## DAY 5–PHP

### Strings

#### Experiments

<table>
<thead>
<tr>
<th>Experiment 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>print 'I have gone to the store.';</td>
</tr>
<tr>
<td>print 'I've gone to the store.';</td>
</tr>
<tr>
<td>print 'Would you pay $1.75 for 8 ounces of tap water?';</td>
</tr>
<tr>
<td>print 'In double-quoted strings, newline is represented by \n';</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>print &quot;I've gone to the store.&quot;;</td>
</tr>
<tr>
<td>print &quot;The sauce cost $10.25.&quot;;</td>
</tr>
<tr>
<td>$cost = '$10.25';</td>
</tr>
<tr>
<td>print &quot;The sauce cost $cost.&quot;;</td>
</tr>
<tr>
<td>print &quot;The sauce cost $061\060.\x32\x35.&quot;;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>print &lt;&lt;&lt; END</td>
</tr>
<tr>
<td>It's funny when signs say things like:</td>
</tr>
<tr>
<td>Original &quot;Root&quot; Beer</td>
</tr>
<tr>
<td>&quot;Free&quot; Gift</td>
</tr>
<tr>
<td>Shoes cleaned while &quot;you&quot; wait</td>
</tr>
<tr>
<td>or have other misquoted words.</td>
</tr>
<tr>
<td>END;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>print &lt;&lt;&lt; PARSLEY</td>
</tr>
<tr>
<td>It's easy to grow fresh:</td>
</tr>
<tr>
<td>Parsley</td>
</tr>
<tr>
<td>Chives</td>
</tr>
<tr>
<td>on your windowsill</td>
</tr>
<tr>
<td>PARSLEY;</td>
</tr>
<tr>
<td>print &lt;&lt;&lt; DOGS</td>
</tr>
<tr>
<td>If you like pets, yell out:</td>
</tr>
<tr>
<td>DOGS AND CATS ARE GREAT!</td>
</tr>
<tr>
<td>DOGS;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>if ($remaining_cards &gt; 0) {</td>
</tr>
<tr>
<td>$url = '/deal.php';</td>
</tr>
<tr>
<td>$text = 'Deal More Cards';</td>
</tr>
<tr>
<td>} else {</td>
</tr>
<tr>
<td>$url = '/new-game.php';</td>
</tr>
<tr>
<td>$text = 'Start a New Game';</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>print &lt;&lt;&lt; HTML</td>
</tr>
<tr>
<td>There are &lt;b&gt;$remaining_cards&lt;/b&gt; left.</td>
</tr>
<tr>
<td>&lt;p&gt;</td>
</tr>
</tbody>
</table>
Experiment 6
$\texttt{html} = <<< END
<div class="$\texttt{divClass}$">
<ul class="$\texttt{ulClass}$">
<li>$\texttt{\$listItem}$'
</li>
</div>'
print $\texttt{html}$;

Experiment 7
$\texttt{\$neighbor} = 'Hilda';$
print $\texttt{\$neighbor}[3]$;

Experiment 8
<?php
if (strpos($_\texttt{POST}'email', '@') === false) {
    print 'There was no @ in the e-mail address!';
}
?>

Experiment 9
<?php
$\texttt{substring} = substr($\texttt{string}, $\texttt{start}, $\texttt{length})$;
$\texttt{username} = substr($_\texttt{GET}'username', 0, 8)$;
?>

Experiment 10
print substr('watch out for that tree', 6, 5);

Experiment 11
print substr('watch out for that tree', 17);

Experiment 12
print substr('watch out for that tree', 20, 5);

Experiment 13
print substr('watch out for that tree', -6);
print substr('watch out for that tree', -17, 5);

Experiment 14
print substr('watch out for that tree', 15, -2);
print substr('watch out for that tree', -4, -1);

Experiment 15
// Everything from position $\texttt{\$start}$ to the end of $\texttt{\$old_string}$
// becomes $\texttt{\$new_substring}$
$\texttt{new_string} = \texttt{substr_replace($\texttt{old_string}, \texttt{new_substring}, \texttt{start});}$

// $\texttt{\$length}$ characters, starting at position $\texttt{\$start}$, become
// $\texttt{\$new_substring}$
$\texttt{new_string} = \texttt{substr_replace($\texttt{old_string}, \texttt{new_substring}, \texttt{start}, \texttt{length});}$
Experiment 16

```php
print substr_replace('My pet is a blue dog.', 'fish.', 12);
print substr_replace('My pet is a blue dog.', 'green', 12, 4);
$credit_card = '4111 1111 1111 1111';
print substr_replace($credit_card, 'xxxx ', 0, strlen($credit_card) - 4);
```

Experiment 17

```php
print substr_replace('My pet is a blue dog.', 'fish.', -9);
print substr_replace('My pet is a blue dog.', 'green', -9, 4);
```

Experiment 18

```php
print substr_replace('My pet is a blue dog.', 'Title: ', 0, 0);
```

Example. Displaying long text with ellipsis

```php
$My pet is a blue dog.

Example. Displaying long text with ellipsis

```php
$credit_card = '4111 1111 1111 1111';
print substr_replace($credit_card, 'xxxx ', 0, strlen($credit_card) - 4);
```

Experiment 19

```php
<?php
$string = "This weekend, I'm going shopping for a pet chicken.";
$vowels = 0;
for ($i = 0, $j = strlen($string); $i < $j; $i++) {
    if (strstr('aeiouAEIOU', $string[$i])) {
        $vowels++;
    }
}
?>
```

Experiment 20

```php
<?php
function lookandsay($s) {
    // initialize the return value to the empty string
    $r = '';
    // $m holds the character we're counting, initialize to the first
    // character in the string
    $m = $s[0];
    // $n is the number of $m's we've seen, initialize to 1
    $n = 1;
    for ($i = 1, $j = strlen($s); $i < $j; $i++) {
        // if this character is the same as the last one
        if ($s[$i] == $m) {
            // increment the count of this character
            $n++;
        } else {
            // otherwise, add the count and character to the return
            // value
            $r .= $n.$m;
            // set the character we're looking for to the current one
            $m = $s[$i];
            // and reset the count to 1
            $n = 1;
        }
    }
    // return the built up string as well as the last count and
    // character
    return $r.$n.$m;
}
```
for ($i = 0, $s = 1; $i < 10; $i++) {
    $s = lookandsay($s);
    print "$s <br/>
};

Experiment 21
<?php
print strrev('This is not a palindrome.');?
>

Experiment 22
<?php
$s = "Once upon a time there was a turtle.";
// break the string up into words
$words = explode(' ', $s);
// reverse the array of words
$words = array_reverse($words);
// rebuild the string
$s = implode(' ', $words);
print $s;
?>

Experiment 23
<?php
$reversed_s = implode(' ', array_reverse(explode(' ', $s)));
?>

Exercises

Question 1
Buat script untuk melakukan pengecekan apakah sebuah string merupakan palindrome atau bukan.

Contoh:
Input : kasur rusak
Output : palindrome

Input : kasur itu rusak
Output : bukan palindrome