introduction to

SCRIPTING, DATABASES, SYSTEM ARCHITECTURE

PHP I: comments, variables, arithmetic, string op's, if statements, data collection

Claus Brabrand

(((brabrand@itu.dk)))

Associate Professor, Ph.D.

(((Software and Systems)))

IT University of Copenhagen

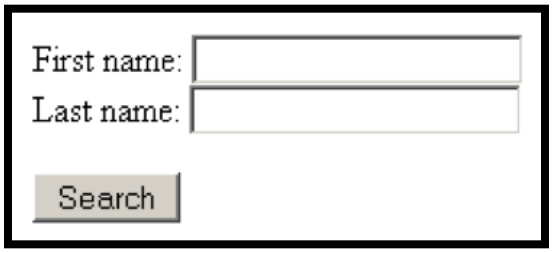
Agenda

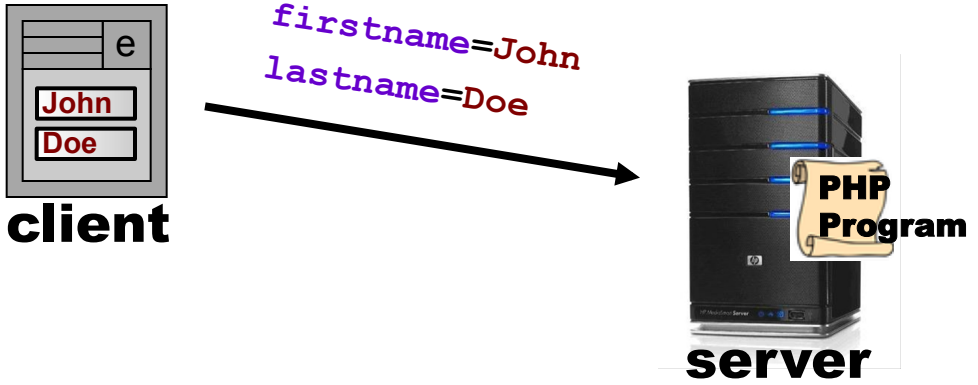
- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Form submission…

- Form submits info to server (PHP) program:

```
<form action="ADDRESS-OF-SCRIPT.php">
    First name:
    <input type="text" name="firstname"/> <br/>
    Last name:
    <input type="text" name="lastname"/>  <br/>
    </p>
    <input type="submit" value="Search"/>
</form>
```





<input ...

<input type="hidden"/>

<textarea>

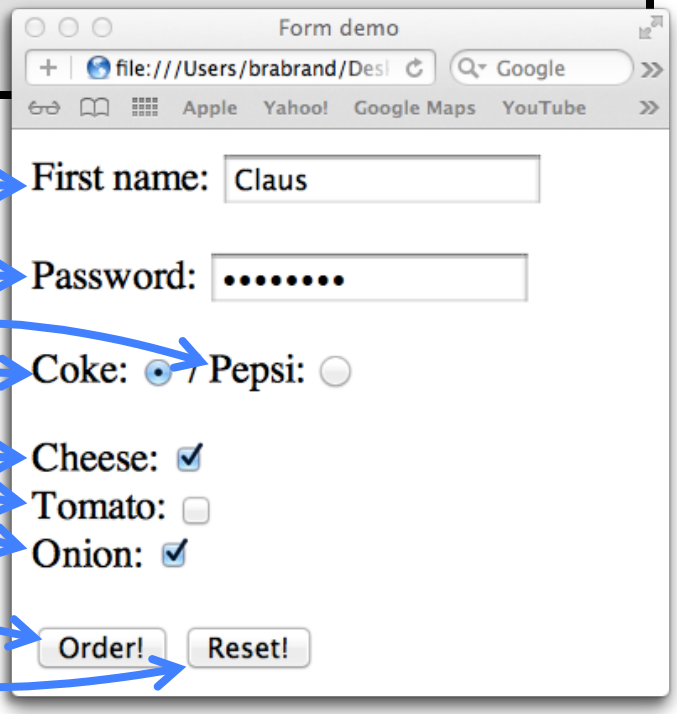
<select>

select-multiple

- Many "type"s of <input>:

```
<form action="address_of_script">
    <p>
    First name: <input type="text" name="firstname" /> <br />
    ...
</form>
```

- <input type="text" .../>
- <input type="password" .../>
- <input type="radio" .../>
- <input type="checkbox" .../>
- <input type="submit" .../>
- <input type="reset" .../>
- ...

