

## CSCI 241

### Strings and Other Data Manipulation Functions in PHP

Department of CS & QM  
Winthrop University  
Fall 2011

Prepared by X. Wang

## Strings

- PHP provides a large library of string functions to help transform, manipulate, and manage strings
- Length of a string
- The `strlen()` function returns the number of eight-bit characters in the subject string
 

```
integer strlen(string subject)
```
- For example:
 

```
print strlen("This is a string");
```

 prints 16

Prepared by X. Wang

2

## Printing and Formatting Strings

- Using `printf()` and `sprintf()`
- The `printf()` and `sprintf()` functions can create formatted output
 

```
integer printf(string format [, mixed args ...])
string sprintf(string format [, mixed args ...])
```
- Both functions expect `format` string with optional arguments
- `printf()` directly prints the output to the output buffer
- `sprintf()` returns the output as a string

Prepared by X. Wang

3

## Printing and Formatting Strings

- Using `printf()` and `sprintf()`
- Conversion types used in `printf()` and `sprintf()`

<code>%%</code>	A literal percent character
<code>%b</code>	An integer formatted as a binary number
<code>%c</code>	An integer formatted as an ASCII character
<code>%d</code>	An integer formatted as a signed decimal number
<code>%u</code>	An integer formatted as an unsigned decimal number
<code>%o</code>	An integer formatted as an octal number
<code>%x</code> or <code>%X</code>	An integer formatted as a hexadecimal number using lowercase letters or uppercase letters
<code>%f</code>	An float formatted with specified decimal places
<code>%s</code>	A string

Prepared by X. Wang

4

## Printing and Formatting Strings

- Using `printf()` and `sprintf()`
- Between `%` and the formatting type specifier, the output width, precision and alignment can be specified
- An integer number before the point specifies the total output width
- An integer number after the point specifies the number of digits after the decimal point
- By default, the output is right-aligned. A minus (-) sign makes output left-aligned
- To make sure all spaces are viewable, an output need to be placed within a `<pre>` `</pre>` element

Prepared by X. Wang

5

## Printing and Formatting Strings

- ```
$pi = 3.1415926;
$v = 1531;
$str = "Solution";
print "<pre>";
printf("\$pi = %d;\n", $pi);
printf("\$pi = %f;\n", $pi);
printf("\$pi = %7E;\n", $pi);
printf("\$pi = %9.3f;\n", $pi);
printf("\$pi = %-9.3f;\n", $pi);
printf("%s: \$v = %d, \$v = %x, \$v = %X", $str, $v, $v, $v);
print "</pre>";
```
- The output is:
 

```
$pi = 3;
$pi = 3.141593;
$pi = 3.1415926;
$pi = 3.142;
$pi = 3.142 ;
Solution: $v = 1531, $v = 5fb, $v = 5FB
```

Prepared by X. Wang

6

## Printing and Formatting Strings

### Other functions

- Please read PHP online manual for the following functions

```
string strtolower(string subject)
string strtoupper(string subject)
string ucfirst(string subject)
string ucwords(string subject)
string ltrim(string subject [, string character_list])
string rtrim(string subject [, string character_list])
string trim(string subject [, string character_list])
string nl2br(string input)
```

Prepared by X. Wang

7

## Comparing Strings

- PHP provides the string comparison functions `strcmp()` and `strncmp()` that compare two strings in alphabetical order, `str1` and `str2`

```
integer strcmp(string str1, string str2)
integer strncmp(string str1, string str2, int length)
```

- If `str1` and `str2` are equal, returns 0
- If `str1` is less than `str2`, returns -1
- If `str1` is greater than `str2`, returns 1
- The functions `strcasecmp()` and `strncasecmp()` are case-insensitive version of `strcmp()` and `strncmp()`

Prepared by X. Wang

8

## Extracting a Substring

- The `substr()` function returns a substring from a source string

```
string substr(string source, int start [, int length])
```

- `start` is the starting position (index),
- `length` specifies how many characters to be extracted
- If no length, extracts all characters up to the end of the source string
- Both `start` and `length` can be negative integers

