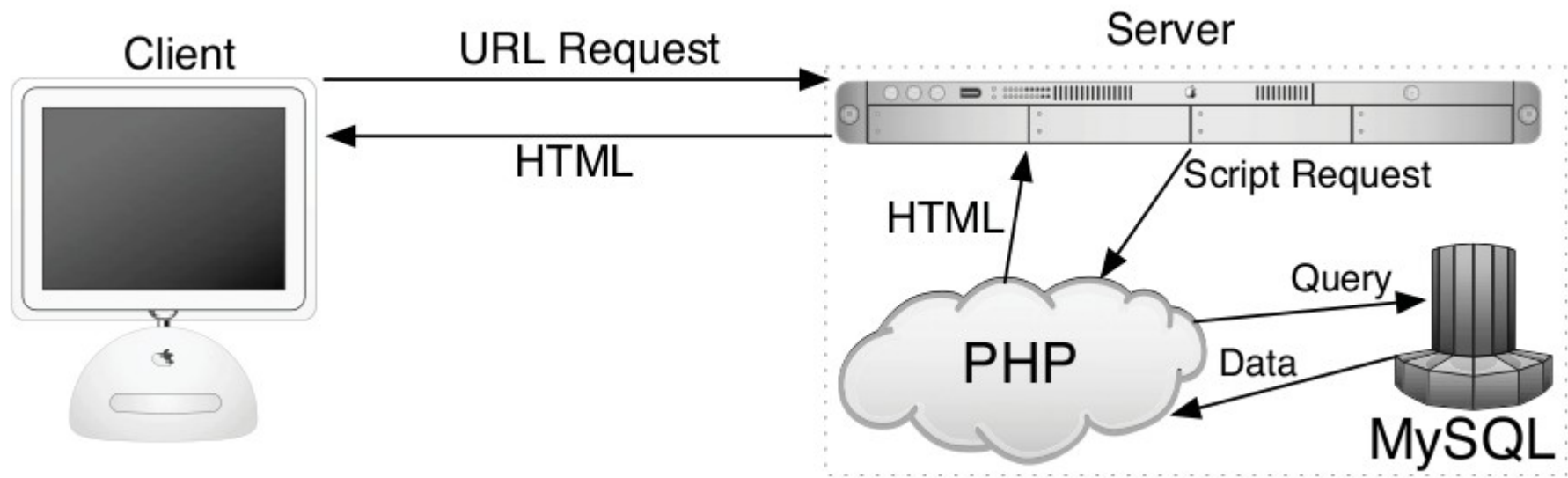


PHP

(Intro and Basic)

Intro

- Its a server side scripting language
- Personal Home Page
- Hypertext Preprocessor
- Make dynamic website, flexible and potent creatures, **more accurately described as applications than a sites** (Larry Ulman)
- More secure



Need to Prepare

- Server side scripting language, ex: PHP, ASP
- Web Server, ex: Apache, IIS, Abyss, Tomcat
- Database Server, ex: MsQL, MySQL, Oracle, for exercise can be used localhost first

Basic

- PHP is an **HTML-embedded** scripting language. This means that you can intermingle PHP and HTML code within the same file.
- `<?php`
.....
`?>`
- Anything placed within these tags will be treated by the Web server as PHP
- Put it anywhere

```
<body>
  This is html<br>
  <?php
  print "This is PHP"; // print - built in function in PHP
  ?>
</body>
```

This is html
This is PHP

```
4      <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5      <title></title>
6  </head>
7  <body>
8      This is html<br>
9      <?php
10     // put your code here
11     print "This is PHP";
12     print "<br>";
13     print "<font color=red>This is also PHP</font>"; //built in function in PHP
14     ?>
15  </body>
16 </html>
17
```

This is html
This is PHP
This is also PHP

- You can use " " or ' ' for sending data to browser

```
8      <!-- THIS IS HTML -->
9      <?php
10     // put your code here
11     print "This is PHP";
12     print "<br>";
13     print '<font color=red>This is also PHP</font><br>';
14     echo '<font color=green>This is also PHP</font>';
15     ?>
16 </body>
```

This is html
This is PHP
This is also PHP
This is also PHP

Variables

- Variables are containers used to temporarily store values.
- These values can be numbers, text, or much more complex data
- PHP has eight types of variables.
 - scalar types—**Boolean** (TRUE or FALSE), **integer**, **floating point** (decimals), and **strings** (characters)
 - two nonscalar (multivalued)—**arrays** and **objects**;
 - plus **resources** (which you'll see when interacting with databases) and **NULL** (which is a special type that has no value).

