

XỬ LÝ DATETIME TRONG PHP

NGƯỜI CHIA SẺ: TẠ HOÀNG AN

XỬ LÝ DATETIME TRONG PHP

► Hàm time()

Hàm time() sẽ lấy timestamp của thời điểm hiện tại. Hàm sẽ trả về số giây tính từ thời điểm 00:00:00 1/1/1970 đến thời điểm hiện tại.

Cú pháp:

```
time();
```

► Hàm strtotime()

Hàm strtotime() sẽ phân tích bất kỳ chuỗi thời gian bằng tiếng anh thành dạng timestamp của thời gian đó.

Cú pháp:

```
strtotime($str);
```

XỬ LÝ DATETIME TRONG PHP

► Hàm strtotime() (tiếp)

Ví dụ:

```
$str = '21:30:45 16-May-2020';  
echo strtotime($str);  
//Kết quả: 1589679045
```

► Hàm strftime()

Hàm ²strftime() sẽ xuất ra chuỗi thời gian theo đúng với định dạng truyền vào. Kết quả xuất ra tùy thuộc vào việc cài đặt ngôn ngữ (locale settings).

Cú pháp:

```
strftime($format, $timestamp);
```

XỬ LÝ DATETIME TRONG PHP

► Hàm strftime() (tiếp)

Trong đó:

- \$format là định dạng thời gian muốn xuất ra.
- \$timestamp là timestamp của thời gian nào đó.

Ví dụ:

```
echo strftime("%A %d-%B-%Y", 1589679045);
```

XỬ LÝ DATETIME TRONG PHP

► Hàm mktime()

Cú pháp:

```
mktime( $hour, $minute, $second, $month, $day, $year);
```

Trong đó:

- \$hour là tham số chỉ giờ.
- \$minute là tham số chỉ phút.
- \$second là tham số chỉ giây.
- \$month là tham số chỉ tháng.
- \$day là tham số chỉ ngày.
- \$year là tham số chỉ năm.

XỬ LÝ DATETIME TRONG PHP

► Hàm mktime() (tiếp)

Ví dụ:

```
$str = mktime(21,30,45,5,19,2020);  
echo $str;
```

► Hàm microtime()

Hàm microtime() sẽ lấy timestamp của thời điểm hiện tại bao gồm cả microseconds.

Cú pháp:

```
microtime ($is_float);
```

\$is_float mang giá trị TRUE hàm sẽ trả về số thuộc kiểu float, nếu không truyền hoặc mang giá trị FALSE, hàm sẽ trả về chuỗi theo dạng "microseconds second".

XỬ LÝ DATETIME TRONG PHP

► Hàm localtime()

Hàm localtime() sẽ lấy thông tin về thời gian bao gồm giờ, phút, giây, ngày, tháng, năm... dựa vào timestamp của thời điểm đó.

Cú pháp:

```
localtime ($timestamp, $is_associative);
```

Trong đó:

- ²\$timestamp là timestamp của thời điểm muốn lấy thông tin. Nếu không truyền nó sẽ lấy timestamp của thời điểm hiện tại.
- \$is_associative là tham số. Nếu không truyền vào hoặc mang giá trị FALSE, hàm sẽ trả về mảng liên tục, nếu \$is_associative mang giá trị TRUE, hàm sẽ trả về một mảng chi tiết với khóa tương minh như sau:

XỬ LÝ DATETIME TRONG PHP

► Hàm localtime() (tiếp)

- ❖ "tm_sec" - số giây.
- ❖ "tm_min" - số phút.
- ❖ "tm_hour" - số giờ.
- ❖ "tm_mday" - ngày.
- ❖ "tm_mon" - tháng.
- ❖ "tm_year" - năm thứ bao nhiêu tính từ 1900.
- ❖ "tm_wday" - ngày thứ mấy trong tuần, từ 0-6.
- ❖ "tm_yday" - ngày thứ bao nhiêu trong năm.
- ❖ "tm_isdst" - daylight savings time có hiệu lực không, giá trị lớn hơn không là có, bằng 0 là không, giá trị âm là không xác định.

XỬ LÝ DATETIME TRONG PHP

► Hàm idate()

Hàm idate() sẽ trả về một số nguyên được định dạng theo định dạng cho trước dựa vào timestamp của một thời điểm nào đó. Nếu timestamp không được truyền vào, hàm idate() sẽ lấy timestamp của thời điểm hiện tại.

Cú pháp:

```
idate( $format, $timestamp);
```

Trong đó:

- \$format là định dạng của giá trị muốn lấy(có thể là thứ, ngày, tháng. v.v.).
- \$timestamp sẽ là timestamp của một thời điểm. Nếu không được truyền vào, \$timestamp sẽ mang giá trị timestamp của thời điểm hiện tại.

XỬ LÝ DATETIME TRONG PHP

► Hàm gmmktime()

Hàm gmmktime() sẽ lấy timestamp của một thời điểm nào đó. Tuy nhiên, khác với mktime() hàm gmmktime() sẽ lấy timestamp ở múi giờ số 0(GMT).

Cú pháp:

```
gmmktime( $hour, $minute, $second, $month, $day, $year);
```

Trong đó:

- \$hour là tham số chỉ giờ.
- \$minute là tham số chỉ phút.
- \$second là tham số chỉ số giây.
- \$month là tham số chỉ tháng.
- \$day là tham số chỉ ngày.
- \$year là tham số chỉ năm.

XỬ LÝ DATETIME TRONG PHP

► Hàm gmdate()

Hàm gmdate() sẽ xuất ra chuỗi thời gian theo định dạng được truyền vào và số timestamp. Tuy nhiên, khác với hàm date(), nó sẽ lấy thời gian GMT(thời gian tại múi giờ số 0).

