTTP request-string



# Input til PHP: HTTP request-string



# Demo

---

variables.php

# PHP Basics: Kontrollstrukturer



A screenshot of a code editor window titled "control\_structures.html". The code is a PHP script with line numbers 1 through 22. It defines two variables (\$name1 and \$name2), two functions (getLastName and welcomeUser), and includes a die statement. The code uses global variables, concatenation, and a for loop. The code editor has a dark theme and shows "Line 22, Column 3" and "Tab Size: 4" at the bottom.

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10 }
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19     }
20
21 welcomeUser();
22 ?>
```

# PHP Basics: Kontrollstrukturer

- \* Hva er en kontrollstruktur?



A screenshot of a code editor window titled "control\_structures.html". The code is a PHP script with the following content:

```
<?php  
$name1="Linus";  
$name2="Knausgård";  
  
function getLastName(){  
    global $name2;  
    $name2="Carlsen";  
    $name2=$_GET["name"];  
}  
  
function welcomeUser(){  
    global $name1,$name2;  
    getLastName();  
    $name2=="Thorvalds" or die("Feil!");  
    $name=$name1." ".$name2;  
    for($i=0; $i<500; $i++)  
        echo($name." Rocks");  
}  
  
welcomeUser();  
?>
```

The code editor shows line numbers 1 through 22. The status bar at the bottom indicates "Line 22, Column 3" and "Tab Size: 4".

# PHP Basics: Kontrollstrukturer

- \* Hva er en kontrollstruktur?
- \* Styring av "programflyt"

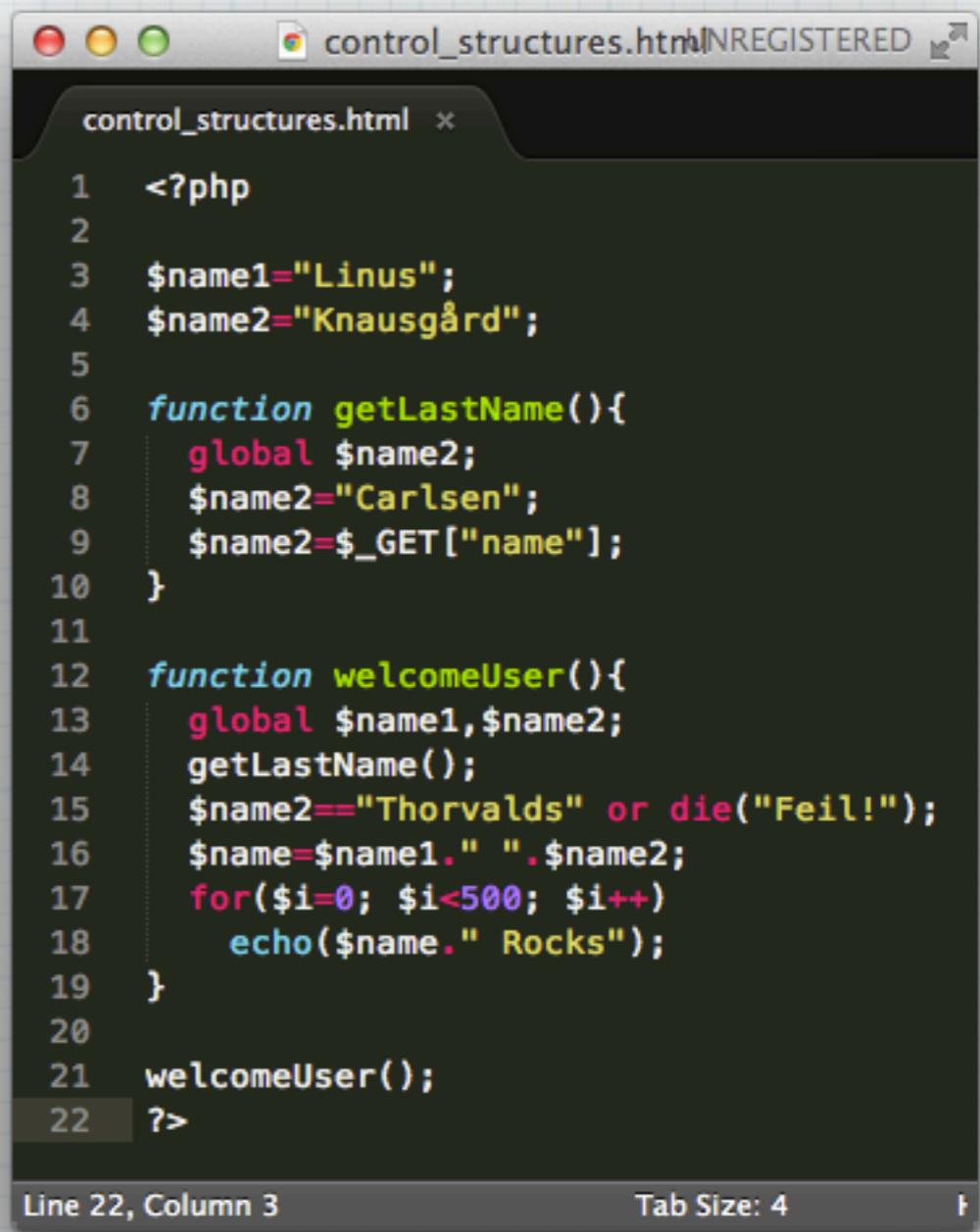


A screenshot of a code editor window titled "control\_structures.html". The code is written in PHP and defines two functions: `getLastName()` and `welcomeUser()`. The `getLastName()` function sets global variables `$name1` and `$name2` to "Linus" and "Knausgård" respectively. It also handles user input via `$_GET["name"]`. The `welcomeUser()` function retrieves the last name from `getLastName()`, checks if it's "Thorvalds" (using `==`), and if so, prints an error message. Otherwise, it concatenates the names and prints them 500 times using a `for` loop. Finally, it calls `welcomeUser()` and ends the script with `?>`. The code editor interface includes tabs for "control\_structures.html" and "NREGISTERED", status bars showing "Line 22, Column 3" and "Tab Size: 4", and standard OS X window controls.

```
<?php
$name1="Linus";
$name2="Knausgård";
function getLastName(){
    global $name2;
    $name2="Carlsen";
    $name2=$_GET["name"];
}
function welcomeUser(){
    global $name1,$name2;
    getLastName();
    $name2=="Thorvalds" or die("Feil!");
    $name=$name1." ".$name2;
    for($i=0; $i<500; $i++)
        echo($name." Rocks");
}
welcomeUser();
?>
```

# PHP Basics: Kontrollstrukturer

- \* Hva er en kontrollstruktur?
- \* Styring av "programflyt"
- \* Funksjoner



A screenshot of a code editor window titled "control\_structures.html". The code is written in PHP and defines two functions: `getLastName()` and `welcomeUser()`. The `getLastName()` function sets global variables `$name1` and `$name2` to "Linus" and "Knausgård" respectively. It also handles user input via `$_GET["name"]`. The `welcomeUser()` function retrieves the last name from `getLastName()`, checks if it's "Thorvalds", and die("Feil!") if not. It then creates a string `$name` by concatenating `$name1` and `$name2`, and loops 500 times to echo `$name` followed by " Rocks". The code ends with a closing PHP tag `?>`.

```
<?php
$name1="Linus";
$name2="Knausgård";
function getLastName(){
    global $name2;
    $name2="Carlsen";
    $name2=$_GET["name"];
}
function welcomeUser(){
    global $name1,$name2;
    getLastName();
    $name2=="Thorvalds" or die("Feil!");
    $name=$name1." ".$name2;
    for($i=0; $i<500; $i++)
        echo($name." Rocks");
}
welcomeUser();
?>
```

# PHP Basics: Kontrollstrukturer

- \* Hva er en kontrollstruktur?
- \* Styring av "programflyt"
- \* Funksjoner
- \* Løkker  
for, while, foreach



A screenshot of a code editor window titled "control\_structures.html". The code is written in PHP and defines two functions: `getLastName()` and `welcomeUser()`. The `getLastName()` function sets global variables `$name1` and `$name2` to "Linus" and "Knausgård" respectively. It also handles a user input variable `$name2` from the `$_GET["name"]` array. The `welcomeUser()` function retrieves the last names, checks if `$name2` is "Thorvalds", and die("Feil!") if it is not. It then creates a string `$name` by concatenating `$name1` and `$name2`, and uses a `for` loop to echo `$name` followed by " Rocks" 500 times. Finally, it calls `welcomeUser()` and ends the script with a closing tag. The code editor shows line numbers 1 through 22 and a tab size of 4.