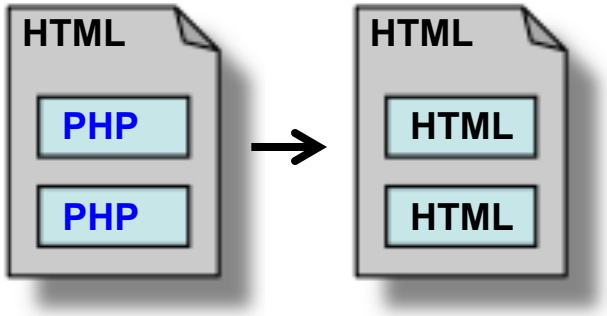


PHP Example: Hello World!

Abbreviation: in general, I'll omit <html> and <body>

```
<html>
    <body>
        <?php
            echo "Hello World!" ;
        ?>
    </body>
</html>
```

Hello World!




```
<?php
    echo "<h1>Hello World!</h1>" ;
    echo "<p/>" ;
    echo "This is <b>bold</b> and <i>italic</i>" ;
?>
```

=

```
<?php echo "<h1>Hello World!</h1>" ; ?>
<p/>
<?php echo "This is <b>bold</b> and <i>italic</i>" ; ?>
```

Hello World!

This is **bold** and *italic*

Variables and Arithmetic

```
<?php   $year = 2012 ;
        echo "The year is: " ;
        echo $year ;  ?>
```

The year is: 2012

```
<?php   $year = 2012 ;  ?>
The year is:
<?php   echo $year ;  ?>
```

The year is: 2012

```
<?php   $year = 2012 ;  ?>
Next year is:
<?php   echo ( $year + 1 ) ;  ?>
```

Next year is: 2013

```
<?php   $year = date("Y") ;  ?>
Next year is:
<?php   echo ( $year + 1 ) ;  ?>
```

Next year is: 2013

Agenda

- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Comments

- A ***comment*** is some text that explains what the code is doing **in "human language"** (for other programmers or oneself, later on)

Note: comments don't do anything!

- **Then; why bother with comments?!?:**
  - 1) Makes code easier to read and understand
  - 2) Helps others understand your code
  - 3) Helps yourself understand your code, later on

One-line + Multi-line comments

- One line comments:

```
<!-- this is an HTML comment -->
<?php
      $year = 2012 ;          // this is a one line PHP comment
      echo "The year is: " ;  # another one line PHP comment
      echo $year ;
?>
<!-- now, we're outside PHP, so we use HTML comments again -->
```

- Multi-line comments:

```
<?php
      $year = 2012 ; /* this is a multi-line comment which is
                        allowed to span more than one line */
      echo "The year is: " ;
      echo $year ;
?>
<!-- HTML comments may always
     span multiple lines... -->
```

Comments: Example

- An Example:

```
<?php
      /* This script will display a welcome
         message and the time of day */

      echo "<h1>Welcome</h1>" ; // display welcome message

      $time = date("h:m") ;      // get time as 'hours:minutes'
      echo "The time is now: "  // write out some text and
      echo $time ;              // write out the time
?>
```

Welcome

The time is now: 23:59

- Comments are ignored when PHP is executed:

```
<?php       echo "Welcome" ;
          // echo "to" ;
             echo "DSDS";      ?>
```

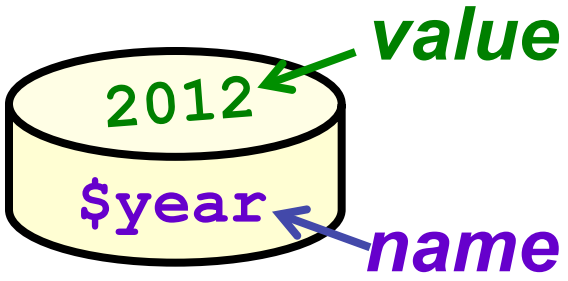
Welcome DSDS

Agenda

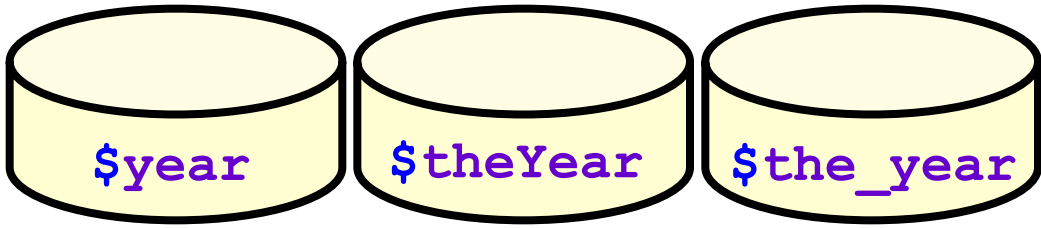
- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Variables

