Вътрешни (вградени) функции в PHP

В тази тема:

1. Функции за обработка на низове
2. Функции за дата, време и календар
3. Функции за хеширане на пароли
4. Функции свързани с променливите и типовете на данните

1. Функции за обработка на низове

int strlen (string $string)

Връща дължината на низа string .

```
$str = 'abcdef';
echo strlen($str); // 6
```


int strpos (string $haystack , mixed $needle [, int $offset = 0])

Връща позицията на първата поява на needle в низа haystack .

haystack - Низът в който се извършва търсенето

needle - Ако needle не е низ, се преобразува в целочислен вид и се третира като код на символа.

offset - Укажете от кой знак в haystack да започне търсенето.

primer-7-01.php

```
$mystring = 'abcde';
$findme   = 'ab';
$pos = strpos($mystring, $findme);

// Използва се оператора ===.  Обикновеното сравняване с оператора ==
// няма да работи както се очаква, тъй като позицията на 'a' е нулевият (първи) знак.
if ($pos === false) {
   echo "Низът '$findme' не беше открит в низа '$mystring'";
} else {
   echo "Низът '$findme'  беше открит в низа '$mystring'";
   echo " и започва от позиция $pos";
}
```


stripos

Връща позицията на първата поява на подниз в низ, без да се отчита регистъра.

Синтаксисът е аналогичен на strops

primer-7-02.php

string **substr (string $string , int $start [, int $length])**

Връща част от string на базата на параметрите start и length.

string - входен низ.

start

- неотрицателно число - върнатата част от низа ще започва от позиция start , считано от началото на низа string , като броенето започва от 0
- отрицателно число, върнатата част от низа ще започва от позиция start , считано от края на низа

length

- при положително число - върнатата част от низа ще съдържа най-много length знака, като се започне от start
- при отрицателно число - представлява броя на знаците от края на string , които да бъдат изпуснати (след като началната позиция е била изчислена при отрицателна стойност на start).

primer-7-03.php

```
// положителни стойности за start и length
echo substr("abcdef", 1, 3);    // връща "bcd"

// отрицателна стойности за start
echo substr("abcdef", -2);    // връща "ef"

// отрицателна стойност на length
substr("abcdef", 2, -1);  // връща "cde"
```

int strcmp (string $str1 , string $str2)

Сравнява двоично два низа (чувствително към регистъра)

str1 - първият низ

str2 - вторият низ.

Връщани стойности:

< 0 ако str1 е по-малък от str2

> 0 ако str1 е по-голям от str2

0 ако са равни

string sprintf (string $format [, mixed $args [, mixed $...]])

Връща низ, създаден посредством форматиращ низ format

format – съставен от:

- директиви - нормални знаци (с изключение на %), които директно се копират в резултатния низ
- описатели на преобразуването, като всеки от тях се заменя с един параметър
  - *d* - аргументът се разглежда като цяло число, а се представя във вид на десетично число със знак.
  - *f* - аргументът се разглежда като число с плаваща запетая и се представя като число с плаваща запетая (в зависимост от локала).
  - *s* - аргументът се разглежда и представя като низ.

primer-7-04.php

```
$order_no = 1001;
$qty = 9;
$price = 20.50;
$amount = $qty * 20.50;
$format = "Поръчка %d: количество - %d, цена - %f, обща сума - %f";
echo sprintf($format, $order_no, $qty, $price, $amount);
```


mixed sscanf (string $str , string $format [, mixed &$...])

Интерпретира низа str в съответствие с формата format

Всеки интервал в низа за форматиране съответства на интервал в изходния низ. Това означава, че дори табулацията \t в низа за форматиран съответства на знак за интервл в изходния низ.

str - Входният низ, който ще се анализира.

format - Интерпретираният формат за str.

... - Незадължителен параметър, чрез който могат да се предават стойности по референция, които съдържат анализираните стойности.

Връщани стойности

Ако само тези два параметъра са предадени на функцията, тя ще върне масив. В противен случай, ако са предадени и незадължителните параметри, функцията ще върне броя на присвоените стойности.

primer-7-05.php

```
// връщане на сериен номер
list($serial) = sscanf("SN/2350001", "SN/%d");
// и датата на производство
$mandate = "Декември 02 2014";
list($month, $day, $year) = sscanf($mandate, "%s %d %d");
echo "Продуктът $serial е произведен на: $day - $month - $year";
```


mixed str_replace (mixed $search , mixed $replace , mixed $subject [, int &$count])

Замества всички срещания на търсения низ със заместващия низ

Функцията връща низ или масив с всички срещания на **search** в **subject** , заменени със стойността указана чрез **replace** .

primer-7-06.php

```
// замяна на всички срещания на 2010 с 2013
$str2013 = str_replace('2010','2013', $str2010);
echo $str2013;
```

string str_repeat (string $input , int $multiplier)

Повтаря низ.

Връща **input** повторен **multiplier** пъти.

input - Низът, който трябва да се повтори.

multiplier - Броят на повторенията на низа input .

```
echo str_repeat("-=", 10);
```

Ще изведе:

-=-=-=-=-=-=-=-=-=-=

md5

Връща хеш на стринг кодиран с md5 алгоритъм

string md5 (string $str [, bool $raw_output = false])

- **str** – стрингът, който ще се хешира
- **raw_output** – при true се връща двоичен низ от 16 символа

Връща: Хеш като 32-символно шестнайсетично число.

Забележка: MD5 алгоритъмът е бърз и ефективен, но не се препоръчва за кодиране на пароли.