```
<?php
$name1="Linus";
$name2="Knausgård";
function getLastName(){
    global $name2;
    $name2="Carlsen";
    $name2=$_GET["name"];
}
function welcomeUser(){
    global $name1,$name2;
    getLastName();
    $name2=="Thorvalds" or die("Feil!");
    $name=$name1." ".$name2;
    for($i=0; $i<500; $i++)
        echo($name." Rocks");
}
welcomeUser();
?>
```

# PHP Basics: Kontrollstrukturer

- \* Hva er en kontrollstruktur?
- \* Styring av "programflyt"
- \* Funksjoner
- \* Løkker  
for, while, foreach
- \* Kondisjonaler  
if-else-elseif



A screenshot of a code editor window titled "control\_structures.html". The code is written in PHP and defines two functions: `getLastName()` and `welcomeUser()`. The `getLastName()` function sets global variables `$name1` and `$name2` to "Linus" and "Knausgård" respectively, or retrieves `$name2` from the `$_GET["name"]` query parameter. The `welcomeUser()` function calls `getLastName()`, checks if `$name2` is "Thorvalds", and if so, die with an error message. It then constructs a name string by concatenating `$name1` and `$name2`, and prints it 500 times using a `for` loop. Finally, it calls `welcomeUser()` and ends the script.

```
<?php
$name1="Linus";
$name2="Knausgård";

function getLastName(){
    global $name2;
    $name2="Carlsen";
    $name2=$_GET["name"];
}

function welcomeUser(){
    global $name1,$name2;
    getLastName();
    $name2=="Thorvalds" or die("Feil!");
    $name=$name1." ".$name2;
    for($i=0; $i<500; $i++)
        echo($name." Rocks");
}
welcomeUser();
?>
```

# PHP Basics: Kontrollstrukturer

```
script4.js
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus...",name2);
7 }
8
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15
16    var name=name1+" "+name2;
17    for(var i=0; i<500; i++)
18        document.writeln(name+" Rocks");
19 }
20

control_structures.html
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10 }
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19 }
20
21 welcomeUser();
22 ?>
```

# PHP Basics: Kontrollstrukturer

Javascript kan be  
øst om et vindu

```
1 var name1="Linus";
2
3 name2="Knausgård";
4
5 function getLastName(){
6     name2="Carlsen";
7     name2>window.prompt("Linus...",name2);
8 }
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15
16    var name=name1+" "+name2;
17    for(var i=0; i<500; i++)
18        document.writeln(name+" Rocks");
19 }
20
```

Line 9, Column 24

Spaces: 4

JavaScript

```
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10 }
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19 }
20
21 welcomeUser();
22 ?>
```

Line 22, Column 3

Tab Size: 4

# PHP Basics: Kontrollstrukturer

Javascript kan be  
øS om et vindu

```
* 1 var name1="Linus";
* 2
* 3     name2="Carlsen";
* 4     name2>window.prompt("Linus...",name1);
* 5 }
* 6
* 7 function welcomeUser(){
* 8     getLastName();
* 9     if(name2!="Thorvalds"){
*10         alert("Feil!");
*11         welcomeUser();
*12     }
*13
*14     var name=name1+" "+name2;
*15     for(var i=0; i<500; i++)
*16         document.writeln(name+" Rocks");
*17
*18 }
*19
*20
```

Line 9, Column 24

Spaces: 4

JavaScript

PHP kan bare  
svare eller  
“avbryte sedning”

```
1 <?php
2
3     $name1="Linus";
4     $name2="Knausgård";
5
6     function welcomeUser(){
7         global $name1,$name2;
8         $name2="Carlsen";
9         $name2=$_GET["name"];
10    }
11
12    function welcomeUser(){
13        global $name1,$name2;
14        getLastName();
15        $name2=="Thorvalds" or die("Feil!");
16        $name=$name1." ".$name2;
17        for($i=0; $i<500; $i++)
18            echo($name." Rocks");
19    }
20
21    welcomeUser();
22 ?>
```

Line 22, Column 3

Tab Size: 4

# PHP Basics: Kontrollstrukturer

Javascript kan be  
øs om et vindu

I javascript kan vi  
bare "prøve igjen"

```
* 1 var name1="Linn";
* 2
* 3 function getLastName(){
* 4     name2="Carlsen";
* 5     name=window.prompt("Hva heter du? ");
* 6     if(name=="Thorvalds"){
* 7         alert("Feil!");
* 8         welcomeUser();
* 9     }
*10
*11     var name=name1+" "+name2;
*12     for(var i=0; i<500; i++)
*13         document.writeln(name+" Rocks");
*14
*15 }
```

Line 9, Column 24

Spaces: 4

JavaScript

PHP kan bare  
svare eller  
"avbryte sedning"

```
UNREGISTERED control_structures.htm REGISTERED
1 <?php
2
3 $name1="Linn";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19
20
21 welcomeUser();
22 ?>
```

Line 22, Column 3

Tab Size: 4

# PHP Basics: Kontrollstrukturer

Javascript kan be  
øs om et vindu

I javascript kan vi  
bare "prøve igjen"

```
* 1 var name1="Linus";
* 2     name2="Knausgård";
* 3
* 4 function getLastname(){
* 5     name="Carlsen";
* 6     name>window.prompt("Linus ...",name);
* 7
* 8 function welcomeUser(){
* 9     if(name2=="Thorvalds"){
*10         alert("Feil!");
*11         welcomeUser();
*12     }
*13
*14     var name=name1+" "+name2;
*15     for(var i=0; i<500; i++)
*16         document.writeln(name+" Rocks");
*17
*18 }
*19
*20
*21
*22
```

Line 9, Column 24

Spaces: 4

JavaScript

Line 22, Column 3

Tab Size: 4

PHP kan bare  
svare eller  
"avbryte sedning"

I PHP?

```
UNREGISTERED control_structures.htm REGISTERED
1 <?php
2
3     $name2="Knausgård";
4
5     function getLastname(){
6         global $name2;
7         $name2="Carlsen";
8
9     if($_GET["name"]){
10
11         function welcomeUser(){
12             global $name1,$name2;
13             getLastname();
14             $name2=="Thorvalds" or die("Feil!");
15             $name=$name1." ".$name2;
16             for($i=0; $i<500; $i++)
17                 echo($name." Rocks");
18
19         }
20
21         welcomeUser();
22     ?>
```

# PHP Basics: Kontrollstrukturer

I javascript kan vi  
bare “prøve igjen”

```
script4.js
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus...","name");
7 }
8 function welcomeUser(){
9     if(name2=="Thorvalds"){
10         alert("Feil!");
11     }
12     welcomeUser();
13 }
14
15 var name=name1+" "+name2;
16 for(var i=0; i<500; i++)
17     document.writeln(name+" Rocks");
18 }
19
20
21
22
```

Line 9, Column 24      Spaces: 4      JavaScript

```
control_structures.htm
1 <?php
2
3     $name2="Knausgård";
4
5     function getLastName(){
6         global $name2;
7         $name2="Carlsen";
8         if($_GET["name"]){
9             echo $_GET["name"];
10        }
11    }
12
13    function welcomeUser(){
14        global $name1,$name2;
15        getLastName();
16        $name2=="Thorvalds" or die("Feil!");
17        $name=$name1." ".$name2;
18        for($i=0; $i<500; $i++)
19            echo($name." Rocks");
20
21        welcomeUser();
22    ?>
```

Line 22, Column 3      Tab Size: 4

PHP kan bare  
svare eller  
“avbryte sedning”

I PHP?

# PHP Basics: Kontrollstrukturer