Prepared by X. Wang

9

## Finding the Position of a Substring

- The `strpos()` function returns the index of the first occurring substring `needle` in the source string

```
integer strpos(string source, string needle [,int offset])
```

- The `strrpos()` function returns the index of the last occurring substring `needle` in the source string

```
integer strrpos(string source, string needle [,int offset])
```

- If the substring is not found by `strpos()` or `strrpos()`, both functions return false
- If the call with the `offset`, the search begins from the `offset`

Prepared by X. Wang

10

## Finding the Position of a Substring

- For example

```
$domain = "php.cs.winthrop.edu";
$a = 0;
while (($b = strpos($domain, ".", $a)) != false) {
    print substr($domain, $a, $b-$a) . "<br />";
    $a = $b + 1;
}
print substr($domain, $a);
```

- The output is:

```
php
cs
winthrop
edu
```

Prepared by X. Wang

11

## Replacing Characters and Substrings

- The `substr_replace()` function returns a copy of the source string with the characters from the position `start` to the end of the string replaced with the `replace` string

```
string substr_replace(string source, string replace,
int start [, int length])
```

- If the optional `length` is supplied, only length characters are replaced
- If `length` is 0, the `replace` string will be inserted at the `start` position

Prepared by X. Wang

12

## Replacing Characters and Substrings

---

- Form example

```
$var = "abcdefghij";
print substr_replace($var, "XYZ", 3)."<br />";
print substr_replace($var, "XYZ", 3, 3)."<br />";
print substr_replace($var, "XYZ", 3, 0)."<br />";
```

- The output is:

```
abcXYZ
abcXYZghij
abcXYZdefghij
```

Prepared by X. Wang

13

## Replacing Characters and Substrings

---

- The `str_replace()` function returns a string created by replacing occurrences of the search string in source with the string replace

```
string str_replace(mixed search, mixed replace,
                  string source)
```

- For example

```
$var = "old-age for the old";
print str_replace("old", "new", $var)."<br />";
```

- The output is

```
new-age for the new
```

Prepared by X. Wang

14

## Integers and Floats

---

- Some mathematical functions

- `integer abs(integer number)`
- `float abs(float number)`
- `float ceil(float number)`
- `float floor(float number)`
- `float round(float number [,integer precision])`  
// by default, rounding to zero decimal places
- `float exp(float number)`
- `float pow(float base, number exp)`
- `float sqrt(float number)`
- `float log(float number [, float base])`
- `float log10(float number)`

Prepared by X. Wang

15

## Integers and Floats

---

- Number system conversions

- `string decbin(integer number)`
- `integer bindec(string binarystring)`
- `string dechex(integer number)`
- `integer hexdec(string hexstring)`
- `string decoct(integer number)`
- `integer octdec(string octalstring)`

Prepared by X. Wang

16

## Integers and Floats

---

- Basic Trigonometry functions

- `float sin(float arg)` // Sine of arg in radians
- `float cos(float arg)` // Cosine of arg in radians
- `float tan(float arg)` // Tangent of arg in radians
- `float asin(float arg)` // Arc sine of arg in radians
- `float acos(float arg)` // Arc cosine of arg in radians
- `float atan(float arg)` // Arc tangent of arg in radians
- `float atan2(float y, float x)` // Arc tangent of x/y
- `float pi()` // Return the value 3.1415926535898
- `float deg2rad(float arg)` // Converts arg degrees to radians
- `float rad2deg(float arg)` // Converts arg radians to degrees

Prepared by X. Wang

17

## Integers and Floats

---

- Testing number results

- `boolean is_nan(float value)`  
returns true if value is not a number
- `boolean is_finite(float value)`  
returns true if value can be represented as a valid floating point number
- `boolean is_infinite(float value)`  
returns true if value cannot be represented as a valid floating point number

Prepared by X. Wang

18

## Integers and Floats

---

- Random number generation
- PHP provides the function `rand()` to create a random number
- The `rand()` function has two version

```
integer rand()
```

```
// between 0 and getrandmax()
```

```
integer rand(integer min, integer max)
```

```
// between min and max (inclusive)
```