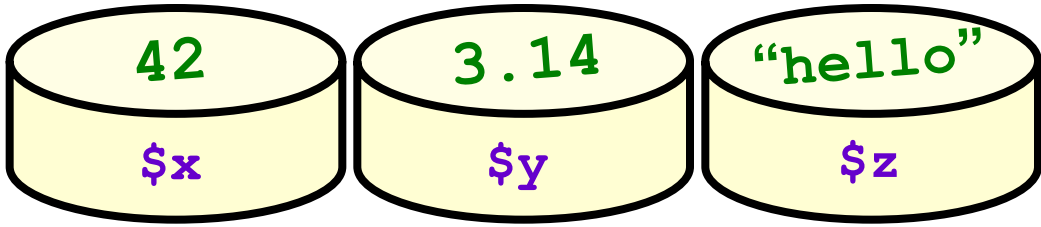
- A ***variable*** is a ***container*** with a ***name*** that can store a ***value***:



- **Name:**
  Starts with a dollar sign '**$**' followed by an ***identifier***



- **Value:**
  Can be a ***number*** or a ***string of text*** or (…)

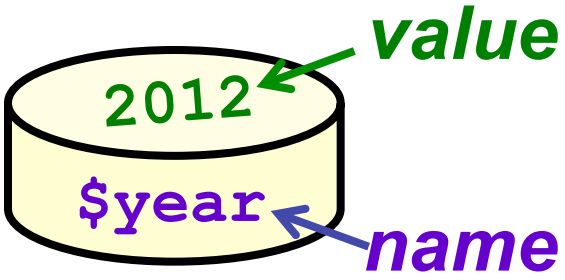


Variable Example

- You've already seen an example of variables:

```
<?php $year = 2012 ;
      echo "The year is:" ;
      echo $year ; ?>
```

The year is: 2012



- Var's can be used in subsequent PHP tags:

```
<?php $year = 2012 ; ?>
The year is:
<?php echo $year ; ?>
```

The year is: 2012

Variable Examples...

- Number Example:

```
<?php $year = 2012 ;
      echo "The year is:" ;
      echo $year ; ?>
```

The year is: 2012

- Decimal Example:

```
<?php $pi = 3.1415926 ;
      echo "<em>Pi</em> (&pi;) is:" ;
      echo $pi ; ?>
```

Pi (π) is: 3.1415926

- String (text) Example:

```
<?php echo "Hi, my name is " ;
      $x = "Claus Brabrand" ;
      echo $x ; ?>
```

Hi, my name is Claus Brabrand

A Larger Example

- We can use multiple variables:

```
<?php $name = "Barack Obama" ;
      $born = 1961 ;
      $country = "The United States of America" ;
      echo "Hello, my name is <b>" ;
      echo $name ;
      echo "</b>" ;
      echo "<p/>" ;
      echo "I am the president of: <br/>" ;
      echo "<em>" ;
      echo $country ;
      echo "</em>" ;
      echo "<p/>" ;
      echo "I am <b>" ;
      echo (2012 - $born) ;
      echo "</b> years old." ;
?>
```

Hello, my name is **Barack Obama**

I am the president of:
The United States of America

I am **51** years old.

Variables are *'case sensitive'*

- Variable names are ***case sensitive*** in PHP:

```
<?php $name = "Here is a some text" ;
      $Name = "Here's some other text" ;
      echo $name ;
      echo "<p/>" ;
      echo $Name ;
?>
```

```
Here is some text

Here's some other text
```

- Use **consistent** naming scheme for your vars:
  - `$first_name`
  - `$firstName`
  - `$FirstName`

Variable Assignment

- **Assignment:** `$year = 2012 ;`

- You can ***change*** the value of a variable:

```
<?php $year = 2012 ;
      echo "The year is: " ;
      echo $year ;

      echo "<p/>" ;

      $year = 1066 ; // re-assignment (value of $year changed)
      echo "The year is now: " ;
      echo $year ;
 ?>
```

The year is: 2012

The year is now: 1066

- "echo" will print the **current** value of a variable

Variable Types

Variables can contain different **types** of values:

- **Integers** (numbers):
  - …, -3, -2, -1, 0, 1, 2, 3, …
- **Floating points** (decimal numbers):
  - 1.5, 2.0, 0.0, 0.41, -123.456, 3.1415926, …
- **Strings** (pieces of text):
  - “ITU”, “42”, “hello!”, “Barack Obama”, “Urgh!@$!”, …
- **Booleans** (truth values):
  - *true*, *false* (there are only these two!)