```
* script4.js UNREGISTERED
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus..."),name2);
7
8 function welcomeUser(){
9     if(name2=="Thorvalds"){
10         alert("Feil!");
11         welcomeUser();
12     }
13     var name=name1+" "+name2;
14     for(var i=0; i<500; i++)
15         document.writeln(name+" Rocks");
16 }
17
18
19 }
20

Line 9, Column 24          Spaces: 4          JavaScript
```

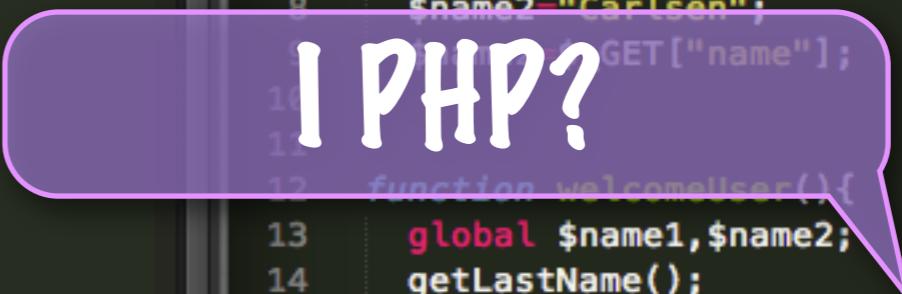
```
* control_structures.html UNREGISTERED
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     if($_GET["name"]){
10         welcomeUser();
11     }
12     global $name1,$name2;
13     getLastName();
14     $name2=="Thorvalds" or die("Feil!");
15     $name=$name1." ".$name2;
16     for($i=0; $i<500; $i++)
17         echo($name." Rocks");
18     }
19     welcomeUser();
20 ?>

Line 22, Column 3          Tab Size: 4
```

I javascript kan vi bare "prøve igjen"

I PHP?

# PHP Basics: Kontrollstrukturer



```
* script4.js UNREGISTERED
script4.js
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus...",name2);
7 }
8
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15
16    var name=name1+" "+name2;
17    for(var i=0; i<500; i++)
18        document.writeln(name+" Rocks");
19 }
20

Line 9, Column 24          Spaces: 4          JavaScript

control_structures.html UNREGISTERED
control_structures.html *
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     if($_GET["name"]){
10        getLastName();
11    }
12    function welcomeUser(){
13        global $name1,$name2;
14        getLastName();
15        $name2=="Thorvalds" or die("Feil!");
16        $name=$name1." ".$name2;
17        for($i=0; $i<500; $i++)
18            echo($name." Rocks");
19    }
20
21 welcomeUser();
22 ?>

Line 22, Column 3          Tab Size: 4
```

# PHP Basics: Kontrollstrukturer

```
script4.js
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus...",name2);
7 }
8
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15
16    var name=name1+" "+name2;
17    for(var i=0; i<500; i++)
18        document.writeln(name+" Rocks");
19 }
20

control_structures.html
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10 }
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19 }
20
21 welcomeUser();
22 ?>
```

# PHP Basics: Kontrollstrukturer



```
* script4.js UNREGISTERED
script4.js
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2>window.prompt("Linus...",name2);
7 }
8
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15
16    var name=name1+" "+name2;
17    for(var i=0; i<500; i++){
18        echo($name." Rocks");
19    }
20
21    welcomeUser();
22 }

control_structures.html UNREGISTERED
control_structures.html *
1 <?php
2
3 $name1="Linus";
4 $name2="Knausgård";
5
6 function getLastName(){
7     global $name2;
8     $name2="Carlsen";
9     $name2=$_GET["name"];
10 }
11
12 function welcomeUser(){
13     global $name1,$name2;
14     getLastName();
15     $name2=="Thorvalds" or die("Feil!");
16     $name=$name1." ".$name2;
17     for($i=0; $i<500; $i++)
18         echo($name." Rocks");
19
20     welcomeUser();
21
22 ?>
```

welcomeUser i JS  
"tar kontakt" med  
bruker via OS

# PHP Basics: Kontrollstrukturen

The screenshot shows a Mac OS X window titled "script4.js". The window contains a portion of a JavaScript script. The code is as follows:

```
1 var name1="Linus";
2 var name2="Knausgård";
3
4 function getLastName(){
5     name2="Carlsen";
6     name2=window.prompt("Linus...",name2);
7 }
8
9 function welcomeUser(){
10    getLastName();
11    if(name2!="Thorvalds"){
12        alert("Feil!");
13        welcomeUser();
14    }
15 }
```

# welcomeUser i JS

## “tar kontakt” med bruker via OS

