

## Using String in PHP

---

### Lecture 6

March 19, 2007

©2007, I.Sarah Al-Shareef

1

## Table of Content

---

- Overview Substring Functions
- Manipulating Substrings
  - Testing Occurrences of Substring
  - Accessing Substring
  - Replacing Substring
- Processing Strings
  - Joining/Disassembling Stings
  - Trimming Blanks/Spaces
  - Controlling Cases
  - Comparing Strings

March 19, 2007

©2007, I.Sarah Al-Shareef

2

## Overview Substring Functions

<u>Function</u>	<u>Description</u>
substr_count()	Counts the number of substring occurrences
strstr()	Finds the first occurrence of a string
strchr()	Can be used as an alias of strstr()
stristr()	Performs the same functionality as strstr() but is case-insensitive
substr_replace()	Replaces text within a portion of a string
strpos()	Finds the position of the first occurrence of a string
substr()	Returns a piece of a string

March 19, 2007

©2007, I.Sarah Al-Shareef

3

## Manipulating Substrings

### ■ Testing the occurrences of a substring:

We can test the occurrences of a specific substring in any string.

- This could be useful in *building a search engine*.
- This is usually used with data stored in a file.
- If your data is stored in a database, use the built-in SQL queries.

### ■ *Remember :*

The string index is starting from 0.

March 19, 2007

©2007, I.Sarah Al-Shareef

4

## L#06: PHP Strings

### Example

---

- Let's build a basic search engine. This search engine will take the block of text and the key to search about with it.

It will return the number of occurrences and their positions (i.e. indexes) in the text.

March 19, 2007

©2007, I.Sarah Al-Shareef

5

### Example solution

---

- To get the number of occurrences, we will use :

```
int substr_count(string $b, string $k)
```

- To get the position of the first occurrence from the position  $\$s$ , we will use:

```
int strpos(string $b, string $k, int $s)
```

March 19, 2007

©2007, I.Sarah Al-Shareef

6

## L#06: PHP Strings

### Example solution (cont.)

---

```
<?php
function searchtext ($block, $key)
{ //First, check if there is any occurrences.
  if (substr_count ($block, $key) == 0)
  { echo "No instances were found of this".
    "search";
  }
  else
  { //Now, we will positions
    //We will start at the beginning.
    $start = 0;
    //Use a flag to stop searching
    $again = true;
  }
}
```

March 19, 2007

©2007, I.Sarah Al-Shareef

7

### Example solution (cont.)

---

```
while ($lookagain)
{ $pos =strpos($block, $key, $start);
  if($pos!=0)
  { echo "$key was found at".
    "position: $pos<br>";
    //We increase the position.
    $start = $pos + 1;
  }
  else
  { //If not found, break.
    $again = false;
  }
}
```

March 19, 2007

©2007, I.Sarah Al-Shareef

8

## L#06: PHP Strings

### Example solution (cont.)

---

```
        echo "$key has".
        substr_count ($block, $key). " matches.";
    }
}
searchtext ("Welcome to selected topics-2".
           "course!", "to");
?>
//Result:
//to was found at position: 8
//to was found at position: 20
//to has 2 matches.
```

March 19, 2007

©2007, I.Sarah Al-Shareef

9

### Manipulating Substrings cont.

---

#### ■ **Accessing a substring:**

We can get a substring from a specific position with a specific length too.

- This could be useful in *if you have a limited space on your website.*
- When the text exceed that length, you have to break it into pages.

March 19, 2007

©2007, I.Sarah Al-Shareef

10

## L#06: PHP Strings

### Example

---

- Let's try to break a given text only if its length more than 10.

March 19, 2007

©2007, I.Sarah Al-Shareef

11

### Example solution

---

- To get a string length, we use:

```
int strlen(string $b)
```

- To get a substring, we use:

```
int substr(string $b, int $pos, int length)
```

March 19, 2007

©2007, I.Sarah Al-Shareef

12

## L#06: PHP Strings

### Example solution cont.

---

```
<?php
    $text = "Hello, how are you today? ".
            "I am fine!";

    if(strlen($text) >= 10)
    {   echo substr($text,0,9);
    }
    else
    {   echo $text;
    }
?>
```

March 19, 2007

©2007, I.Sarah Al-Shareef

13

### Manipulating Substrings cont.

---

- Getting a substring by defining its starting:
  - It could be useful for validation process.
- The used functions:
  - string **strstr**(string \$b, string \$s)
  - string **striestr**(string \$b, string \$s)  
same as strstr but it is case-insensitive.