Variable Types (continued)

- PHP is not very strict with variables:
  - We don't need to ***declare*** variables (define its type)
  - We don't need to ***initialize*** variables (set initial value)
- A variable can even have ***different types*** (of values) during execution:

```
<?php
      $x = 42 ;         // the value of 'x' is a number
      echo $x ;
      echo "<p/>" ;
      $x = "Hello!" ; // ...the value is now a string
      echo $x ;
      $x = 3.14 ;       // ...the value is now a decimal
      echo $x ;
?>
```

```
42
Hello!
3.14
```

Agenda

- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Strings

- A ***string*** is a ***sequence of characters*** enclosed in quotation marks (‘…’) or (“…”):

- **Single quotes**:
  - `'text', 'text with space', ' ', '42', 'in fact, any symbols %*!@', …`

- **Double quotes**:
  - `"text", "text with space", " ", "42", "in fact, any symbols %*!@", …`

Single vs Double Quotes

- Single quotes (‘…’) vs Double quotes (“…”):

- **Single quotes**:
  - We can ***not*** use variables inside:

```
<?php $year = 2012 ;
      echo 'The year is: $year' ;
?>
```

The year is: $year

- **Double quotes**:
  - We ***can*** use variables inside:

Many therefore prefer to use double quotes!

```
<?php $year = 2012 ;
      echo "The year is: $year" ;
?>
```

The year is: 2012

Character Escaping

- Common character escapings in text strings:
  - `\n` (newline character)
  - `\t` (tabulation character)
  - `\$` (dollar sign character)
  - `\'` (single quote character)
  - `\"` (double quote character)
  - `\\` (backslash character)
- Examples:

```
<pre>
    <?php echo "hello\nworld\t!" ; ?>
</pre>
```

```
Hello
world    !
```

```
<?php echo "He said \"You need this stuff!\" to the students" ; ?>
```

He said “You need this stuff!” to the students

You can mix quotes ‘ ” ’ or “ ‘ “

- Note: You do ***not*** have to escape...:
  - ...double quotes **inside** single quotes:

```
<?php echo 'Here: "double quotes" inside single quotes.' ; ?>
```

Here: “double quotes” inside single quotes.

  - ...single quotes **inside** double quotes:

```
<?php echo "Here: 'single quotes' inside double quotes." ; ?>
```

Here: ‘single quotes’ inside double quotes.

- However...:

```
<?php echo 'But \'same kind\' of quotes need escaping.' ; ?>
```

But ‘same kind’ of quotes need escaping.

```
<?php echo 'But \"same kind\" of quotes need escaping." ; ?>
```

But “same kind” of quotes need escaping.

String Concatenation

- String concatenation via the ‘ . ’ (dot) operator:
  - `( “string1” . “string2” )`

- Examples:

```
<?php echo ( “dan” . “mark” ) ; ?>
```

danmark

```
<?php $h = “Hello ” ;
      $w = “world!” ;

      $x = ( $h . $w ) ; // string concatenation!

      echo “<h1>$x</h1>” ;
?>
```

Hello world!

EXERCISE

1) Make the bottom PHP program work
2) Change it to echo "Obama, Barack"

- String concatenation via the '.' (dot) operator:
  - `( "string1" . "string2" )`

- Examples:

```
<?php echo ( "dan" . "mark" ) ; ?>
```

danmark

```
<html><body>
<?php $first = "Barack" ;
      $last = "Obama" ;

      $temp = ( $first . " " ) ; // concatenation!
      $name = ( $temp . $last ) ; // concatenation!
      echo "<h1>$name</h1>" ;
?>
</body></html>
```

Barack Obama

Agenda

- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Arithmetic

- You have already seen (simple) ***arithmetic***:

```
<?php $year = 2012 ;
      echo "Next year is: " ;
      echo ( $year + 1 ) ;
?>
```