- **Identifier** (variable's name)—must start with a dollar sign (\$),
- example: **\$input**, \$a
- U can assign the value,
- Ex: **\$input = "seven";**, \$a=7;
- **"="** be called **assignment operator**

String

- `$one = "satu"; $two = "dua"; $three = "tiga";`
- Concatenation is like addition for strings, **concatenation operator (.)**
- Then u can use,

- ```
$one="satu";$two="dua";$three="tiga";
echo $one.$two.$three;
echo $one." ".$two." ".$three;
```

satuduatiga  
satu dua tiga

```
echo "$one
";
```

satu

- Built in String function → a lot, explore it by Yourself

- ```
echo strtoupper($one)."<br>";
$one2="SaTu JuGa";
echo strtolower($one2)."<br>";
?>
```

```
echo strtoupper($one)."<br>";
echo strtolower($one2)."<br>";
$one2="satu juga";
echo ucfirst($one2)."<br>";
echo ucwords($one2)."<br>";
?>
```

SATU
satu juga

SATU

Satu juga
Satu Juga

`addslashes` — Quote string with slashes in a C style

`addslashes` — Quote string with slashes

`bin2hex` — Convert binary data into hexadecimal representation

`chop` — Alias of `rtrim`

`chr` — Return a specific character

`chunk_split` — Split a string into smaller chunks

`convert_cyr_string` — Convert from one Cyrillic character set to another

`convert_uuencode` — Decode a uuencoded string

`convert_uuencode` — Uuencode a string

`count_chars` — Return information about characters used in a string

`crc32` — Calculates the crc32 polynomial of a string

`crypt` — One-way string encryption (hashing)

`echo` — Output one or more strings

`explode` — Split a string by string

`fprintf` — Write a formatted string to a stream

`get_html_translation_table` — Returns the translation table used by `htmlspecialchars` and `htmlentities`

`hebrew` — Convert logical Hebrew text to visual text

`hebrevc` — Convert logical Hebrew text to visual text with newline conversion

`html_entity_decode` — Convert all HTML entities to their applicable characters

`htmlentities` — Convert all applicable characters to HTML entities

`htmlspecialchars_decode` — Convert special HTML entities back to characters

`htmlspecialchars` — Convert special characters to HTML entities

`implode` — Join array elements with a string

`join` — Alias of `implode`

`lcfirst` — Make a string's first character lowercase

`levenshtein` — Calculate Levenshtein distance between two strings

`localeconv` — Get numeric formatting information

`ltrim` — Strip whitespace (or other characters) from the beginning of a string

`md5_file` — Calculates the md5 hash of a given file

`md5` — Calculate the md5 hash of a string

- **etc**

- Note:
 - PHP has a lot built in function → Authentication, Audio, Image, SE, Date/Time, Calender, Database, Crypto, Compression, etc
 - Make sure always get involved to <http://php.net> and the other tutorial, forum, PHP School, etc
 - Have a good IDE, make easier to learn the function

- It makes sense to make dynamic website
- PHP and CGI-scripting server can change/ manage client view from server
- Jadi server script such PHP lebih banyak kelebihan dr client scripting
- Lebih secure dan some parts dapat menggantikan client scripting

```
<style>
    .col1{
        color: blue;
    }
    .col2{
        color:red;
    }
</style>
</head>
<body>
    <script type="text/javascript">
        todaynow = new Date(); //get day and time
        document.write("<p> <font class='col1'>" + todaynow + "</font></p>");
    </script>
    <?php

    date_default_timezone_set('Asia/Indonesia');
    echo "<p class='col2'>".date('l jS \of F Y h:i:s A');echo "</p><br>";
    ?>
```

Tue Mar 29 2011 02:48:55 GMT+0700 (WIT)