Пример:

```
$str = 'apple';

if (md5($str) === '1f3870be274f6c49b3e31a0c6728957f') {
    echo "Искате ли зелена ябълка?";
}
```


crc32

Изчислява 32 битова контрола сума (CRC - Cyclic Redundancy Checksum). Използва се за проверка на интегритета на данните, които се предават.

int crc32 (string $str)

Пример:

```
$checksum = crc32("Това е примерен текст.");
printf("%u\n", $checksum);
```

2. Функции за дата, време и календар

string date (string $format [, int $timestamp])

Форматира местно време/дата

format - Формата на низа за дата, който ще се извежда

ден: d - 2 цифри с водещи нули; D – текстово представяне с 3 букви, j - ден от месеца без водещи нули;

месец: F - пълно текстово представяне, m - Цифрово представяне на месец, с водещи нули, n - Цифрово представяне на месец, без водещи нули;

година: Y – представяне с 4 цифри, y – представяне с 2 цифри

primer-7-07.php

```
echo "Днес е " . date("d.m.Y") . "<br>";        //Днес е 01.12.2014
echo "Текущото време е " . date("h:i:sa");   Текущото време е 08:59:48pm
```


bool checkdate (int $month , int $day , int $year)

Проверява валидността на дата, образувана от аргументите. Една дата се счита за валидна, ако всеки от параметрите е правилно дефиниран.

Връщани стойности

Връща TRUE ако подадената дата е валидна и FALSE - иначе.

primer-7-08.php

```
echo "Проверка на аргументите за дата: 12, 31, 2014";
if (checkdate(12, 31, 2014))
        echo " - ВЕРНИ!";
else
        echo " - ГРЕШНИ!";
echo "<br>";
echo "Проверка на аргументите за дата: 13, 31, 2014";
if (checkdate(13, 31, 2014))
        echo " - ВЕРНИ!";
else
        echo " - ГРЕШНИ!";
```


int cal_days_in_month (int $calendar , int $month , int $year)

Връща броя дни в месец за посочени година и месец и календар

primer-7-09.php

```
$month = 12;
$year = 2014;

$num = cal_days_in_month(CAL_GREGORIAN, $month, $year); // 31
echo "Броят дни през $month.$year са $num";
```

3. Функции за хеширане на пароли

password_hash

Хешира пароли използвайки сигурен алгоритъм.

string password_hash (string $password , integer $algo [, array $options])

password – потребителската парола

algo – алгоритъмът, който ще се използва за хеширане на паролата:

algo :

- **PASSWORD_DEFAULT** – използва bcrypt алгоритъм, който е подразбиращ се в PHP 5.5.0. Проектиран за да се променя през времето, като по-нови и сигурни алгоритми ще се добавят в PHP. По-тази причина дължината на резултатния стринг ще се променя през времето. Резултатът може да се съхранява в колона на таблица от БД с дължина около 255 символа;
- **PASSWORD_BCRYPT** – използва CRYPT_BLOWFISH алгоритъм. Резултатът винаги е 60 символа.

Връщан резултат:

Стринг с хешираната парола или FALSE при грешка.

Пример: Хеширане на парола с подразбиращия се алгоритъм и **ASSWORD_BCRYPT**.

primer-7-10.php

```
$password_text = "parola123!";

$result_hash_default = password_hash($password_text,
PASSWORD_DEFAULT);
$result_hash_BCRYPT = password_hash($password_text,
PASSWORD_BCRYPT);

echo "Хеширане на $password_text, с DEAFAULT и
BCRYPT:"."<p>";
echo $result_hash_default;
echo "<br>";
echo $result_hash_BCRYPT;
```


password_verify

Проверява дали парола съответства на хеша

boolean password_verify (string $password , string $hash)

- **password** – потребителска парола
- **hash** – хеш създаден с password_hash()

Връщан резултат:

- TRUE или FALSE

Пример: Проверка на парола.

primer-7-11.php

```
$password_text = "parola123!";
$password_hash_result = password_hash($password_text,
PASSWORD_DEFAULT);

$password_user = "parola12345";

if (password_verify($password_user,
$password_hash_result)) {
    echo 'Валидна парола!';
} else {
    echo 'Невалидна парола!';
}
```


4. Функции свързани с променливите и типовете на данните

serialize

Генерира удобно за съхранение представяне на стойност

string serialize (mixed $value)

Поддържа всички типове, освен resource.

Връщани стойности:

Връща низ, съдържащ byte-stream

За преобразуване на сериализиран низ обратно към PHP стойност, се използва функцията unserialize().

unserialize

Създава PHP стойност от подходящо за съхранение представяне на стойност

mixed unserialize (string $str)

Взема дадена сериализирана променлива и превръща стойността й отново в PHP стойност.

Пример: Сериализиране на масив и записване стойността в променлива стринг, след което се преобразува отново в масив с десериализиране.

primer-7-12.php

```
$fruits = array (
    "fruits"  => array("a" => "orange", "b" => "banana", "c" =>
"apple"));

echo "Масивът преди сериализацията"."<br>";
print_r($fruits);
echo "<p>";

$fruits_str_serialize = serialize($fruits);
echo "Стрингът след сериализацията"."<br>";
echo $fruits_str_serialize;
echo "<p>";

$fruits_normal =  unserialize($fruits_str_serialize);
echo "Масивът след десериализацията"."<br>";
print_r($fruits_normal);
```


Използвани източници:

- Ръководство по PHP: Справочник на функциите. http://php.net/manual/bg/funcref.php

Допълнителна литература:

- **Making a login form using PHP.** http://www.html-form-guide.com/php-form/php-login-form.html