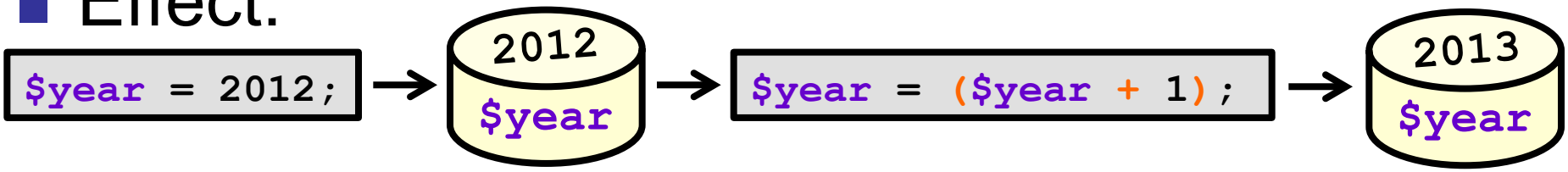
Next year is: 2013

- Another example (that also uses assignment):

```
<?php $year = 2012 ;
      echo "Year: <b>$year</b> <br/>" ;
      $year = ( $year + 1 ) ;
      echo "Year: <b>$year</b>" ;
?>
```

Year: **2012**
Year: **2013**

- Effect:



More Arithmetic

- Examples:

```php
<?php /* simple EUR to DKK converter */
      $rate = 7.5 ;
      $euro = 100 ;

      echo "For <b>$euro</b> EUR, you get " ;
      $kroner = ( $euro * $rate ) ;          // multiplication!
      echo "<b>$kroner</b> DKK." ;
?>
```

For **100** EUR, you get **750** DKK.

```php
<?php $rate = 7.5 ;
      $kroner = 100 ;

      echo "For <b>$kroner</b> DKK, you get " ;
      $euro = ( $kroner / $rate ) ;          // division!
      echo "<b>$euro</b> EUR." ;
?>
```

For **100** DKK, you get **13.333333** EUR.

More Arithmetic (rounding off)

- We can use the ***built-in function*** `round(...)` to "round off" result:

```
<?php $rate = 7.5 ;
      $kroner = 100 ;

      echo "For <b>$kroner</b> DKK, you get " ;
      $euro = ( $kroner / $rate ) ;          // division!

      $result = round( $euro ) ;             // rounded off!

      echo "ca. <b>$result</b> EUR." ;
?>
```

For **100** DKK, you get ca. **13** EUR.

Arithmetic Operators

- ***Arithmetic operators:***
  - `+` (addition) — `( 1 + 2 )`
  - `-` (substraction) — `( 3 - 4 )`
  - `*` (multiplication) — `( 5 * 6 )`
  - `/` (division) — `( 7 / 8 )`

- Example:

```
<?php $x = 2 ;
      $y = 5 ;
      $result = ((1 + (2 - $x)) * ($y / 2)) ;
      echo "Result is: <b>$result</b>" ;
?>
```

Result is: **2.5**

Increment and Decrement

- ***Increment*** and ***decrement*** operators:
  - **$x++** (increment the value of variable x by one)
    - Same as: "**$x = ($x + 1)**"
  - **$x--** (decrement the value of variable x by one)
    - Same as: "**$x = ($x - 1)**"

- Example:

```
<?php $x = 7 ;
      echo "'x' is: <b>$x</b>" ;
      echo "<p/>" ;
      $x++ ;
      echo "'x' is: <b>$x</b>" ;
?>
```

'x' is: **7**
'x' is: **8**

Agenda

- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

The ‘if’ statement

- The ‘if’ statement is used to choose between different code (depending on some *condition*):

```
if ( condition ) {
    // do something...
}
```

- Example:

```
<?php /* service kind of like: [ www.IsItFridayYet.net ] */
      $weekday = date("w") ; // get day of week
      if ( $weekday == 5 ) {
          echo "<h1>Yes, it is Friday!</h1>" ;
      }
?>
```

Yes, it is Friday!*

*if indeed run on a Friday :)

The 'if-else' statement

- Similarly for the 'if-else' statement…:

```
if ( condition ) {
    // do something...
} else {
    // do something else...
}
```