Tuesday 29th of March 2011 02:48:55 AM

```

<body bgcolor="pink" class="b">
  <table border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td>
        <?php
          date_default_timezone_set('Asia/Indonesia');
          echo "<p class='col1'>".date('l jS \of F Y');echo "</p><br>";
        ?>
      </td>
    </tr>
    <tr>
      <td>
        <?php
          date_default_timezone_set('Asia/Indonesia');
          echo "<p class='col2'>".date('h:i:s A');echo "</p><br>";
        ?>
      </td>
    </tr>
  </table>
</body>

```

Tuesday 29th of March 2011

03:09:11 AM

Number

```
$a=7;  
$b=7.17;  
$c=-7.7171717171;  
$d=7e17;  
$e=1765434271;  
$f=17e-7;  
echo $a*$c." <br>";  
echo $a+$b." <br>";  
echo $a+$c." <br>";  
echo $a-$c." <br>";  
echo $a*$f." <br>";  
echo "<font color=green>".$a*$f."</font><br>";  
?>
```

```
-54.0202020197  
14.17  
-0.7171717171  
14.7171717171  
1.19E-5  
1.19E-5
```

- Arithmetic Operator

Arithmetic Operators	
OPERATOR	MEANING
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

```
<style type="text/css">
  .class1{
    color: red;
  }
</style>
```

```
echo "<font color=green>". $a*$f ." </font><br>";
echo '<font class="class1">' . $e*$f .' <font><br>';
echo '<font class="class1">' . $c*$f .' <font><br>';
```

1.19E-5

3001.2382607

-1.3119191919191E-5

- Arithmetic built in function → a lot, explore by Urself

```
print round($b). "<br>";  
print round($c,3). "<br>";  
?>
```

7
-7.717

```
echo number_format($c). "<br>";  
echo number_format($c*$b). "<br>";  
echo number_format($c*$b,2). "<br>";  
?>
```

-8
-55
-55.33

Conditional

- if, else, elseif

-

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

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

```
if (condition1) {
// .....
} elseif (condition2) {
// .....
} else {
// ....
}
```


Comparative and Logical Operators

SYMBOL	MEANING	TYPE	EXAMPLE
==	is equal to	comparison	\$x == \$y
!=	is not equal to	comparison	\$x != \$y
<	less than	comparison	\$x < \$y
>	greater than	comparison	\$x > \$y
<=	less than or equal to	comparison	\$x <= \$y
>=	greater than or equal to	comparison	\$x >= \$y
!	not	logical	!\$x
&&	and	logical	\$x && \$y
	or	logical	\$x \$y
XOR	and not	logical	\$x XOR \$y

```
7 $a=7;
8 $b=7.17;
9 $c=-7.7171717171;
10 $d=7e17;
11 $e=1765434271;
12 $f=17e-7;
13
14 if($a>0){
15     echo $a." is positif number";
16 }
```

7 is positif number

```
18 if (($a>$b)){
19     echo "subs is";echo $a-$b."<br>";
20 }else{
21     echo "subs is negatif ";echo $a-$b."<br>";
22 }
23
```

subs is negatif -0.17

```

3
4 if(($b>0) and ($c>0)){
5     echo "both positif then the sum is ";echo $b+$c."<br>";
6 }
7 else if(($b>0) and ($c<0)){
8     echo "better subs them to reach pos value ";echo $b-$c."<br>";
9 }
10 else{
11     echo "one of them is positif<br>";
12 }

```

better subs them to reach pos value 14.8871717171

```

3
4 if(($d>0) and ($e>0)){
5     echo "both positif then the sum is ";echo $d+$e."<br>";
6 }
7 else if(($d>0) and ($e<0)){
8     echo "better subs them to reach pos value ";echo $d-$e."<br>";
9 }
10 else{
11     echo "one of them is positif<br>";
12 }