```
control_structures.html *  
1 <?php  
2  
3 $name1="Linus";  
4 $name2="Knausgård";  
5  
6 function getLastName(){  
7     global $name2;  
8     $name2="Carlsen";  
9     $name2=$_GET["name"];  
10 }  
11  
12 function welcomeUser(){  
13     global $name1,$name2;  
14     getLastName();  
15     $name2--"Thorvalds" or die("Fail!");
```

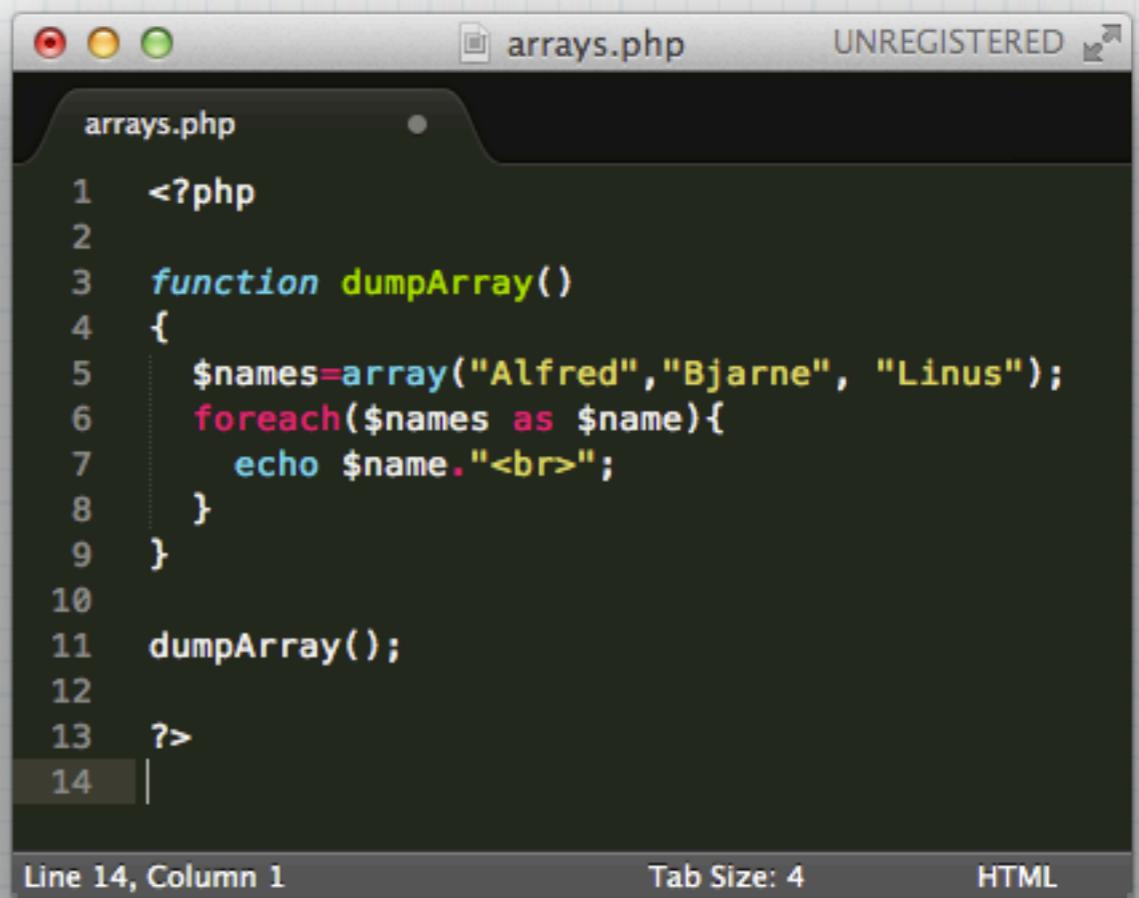
# PHP leser bare `$_GET`. Kan ikke “be om” data direkte

# Demo

---

control\_structures.php

# PHP Basics: Datastrukturer - Array



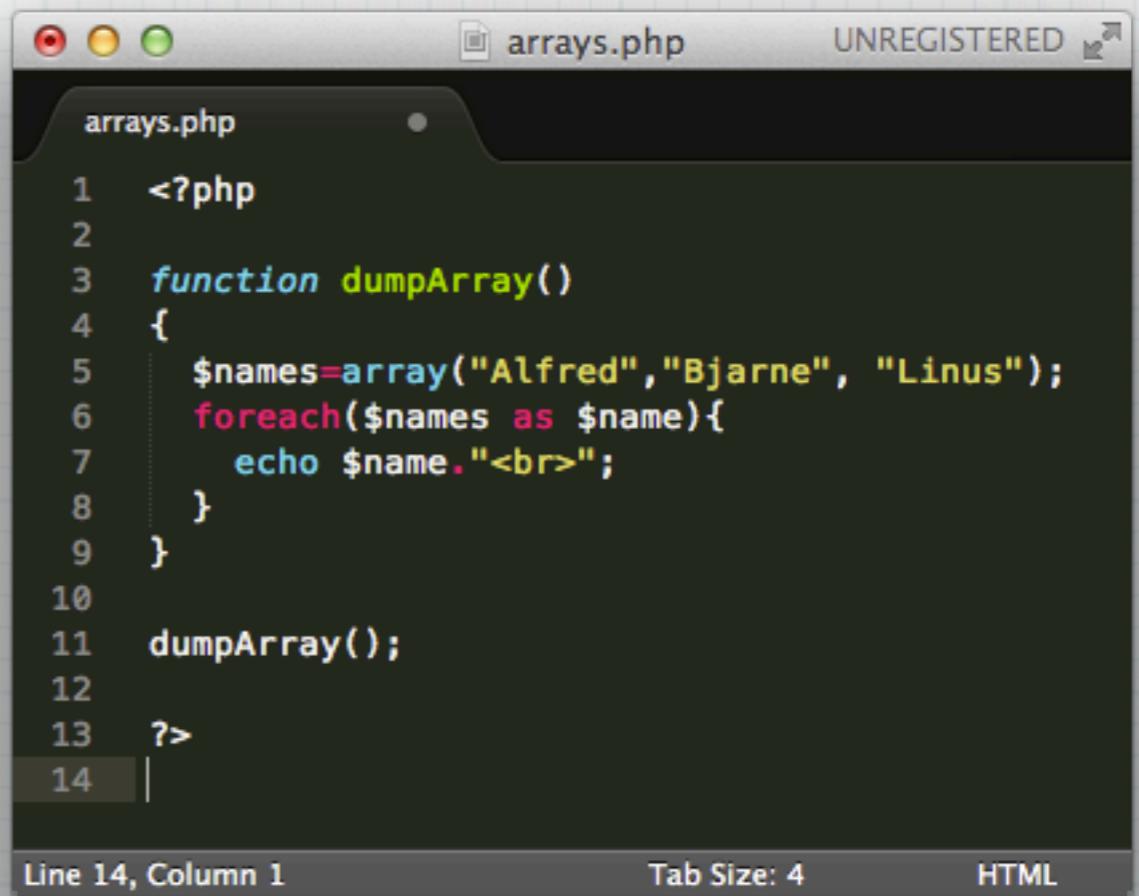
```
arrays.php
1 <?php
2
3 function dumpArray()
4 {
5     $names=array("Alfred","Bjarne", "Linus");
6     foreach($names as $name){
7         echo $name."<br>";
8     }
9 }
10
11 dumpArray();
12
13 ?>
14 |
```

arrays.php UNREGISTERED

Line 14, Column 1 Tab Size: 4 HTML

# PHP Basics: Datastrukturer - Array

- \* Et array er en indeksert liste av objekter (eller typer)

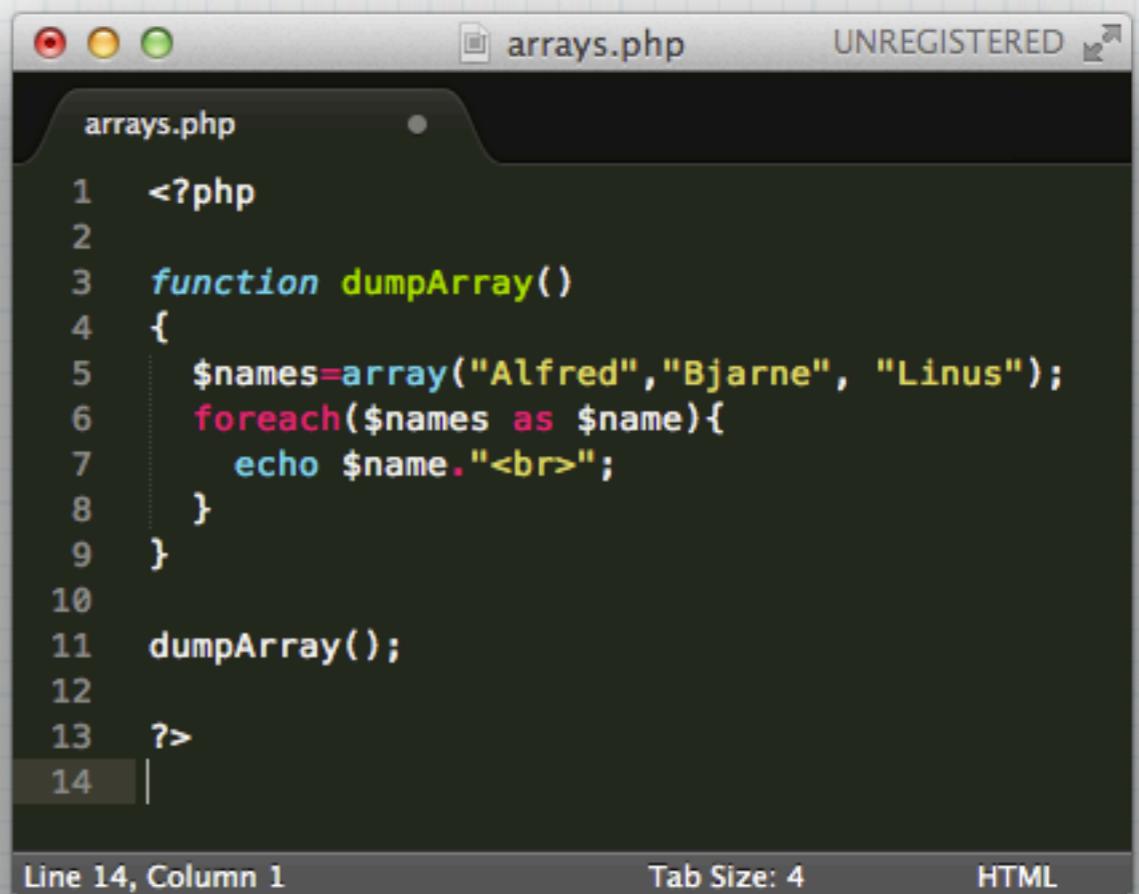


A screenshot of a code editor window titled "arrays.php". The code is a simple PHP script that defines a function called "dumpArray". This function creates an array of names ("Alfred", "Bjarne", "Linus") and then loops through the array, printing each name on a new line. The code editor interface includes a toolbar at the top, a status bar at the bottom with "Line 14, Column 1", "Tab Size: 4", and "HTML" buttons.