- Example:

```
<?php // service kind of like: [ www.IsItFridayYet.net ]
      $weekday = date("w") ; // get day of week
      if ( $weekday == 5 ) {
          echo "<h1>Yes, it is Friday!</h1>" ;
      } else {
          echo "<h1>No.</h1>" ;
      }
?>
```

Yes, it is Friday!*

*if indeed run on a Friday :)

‘if’ statements may be nested

- Example with **nested** if-statements:

```
<?php $weekday = date(“w”) ;
      if ( $weekday == 5 ) {
          // it is Friday
          $time = date(“h”) ;
          if ($time < 14) {
              echo “<h1>It’s DSDS time!</h1>” ;
          } else {
              echo “<h1>Yes, it is weekend!!!</h1>” ;
          }
      } else {
          // it is not Friday
          echo “<h1>No.</h1>” ;
      }
?>
```

It’s DSDS time!*

* under certain conditions

Booleans

- A boolean can take only two values:
  - `true`
  - `false`

- Example:

```
if ( true ) {
    echo "We will always print this." ;
}
```

We will always print this.

```
if ( false ) {
    echo "We will never print this." ;
}
```

Boolean expressions

- ***Comparative operators:***
  - `==` (is equal to) `( 1 == 1 )`
  - `!=` (is different from) `( 2 != 3 )`
  - `<` (is less than) `( 4 < 5 )`
  - `>` (is greater than) `( 7 > 6 )`
  - `<=` (is less than or equal to) `( 8 <= 9 )`
  - `>=` (is greater than or equal to) `( 0 >= 0 )`

```
<?php $x = 1 ;
      $y = 2 ;
      $result = ( $x > $y ) ;
      echo "Result is: <b>$result</b>" ;
?>
```

Result is: **false**

Question: why don't we use '=' to compare?

Example

Height: 184 cm.
Weight: 85.5 kg.

just a tiny bit

Your BMI is: **25.254017013233** which is too high!

```php
<?php /* Body Mass Index (BMI) calculator */
      $height = 184.0 ;
      $weight = 85.5 ;
      echo "Height: $height cm.<br/>" ;
      echo "Weight: $weight kg.<p/>" ;

      $bmi = $weight / (($height / 100) * ($height / 100)) ;
      echo "Your BMI is: <b>$bmi</b> " ;

      if ( $bmi < 20.0 ) {
          echo "which is too low!" ;
      } elseif ( $bmi > 25.0 ) {
          echo "which is too high!" ;
      } else {
          echo "which is normal." ;
      }
?>
```

Operator Precedense

- Arithmetic operators:
  - +, -, *, /
- Comparison operators:
  - <, >, <=, >=
- Equality operators:
  - ==, !=


```
<?php echo ($a+$b>=$c) ; /* interpreted as "($a+$b)>=$c" */ ?>
```

```
<?php echo ($a==$b+$c) ; /* interpreted as "$a==($b+$c)" */ ?>
```

```
<?php echo ($a!=$b<$c) ; /* interpreted as "$a!=($b<$c)" */ ?>
```

Boolean expressions

- ***Logical operators:***
  - `!` (not)
  - `&&` (and)
  - `||` (or)

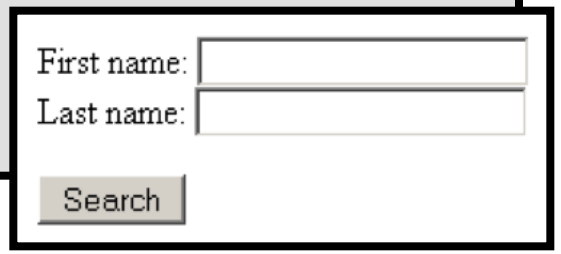
```
!($x>1)
($x>3) && ($x<9)
($x<4) || ($x>7)
```

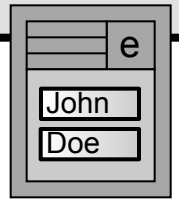
Agenda

- RECAP
- COMMENTS
- VARIABLES
- STRING OPERATIONS
- ARITHMETIC
- IF STATEMENTS
- DATA COLLECTION

Data Collection from FORMS

```
<form action="my_script.php">
    <p>
    First name: <input type="text" name="firstname" /> <br />
    Last name:  <input type="text" name="lastname" />  <br />
    </p>
    <input type="submit" value="Search" />
</form>
```





client

firstname=John
lastname=Doe

server

```
<?php $x = $_REQUEST['firstname'] ;
      $y = $_REQUEST['lastname'] ;

      $user = ( $x . " " . $y ) ;

      echo "Welcome: <b>$user</b>" ;
?>
```