March 19, 2007

©2007, I.Sarah Al-Shareef

14

## L#06: PHP Strings

### Example

---

- Build a script that get the domain name from a given URL.

```
<?php
$url = "www.uqu.edu.sa";
$domain = strstr($url, ".");
echo $domain;
?>
```

March 19, 2007

©2007, I.Sarah Al-Shareef

15

### Manipulating Substrings cont.

---

- Replace a substring with another
- Used functions:
  - string **substr\_replace**( string \$b, string \$replace, int \$start, int \$length)  
The start and length are optional.

March 19, 2007

©2007, I.Sarah Al-Shareef

16

# L#06: PHP Strings

## Overview String Functions

<b>Function</b>	<b>Description</b>
strlen()	Length of a given string
implode()/ join()	Concatenate group of strings by a specified character
explode()/ strtok()	Separate a string into parts
strtoupper()	Convert all characters case to caps
strtolower()	Convert all characters case to caps
ucfirst()	Convert only 1 <sup>st</sup> letter of the 1 <sup>st</sup> word to caps
ucwords()	Convert only 1 <sup>st</sup> letter of each word
trim()	Delete the spaces from both beginning/end
rtrim()	Delete the spaces from end
ltrim()	Delete the spaces from beginning

March 19, 2007

©2007, I.Sarah Al-Shareef

17

## Overview String Functions cont.

<b>Function</b>	<b>Description</b>
nl2br()	Convert every newline (i.e. ↵) to  
wraptxt()	Put   after specified number of characters
strcmp()	Compare between two strings and return an int. If it is 0, they are equal. If it is +ve the first is larger. If -ve the second is larger.
strncmp()	Same as strcmp() but it won't be applied on the whole string but on a specified length.
strcasecmp()	Same as strcmp() but case insensitive.
strncasecmp()	Same as strncmp() but case insensitive.

March 19, 2007

©2007, I.Sarah Al-Shareef

18

### Processing Strings

---

#### ■ Joining Strings:

- The simple concatenate operator “.”
- Using:

```
string implode(string glue, array pieces)
string join(string glue, array pieces )
```

March 19, 2007

©2007, I.Sarah Al-Shareef

19

### Processing String cont.

---

#### ■ Disassembling String

- Using:

```
array explode(string $sep, string $str,
int length$)
```

Length is optional

```
string strtok(string $str, string $token)
```

We'll send the first parameter only once at the beginning.  
Then we just send the token. Usually a loop is used.

March 19, 2007

©2007, I.Sarah Al-Shareef

20

## L#06: PHP Strings

### Processing Strings cont.

---

#### ■ **Example:**

Use the user email to define his/her Username and Mail Server.

March 19, 2007

©2007, I.Sarah Al-Shareef

21

### Processing String cont.

---

#### ■ **Controlling Case:**

- For converting the whole string, use:

```
string strtoupper ( string $str )
```

```
string strtolower ( string $str )
```

- For converting the first letter only, use:

```
string ucfirst ( string $str )
```

- For converting the first letter of each word, use:

```
string ucwords ( string $str )
```

March 19, 2007

©2007, I.Sarah Al-Shareef

22

### Processing String cont.

---

#### ■ Trimming Spaces:

##### ■ Using:

```
string trim(string $str)
string ltrim(string $str)
string rtrim(string $str)
```

March 19, 2007

©2007, I.Sarah Al-Shareef

23

### Processing String cont.

---

#### ■ Wrapping Text:

##### ■ To convert every newline to a <br>, use:

```
string nl2br ( string $text )
```

##### ■ To wrap a given text into a specific width:

```
string wordwrap ( string $text,
                  int $width, string break )
```

March 19, 2007

©2007, I.Sarah Al-Shareef

24

### Processing Strings cont.

---

- **Comparing Strings:**

- Use == operator  
But it isn't precise... specially with existing of blanks.

- So, use:

```
int strcmp ( string str1, string str2 )
```

- If you want to compare specific length:

```
int strncmp(string str1, string str2, int len)
```

### Processing Strings cont.

---

- **Case-insensitive comparing:**

```
int strcasecmp ( string str1, string str2 )
```

```
int strncasecmp ( string str1, string str2,  
int len )
```