```
<?php
function dumpArray()
{
    $names=array("Alfred","Bjarne", "Linus");
    foreach($names as $name){
        echo $name."<br>";
    }
}
dumpArray();
?>
```

# PHP Basics: Datastrukturer - Array

- \* Et array er en indeksert liste av objekter (eller typer)
- \* Initialiseres slik:  
`$arr=array("a", "b", 1, 2);`

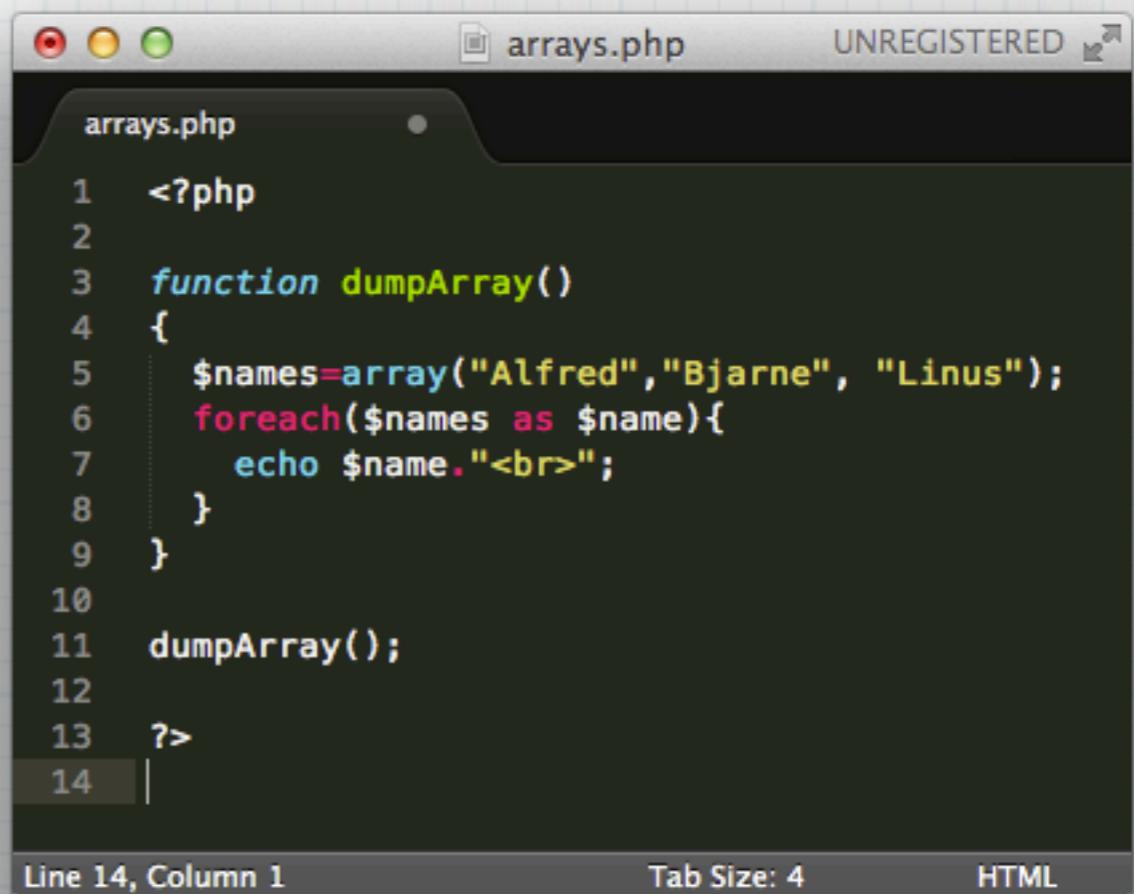


```
arrays.php
1 <?php
2
3     function dumpArray()
4     {
5         $names=array("Alfred","Bjarne", "Linus");
6         foreach($names as $name){
7             echo $name."<br>";
8         }
9     }
10    dumpArray();
11
12    ?>
13
14 |
```

Line 14, Column 1      Tab Size: 4      HTML

# PHP Basics: Datastrukturer - Array

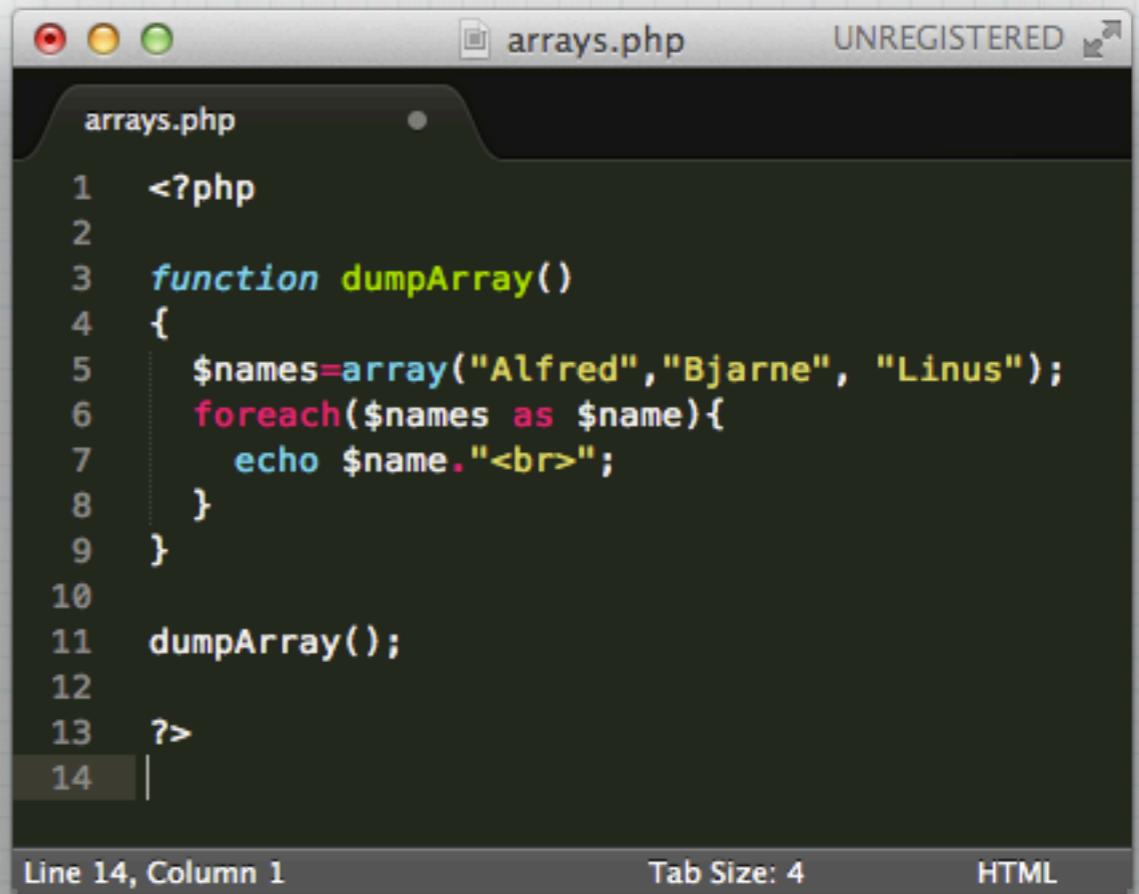
- \* Et array er en indeksert liste av objekter (eller typer)
- \* Initialiseres slik:  
`$arr=array("a", "b", 1, 2);`
- \* Itereres med vanlig løkke  
eller "foreach(\$arr as \$a)"