Welcome: **John Doe**

The BMI Web Service (HTML)

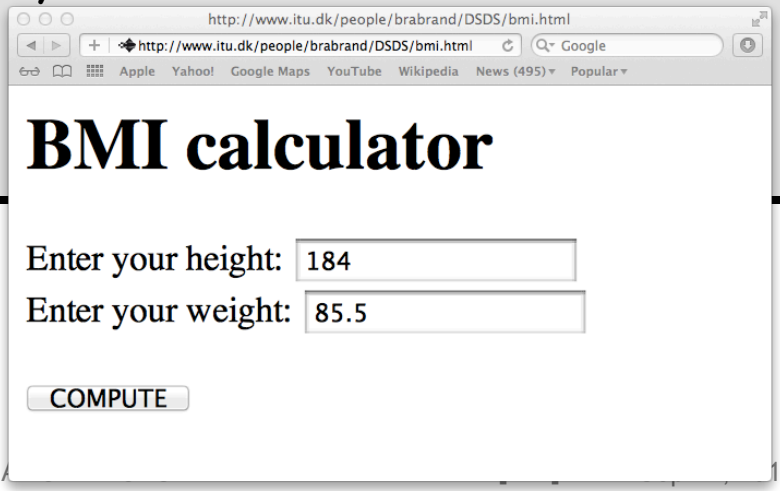
- This HTML form submits to the PHP script…

```
<html>
<body>
<h1>BMI calculator</h1>

<form action="http://www.itu.dk/people/brabrand/DSDS/bmi.php">

Enter your height: <input type="text" name="height" /><br/>
Enter your weight: <input type="text" name="weight" /><p/>
<input type="submit" value="COMPUTE" />

</body>
</html>
```



The BMI Web Service (PHP)

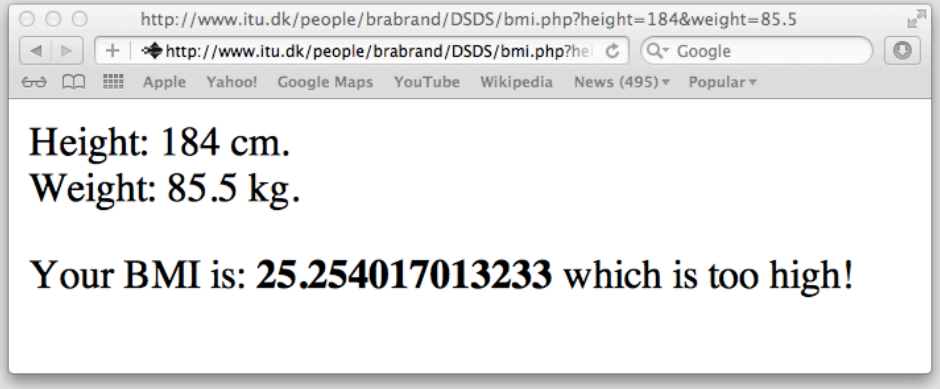
```
<-! This file is located at: "http://www.itu.dk/people/brabrand/DSDS/bmi.php" -->
<html>
    <body>
        <?php
      /* Body Mass Index (BMI) calculator */

      $h = $_REQUEST['height'] ;
      $w = $_REQUEST['weight'] ;

      echo "Height: $h cm.<br/>" ;
      echo "Weight: $w kg.<p/>" ;

      $bmi = $w / (($h / 100) * ($h / 100)) ;
      echo "Your BMI is: <b>$bmi</b> " ;

      if ( $bmi < 20.0 ) {
          echo "which is too low!" ;
      } elseif ( $bmi > 25.0 ) {
          echo "which is too high!" ;
      } else {
          echo "which is normal." ;
      }
      ?>
    </body>
</html>
```



On Handing in Assignments

- Put your **A2** hand-in solution directly in your special "**e2012/DSDS/**" directories:

- **FILE:** /import/stud/www/e2012/*username*/A2/
- **WWW:** http://www.itu.dk/stud/e2012/DSDS/*username*/A2/

- **Don't put them here any more:**

- ~~**FILE:** /home/*username*/public_html/DSDS/A2/~~
- ~~**WWW:** http://www.itu.dk/~*username*/DSDS/A2/~~

Any questions?