```

both positif then the sum is 7.00000001765E+17

```
24 if(($c>0) and ($e>0)){
25     echo "both positif then the sum is ";echo $d+$e."<br>";
26 }
27 else if(($c>0) and ($e<0)){
28     echo "better subs them to reach pos value ";echo $d-$e."<br>";
29 }
30 else{
31     echo "one of them is positif<br>";
32 }
```

one of them is positif

```
34 $g="Sebelas";$h="11Sebelas11";
35 if (strlen($g)<10){
36     echo "reject, should at least 10 chars<br>";
37 }else{
38     echo "accepted<br>";
39 }
```

rejected, should at least 10 chars

```
11 if(preg_match('[0-9]', $g)){
12     echo "accepted<br>";
13 }
14 else{
15     echo "rejected<br>";
16 }
```

```
0
1 if(preg_match('[0-9]', $h)){
2     echo "accepted<br>";
3 }
4 else{
5     echo "rejected<br>";
6 }
_
```

rejected
accepted

- Switch

The switch conditional compares the value of \$variable to the different cases. When it finds a match, the following code is executed, up until the break. If no match is found, the default is executed, assuming it exists (it's optional)

```
switch ($variable) {  
  case 'value1':  
    // Do this.  
    break;  
  case 'value2':  
    // Do this instead.  
    break;  
  default:  
    // Do this then.  
    break;  
}
```

```
$i=date(N);
switch ($i){
    case 1:
        echo "OMG...look its already Monday, move!!!<br>";
        break;
    case 2:
        echo "Hmm...Tuesday morning, Web Programming time,
            no worries its the easiest course i have<br>";
}
```

Hmm...Tuesday morning, Web Programming time, no worries its the easiest course i have

Loop

- for, while, foreach

```
while (condition)
{
// ....
}
```

As long as the condition part of the loop is true, the loop will be executed.

```
for (initial; condition; closing) {
// .....
}
```

Upon first executing the loop, the initial expression is run. Then the condition is checked and, if true, the contents of the loop are executed. After execution, the closing expression is run and the condition is checked again.


```

34 $j=0; *
35 while ($j<=7){ **
36     for ($k=0;$k<=$j;$k++){ ***
37         echo "*"; ****
38     } *****
39     echo "<br>";
40     $j++;
41 }

```

```

34 $j=0; *
35 while ($j<=7){ **
36     for ($k=0;$k<=$j;$k++){ ***
37         if (($k%2)==0){ ****
38             echo "<font color=red>*</font>"; *****
39         }elseif(($k%2)==1){
40             echo "<font color=green>*</font>";
41         }
42     }
43     echo "<br>";
44     $j++;
45 }

```

Working with Form

- Form is very important in dynamic page and perhaps the most important one
- U have already learned in HTML section, havent U? Mean already understood, not yet? U still have a time

- Lets break this simple example

```
1 | <body>
2 |   <form name="ex1" method="post" action="index3.php">
3 |
4 |   </form>
```

tag action dictates to
which page the form data will be sent

The method attribute of a form dictates
how the data is sent to the handling page

- The **get method** sends the submitted data to the receiving page as a series of name-value pairs appended to the URL.
- The benefit of using the get method is that the resulting page can be bookmarked in the user's Web browser (since it's a URL).
- For that matter, you can also click Back in your Web browser to return to a get page, or reload it without problems

- Sends the submitted data to the receiving page as a series of name-value pairs appended to the URL
ex: `http://example.com/?name=uns`
- But there is a **limit in how much data** can be transmitted via get, and this method is **less secure** (since the data is visible) and can be entered via URL
- Generally, get is used **for requesting information**, like a particular record from a database or the results of a search (searches almost always use get)

- The post method is used when an action is required,
- as when a database record will be updated or an email should be sent.
- More secure, not embed value in URL

```

<body>
  <form name="ex1" method="post" action="index3.php">
    <h4>Ok buddy, gimme a brief about U</h4>
    Name      : <input name="Uname" type="text" size="15" maxlength="100"><br>
    Gender    : <input type="radio" name="Ugender" value="female">Female <input type="radio" name="Ugender" value="male">Male <br>
    Email     : <input name="Uemail" type="text" size="30" maxlength="500"><br>
    Mobilephone : <input name="Umphone" type="text" size="15" maxlength="15"><br>
    A brief of U, dont be shy write here : <br>
    <textarea name="UU" rows="5" cols="50"></textarea><br>
    <input type="submit" name="Usub" value="Send"> <input type="reset" name="Ures" value="clear">
  </form>
</body>

```

Ok buddy, gimme a brief about U

Name :

Gender : Female Male

Email :

Mobilephone :

A brief of U, dont be shy write here :