```
arrays.php
1 <?php
2
3 function dumpArray()
4 {
5     $names=array("Alfred", "Bjarne", "Linus");
6     foreach($names as $name){
7         echo $name."<br>";
8     }
9 }
10
11 dumpArray();
12
13 ?>
14 |
```

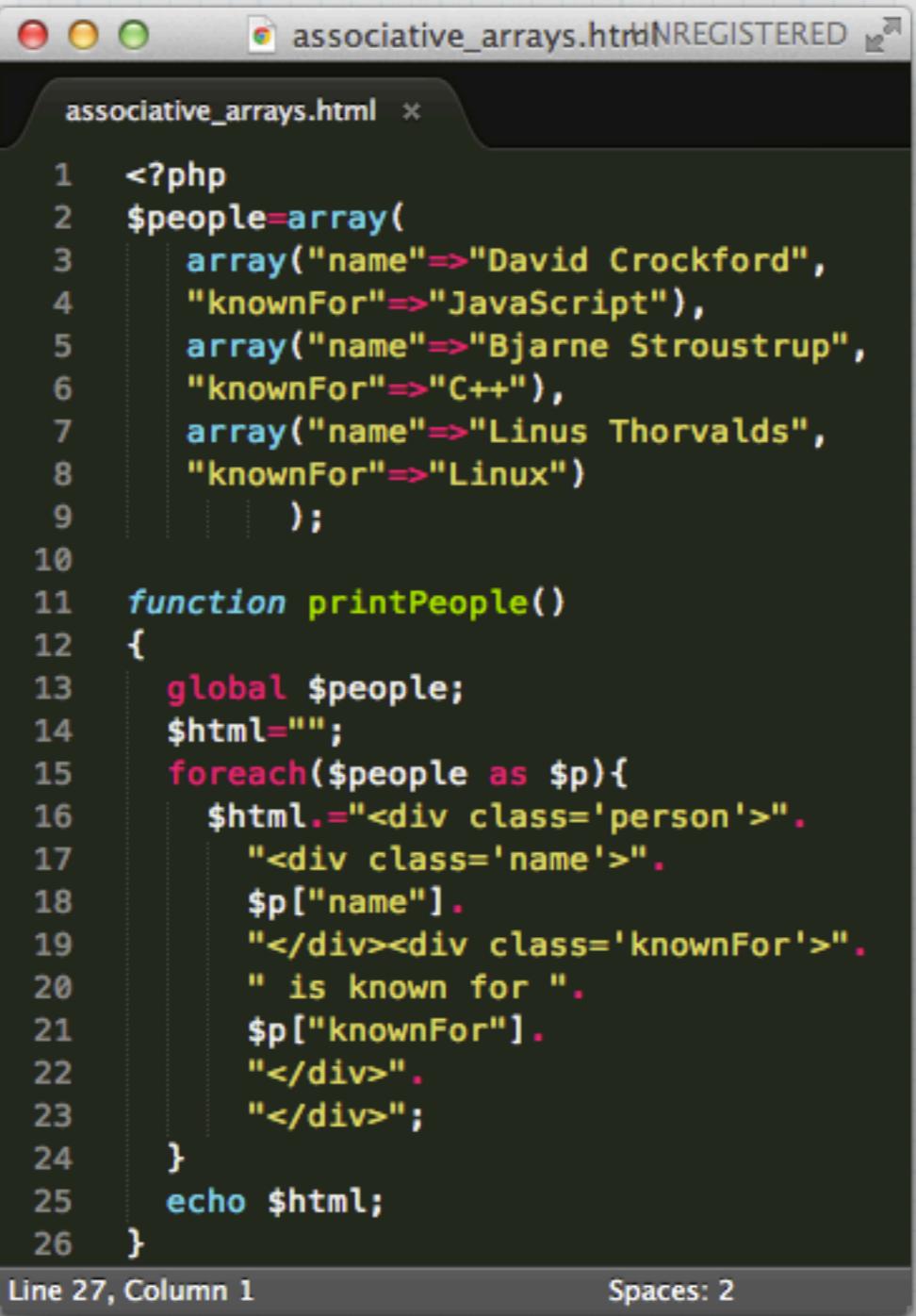
# PHP Basics: Datastrukturer - Array

- \* Et array er en indeksert liste av objekter (eller typer)
- \* Initialiseres slik:  
`$arr=array("a", "b", 1, 2);`
- \* Itereres med vanlig løkke eller "foreach(\$arr as \$a)"
- \* Aksesseres med "indeksoperator", arr[2]



```
<?php
function dumpArray()
{
    $names=array("Alfred", "Bjarne", "Linus");
    foreach($names as $name){
        echo $name."<br>";
    }
}
dumpArray();
?>
```

# PHP Basics: Associative arrays



A screenshot of a code editor window titled "associative\_arrays.html". The code is a PHP script that defines an associative array \$people containing three elements, each with a name and a list of things they are known for. It then defines a function printPeople() that iterates over the array and prints out HTML code for each person, including their name and the things they are known for.

```
1 <?php
2 $people=array(
3     array("name"=>"David Crockford",
4           "knownFor"=>"JavaScript"),
5     array("name"=>"Bjarne Stroustrup",
6           "knownFor"=>"C++"),
7     array("name"=>"Linus Thorvalds",
8           "knownFor"=>"Linux")
9 );
10
11 function printPeople()
12 {
13     global $people;
14     $html="";
15     foreach($people as $p){
16         $html.= "<div class='person'>" .
17             "<div class='name'>" .
18                 $p["name"] .
19             "</div><div class='knownFor'>" .
20                 " is known for " .
21                 $p["knownFor"] .
22             "</div>" .
23             "</div>";
24     }
25     echo $html;
26 }
```

Line 27, Column 1      Spaces: 2

# PHP Basics: Associative arrays

- \* Et assosiativt array er en ordnet liste av navngitte objekter. Som objekter i JS.



The screenshot shows a code editor window titled "associative\_arrays.html". The code is a PHP script that defines an associative array \$people containing three entries. Each entry is another array with "name" and "knownFor" keys. A function printPeople() is defined to output this data as HTML. The code editor has syntax highlighting and line numbers. The status bar at the bottom shows "Line 27, Column 1" and "Spaces: 2".

```
1 <?php
2 $people=array(
3     array("name"=>"David Crockford",
4           "knownFor"=>"JavaScript"),
5     array("name"=>"Bjarne Stroustrup",
6           "knownFor"=>"C++"),
7     array("name"=>"Linus Thorvalds",
8           "knownFor"=>"Linux")
9 );
10
11 function printPeople()
12 {
13     global $people;
14     $html="";
15     foreach($people as $p){
16         $html.= "<div class='person'>".
17               "<div class='name'>".
18               $p["name"].
19               "</div><div class='knownFor'>".
20               " is known for ".
21               $p["knownFor"].
22               "</div>".
23               "</div>";
24     }
25     echo $html;
26 }
```

Line 27, Column 1      Spaces: 2

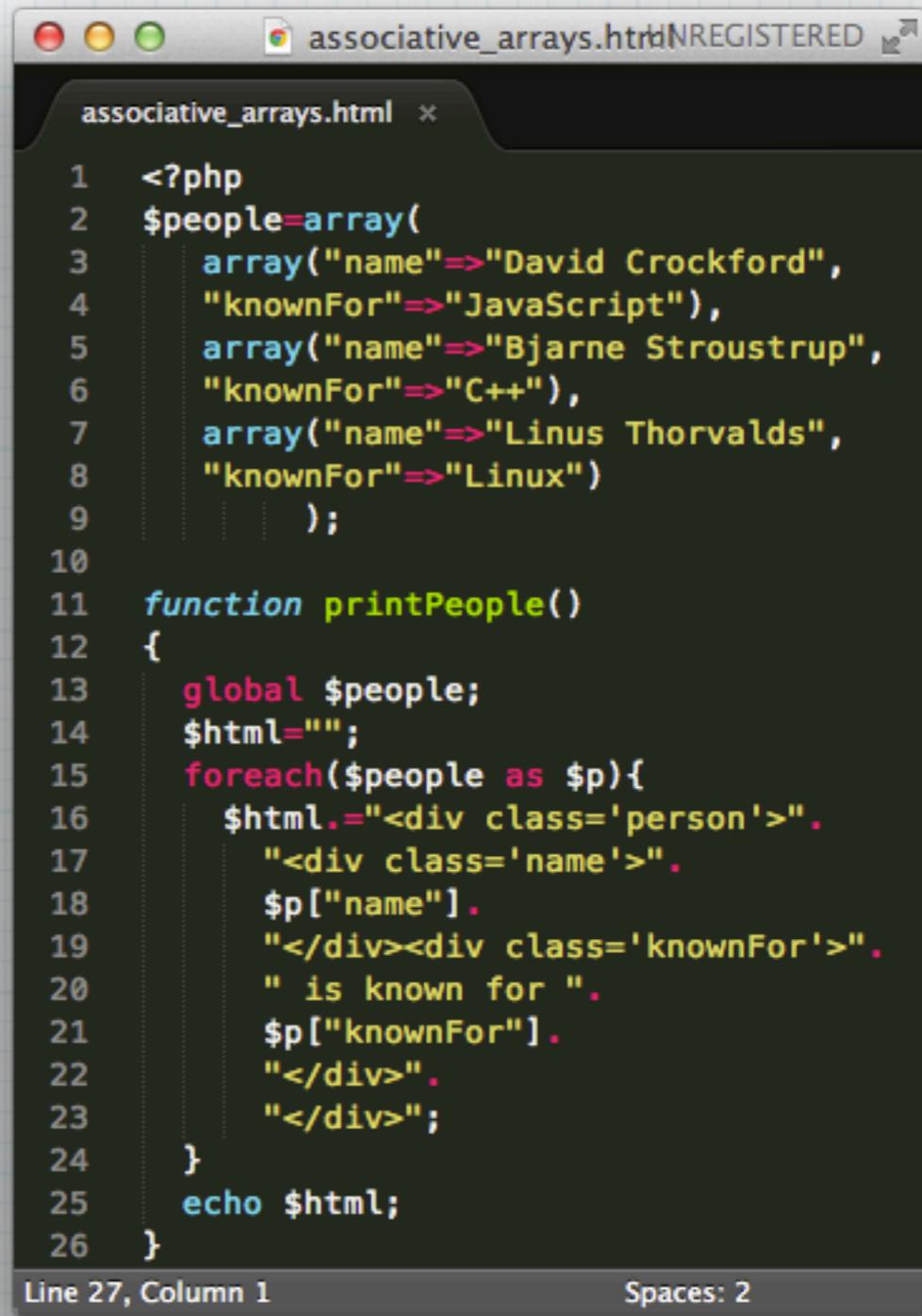
# PHP Basics: Associative arrays

- \* Et assosiativt array er en uordnet liste av navngitte objekter. Som objekter i JS.
  - \* Initialiseres slik:  
`obj=array("a"=>1,  
"b"=>array(1,2), "c"=>"d");`

```
associative_arrays.html *  
1  <?php  
2  $people=array(  
3      array("name"=>"David Crockford",  
4      "knownFor"=>"JavaScript"),  
5      array("name"=>"Bjarne Stroustrup",  
6      "knownFor"=>"C++"),  
7      array("name"=>"Linus Thorvalds",  
8      "knownFor"=>"Linux")  
9      );  
10  
11  function printPeople()  
12  {  
13      global $people;  
14      $html="";  
15      foreach($people as $p){  
16          $html.= "<div class='person'>".  
17              "<div class='name'>".  
18                  $p["name"].  
19              "</div><div class='knownFor'>".  
20                  " is known for ".  
21                  $p["knownFor"].  
22              "</div>".  
23              "</div>";  
24      }  
25      echo $html;  
26  }
```

# PHP Basics: Associative arrays

- \* Et assosiativt array er en ordnet liste av navngitte objekter. Som objekter i JS.
- \* Initialiseres slik:  
`obj=array("a"=>1,  
"b"=>array(1,2), "c"=>"d");`
- \* Itereres med vanlig løkke eller "foreach"



The screenshot shows a code editor window titled "associative\_arrays.html". The code is a PHP script that defines an associative array \$people and a function printPeople(). The array contains three entries: David Crockford, Bjarne Stroustrup, and Linus Thorvalds, each associated with their knownFor projects. The printPeople() function iterates over the array, generating HTML output for each person's name and knownFor projects.

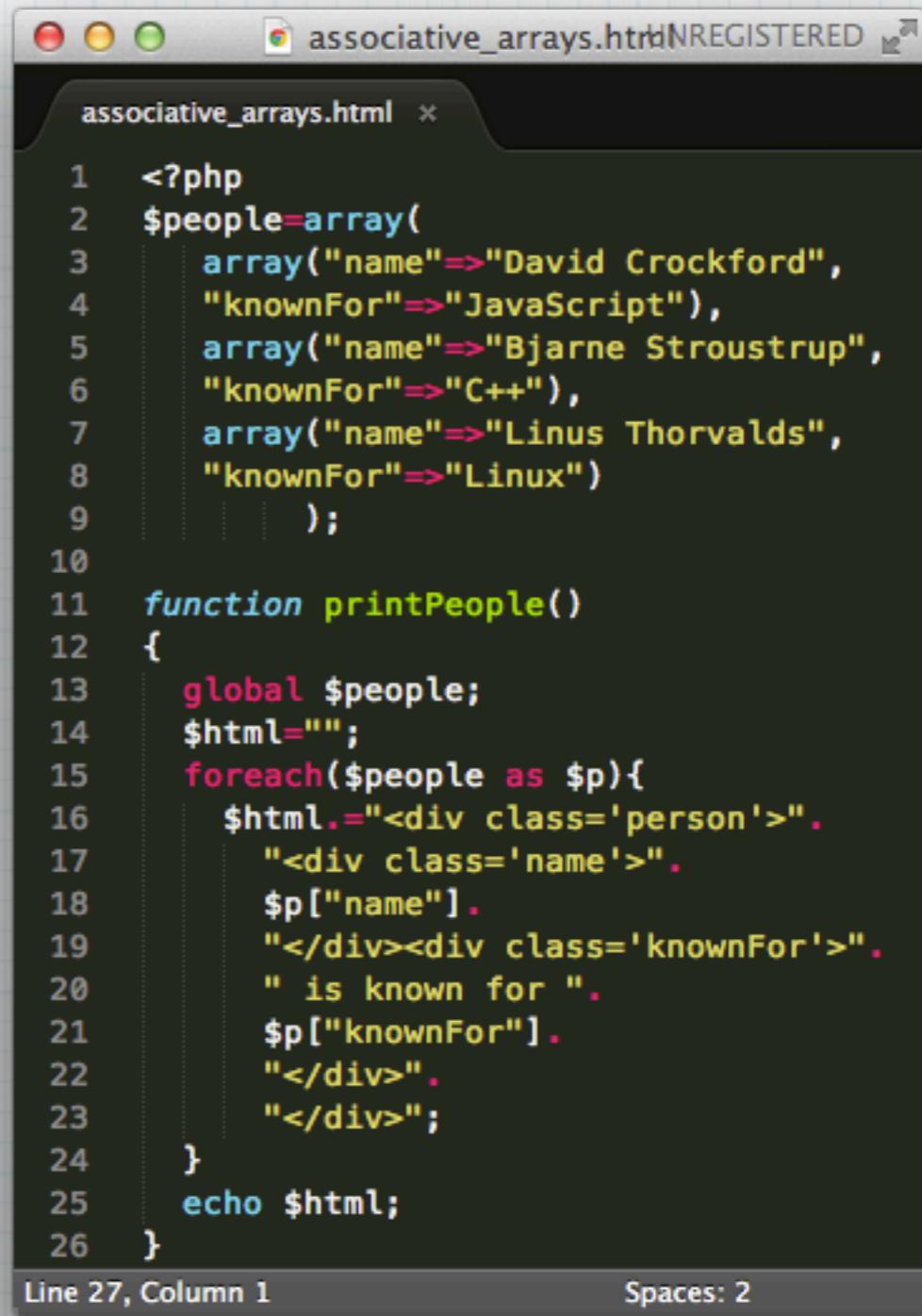
```
<?php
$people=array(
    array("name"=>"David Crockford",
        "knownFor"=>"JavaScript"),
    array("name"=>"Bjarne Stroustrup",
        "knownFor"=>"C++"),
    array("name"=>"Linus Thorvalds",
        "knownFor"=>"Linux")
);