- Let php handle it
- Whatever the user types into those elements will be accessible via a PHP variable named `$_REQUEST['...']`
- Yes, case sensitive, below not same
- `$_REQUEST['Uname']`,
`$_REQUEST['UName']`,
`$_REQUEST['uname']`


```
<?php
```

```
    // remove unneeded chars
```

```
$name=strip_tags($_REQUEST['Uname']);//
$gender=$_REQUEST['Ugender'];
$email=strip_tags($_REQUEST['Uemail']);
$mphone=strip_tags($_REQUEST['Umphone']);
$u=strip_tags($_REQUEST['UU']);
echo "Thanks buddy, Ur name ";
echo ucwords($name)."<br>";
if (strlen($name)>=40){
    echo "ow MG, U have such a long name buddy ";
}
echo "and U are an happy ".$gender." ,i hope<br>";
echo "we can contact U, email ".$email." <br>";
if (preg_match('^\\@?\\.^', $email)){
echo "see what just U got in ".$email." <br>";
}else{
    echo "but is this true email, buddy? please check it again<br>";
}
echo "or mobilephone ".$mphone;
if (preg_match('/[0..9]/', $mphone)){
    echo " so dont angry if we will bother U<br>";
}
else{
echo " but, is this true number, check it again";
}

```

```
?>
```

Ok buddy, gimme a brief about U

Name :

Gender : Female Male

Email :

Mobilephone :

A brief of U, dont be shy write here :

Thanks buddy, Ur name Yuni
and U are an happy female ,i hope
we can contact U, email yu@yu
but is this true email, buddy? please check it again
or mobilephone 09878765 so dont angry if we will bother U

Name :

Gender : Female Male

Email :

Mobilephone :

A brief of U, dont be shy write here :

Thanks buddy, Ur name Yuni
and U are an happy female ,i hope
we can contact U, email yu@yu.yu
see what just U got in yu@yu.yu
or mobilephone 877h87787887 but, is this true number, check it again

Character Classes

CLASS	SHORTCUT	MEANING
[0-9]	\d	Any digit
[\f\r\t\n\v]	\s	Any white space
[A-Za-z0-9_]	\w	Any word character
[^0-9]	\D	Not a digit
[^\f\r\t\n\v]	\S	Not white space
[^A-Za-z0-9_]	\W	Not a word character

Meta-Characters

CHARACTER	MEANING
\	Escape character
^	Indicates the beginning of a string
\$	Indicates the end of a string
.	Any single character except newline
	Alternatives (or)
[Start of a class
]	End of a class
(Start of a subpattern
)	End of a subpattern
{	Start of a quantifier
}	End of a quantifier

Another ex

www.d

```
<form name="up" method="post" action="upload.php" enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="1000">
  Upload file: <input name="up2" type="file">
  <input type="submit" value="Upload">
```

Upload file:

```
$BASEDIR = "/home/dww/";
if (!file_exists($BASEDIR.$_FILES['up2']['name'])) {
    move_uploaded_file($_FILES['up2']['tmp_name'],$BASEDIR.$_FILES['up2']['name']);
    print("File uploaded successfully");
}
else {
    print("File already exists");
}
```

- In e-commerce

```

<?php
// put your code here
if ($Calc=true){
    $i=$_POST['Item'];
    $n=$_POST['Num'];
    switch($i){
        case 'Tas type TS23':
            $price="30000";break;
        case 'Tas type TS33':
            $price="35000";break;
        case 'Tas type TS75':
            $price="50000";break;
        case 'Tas type TS17':
            $price="60000";break;
        case '5':
            $price="75000";break;
    }
    echo "Price of item: ".$price;
    echo ", Number of item: ".$n;
    $total = $price * $n;
    echo ", Total purchase: ".$total";
}
?>

```

```

<form name="comm" action="Formcomm.php" method="POST">
    Item = <select name="Item" class="c">
        <option>Tas type TS23</option>
        <option>Tas type TS33</option>
        <option>Tas type TS75</option>
        <option>Tas type TS17</option>
        <option>Tas type TS09</option>
    </select>
    Number = <input type="text" name="Num" value="1" size="1" class="c"/>
    <input type="submit" value="Calculate" name="Calc" class="c"/>
</form>

```

Item = Number =

Price of item: 30000, Number of item: 8, Total purchase:240000