function printPeople()
{
    global $people;
    $html="";
    foreach($people as $p){
        $html.= "<div class='person'>".
            "<div class='name'>". $p["name"] .
            "</div><div class='knownFor'>". " is known for " .
            $p["knownFor"] .
            "</div>". "</div>";
    }
    echo $html;
}
```

Line 27, Column 1      Spaces: 2

# PHP Basics: Associative arrays

- \* Et assosiativt array er en ordnet liste av navngitte objekter. Som objekter i JS.
- \* Initialiseres slik:  
`obj=array("a"=>1,  
"b"=>array(1,2), "c"=>"d");`
- \* Itereres med vanlig løkke eller "foreach"
- \* Aksesseres med "indeksoperator", `obj["b"]`



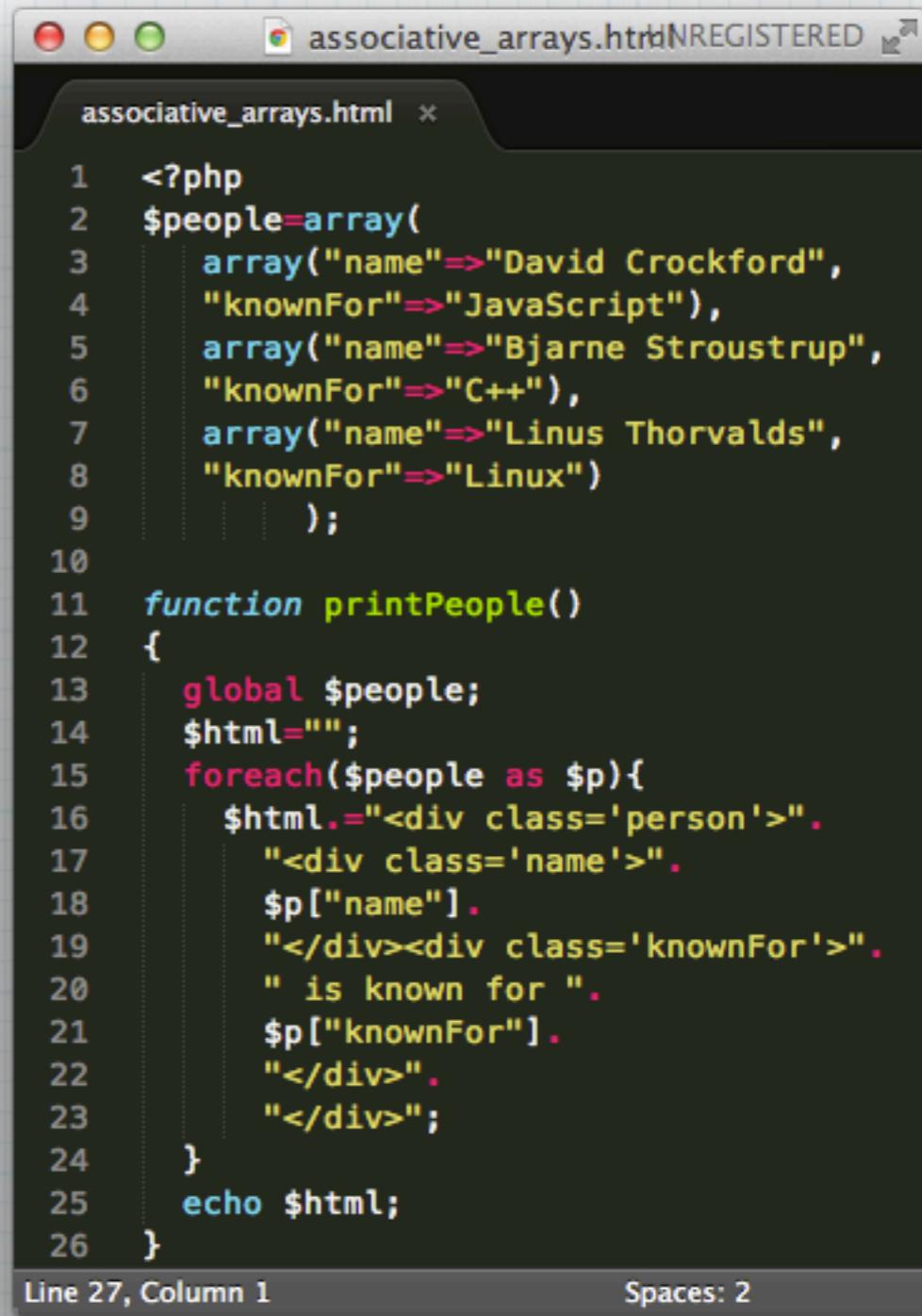
The screenshot shows a code editor window titled "associative\_arrays.html". The code is a PHP script that defines an associative array \$people and a function printPeople(). The array contains three entries: David Crockford, Bjarne Stroustrup, and Linus Thorvalds, each associated with their knownFor projects. The printPeople() function iterates over the array, creating an HTML string where each person's name and knownFor project are wrapped in div elements with classes 'name' and 'knownFor' respectively.

```
<?php  
$people=array(  
    array("name"=>"David Crockford",  
          "knownFor"=>"JavaScript"),  
    array("name"=>"Bjarne Stroustrup",  
          "knownFor"=>"C++"),  
    array("name"=>"Linus Thorvalds",  
          "knownFor"=>"Linux")  
);  
  
function printPeople()  
{  
    global $people;  
    $html="";  
    foreach($people as $p){  
        $html.= "<div class='person'>".  
               "<div class='name'>".  
               $p["name"].  
               "</div><div class='knownFor'>".  
               " is known for ".  
               $p["knownFor"].  
               "</div>".  
               "</div>";  
    }  
    echo $html;  
}
```

Line 27, Column 1      Spaces: 2

# PHP Basics: Associative arrays

- \* Et assosiativt array er en ordnet liste av navngitte objekter. Som objekter i JS.
- \* Initialiseres slik:  
`obj=array("a"=>1,  
"b"=>array(1,2), "c"=>"d");`
- \* Itereres med vanlig løkke eller "foreach"
- \* Aksesseres med "indeksoperator", `obj["b"]`
- \* Kompatibelt med JSON!



The screenshot shows a code editor window titled "associative\_arrays.html". The code is a PHP script that defines an associative array \$people containing three entries, each with a name and an array of knownFor values. It then defines a printPeople() function that iterates over the array and prints each person's name and their knownFor values. The code is syntax-highlighted, with PHP tags in blue, variables in green, and strings in yellow. The code editor interface includes a toolbar at the top and status bars at the bottom indicating the current line and column.

```
<?php  
$people=array(  
    array("name"=>"David Crockford",  
          "knownFor"=>"JavaScript"),  
    array("name"=>"Bjarne Stroustrup",  
          "knownFor"=>"C++"),  
    array("name"=>"Linus Thorvalds",  
          "knownFor"=>"Linux")  
);  
  
function printPeople()  
{  
    global $people;  
    $html="";  
    foreach($people as $p){  
        $html.= "<div class='person'>".  
               "<div class='name'>".  
               $p["name"].  
               "</div><div class='knownFor'>".  
               " is known for ".  
               $p["knownFor"].  
               "</div>".  
               "</div>";  
    }  
    echo $html;  
}
```

Line 27, Column 1      Spaces: 2

# Demo

---

arrays.html  
associative\_arrays.php / people.php