Cú pháp:

```
gmdate( $format, $timestamp);
```

Trong đó:

- \$format là định dạng thời gian muốn xuất ra.
- \$timestamp là số nguyên chỉ timestamp của một thời gian nào đó. Nếu không truyền tham số này, hàm sẽ lấy timestamp của thời gian hiện tại.

XỬ LÝ DATETIME TRONG PHP

► Hàm getdate()

Hàm getdate() sẽ lấy thông tin về thời gian. Cụ thể hàm sẽ lấy thông tin về thứ, ngày, tháng, năm, giờ, phút, giây ... của thời gian hiện tại hoặc thời gian nào đó tùy thuộc vào tham số người dùng truyền vào.

Cú pháp:

```
getdate ($timestamp)
```

\$timestamp là một số nguyên Unix timestamp. Nếu không truyền \$timestamp hàm sẽ lấy timestamp của thời gian hiện tại.

Ví dụ:

```
$info = getdate();  
echo "<pre>";  
print_r($info);  
echo "</pre>";
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date()

Hàm date() sẽ xuất ra một chuỗi thời gian dựa theo định dạng được truyền vào và số nguyên timestamp.

Cú pháp:

```
date( $format, $timestamp);
```

Trong đó:

- \$format² là định dạng thời gian mà hàm sẽ trả về.
- \$timestamp là một số nguyên Unix timestamp. nếu không truyền tham số này, nó sẽ lấy timestamp của thời điểm hiện tại.

XỬ LÝ DATETIME TRONG PHP

► Hàm date() (tiếp)

Ví dụ:

```
echo date("F j, Y, g:i a")."<br />";  
echo date("m.d.y")."<br />";  
echo date("j, n, Y")."<br />";  
echo date("Ymd")."<br />";  
echo date('h-i-s, j-m-y, it is w Day')."<br />";  
echo date("\i\t \i\s \t\h\e jS \d\a\y.')."<br />";  
echo date("D M j G:i:s T Y")."<br />";  
echo date('H:m:s \m \i\s\ \m\o\n\t\h')."<br />";  
echo date("H:i:s")."<br />";  
echo date("Y-m-d H:i:s")."<br />";
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_timezone_set()`

Hàm `date_timezone_set()` sẽ thiết lập timezone cho đối tượng thời gian được truyền vào

Cú pháp:

```
date_timezone_set( $object, $timezone);
```

Trong đó:

- `$object` là đối tượng cần thiết lập lại timezone.
- `$timezone` là timezone sẽ thiết lập cho đối tượng `$object`.

Ví dụ:

```
$date = date_create();  
date_timezone_set($date, timezone_open('Europe/London'));  
echo date_format($date, 'Y-m-d H:i:sP');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_timezone_get()`

Hàm `date_timezone_get()` sẽ lấy về timezone của đối tượng thời gian được truyền vào

Cú pháp:

```
date_timezone_get( $object);
```

\$object là đối tượng thời gian cần lấy thông tin timezone

► Hàm `date_timestamp_set()`

Hàm `date_timestamp_set()` sẽ đặt lại thời gian của đối tượng mốc thời gian dựa vào unix timestamp.

Cú pháp:

```
date_timestamp_set( $object, $timestamp);
```


XỬ LÝ DATETIME TRONG PHP

► Hàm `date_timestamp_set()` (tiếp)

Trong đó:

- `$object` là đối tượng thời gian.
- `$timestamp` là số giây từ 00:00:00 1/1/1970 đến một mốc thời gian muốn đặt lại cho `$object`.

Ví dụ:

```
$date = date_create();  
  
date_timestamp_set($date, 1493899800);  
  
echo date_format($date, 'Y-m-d H:i:s');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_timestamp_get()

Hàm date_timestamp_get() sẽ lấy số giây tính từ 0 giờ 0 phút 0 giây ngày 1 tháng 1 năm 1970 cho đến mốc thời gian của đối tượng.

Cú pháp:

```
date_timestamp_get( $object);
```

\$object là đối tượng mốc thời gian

Ví dụ:

```
$date = date_create();  
date_time_set($date, 15, 40);  
echo date_timestamp_get($date);
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_time_set()`

Hàm `date_time_set()` sẽ đặt lại thời gian cho đối tượng thời gian đã khởi tạo trước đó. Cụ thể hàm `date_time_set()` sẽ đặt lại giờ phút giây được truyền vào cho đối tượng mốc thời gian.

Cú pháp:

```
date_time_set( $object, $hour, $minute, $second);
```

Trong đó:

- `$object` là đối tượng thời gian đã khởi tạo trước đó.
- `$hour` là tham số chỉ giờ sẽ được đặt lại trong đối tượng `$object`.
- `$minute` là tham số chỉ phút sẽ được đặt lại trong đối tượng `$object`.
- `$second` là tham số chỉ giây sẽ được đặt lại trong đối tượng `$object`. Nếu không truyền `$second` sẽ mang giá trị 0.

XỬ LÝ DATETIME TRONG PHP

► Hàm date_time_set() (tiếp)

Ví dụ:

```
$date = date_create();  
date_time_set($date, 15, 45);  
echo date_format($date, 'Y-m-d H:i:s');
```

► Hàm date_sub()

Hàm date_sub() sẽ trừ đi một khoảng thời gian nhất định từ đối tượng mốc thời gian truyền vào ban đầu.

Cú pháp:

```
date_sub( $object, $interval);
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_sub() (tiếp)

Trong đó:

- \$object là đối tượng mốc thời gian.
- \$interval là một đối tượng quy định khoảng thời gian sẽ trừ đi tính từ mốc thời gian \$object.

Ví dụ:

```
$date = date_create('2020-05-1');  
date_sub($date, date_interval_create_from_date_string('1  
days'));  
echo date_format($date, 'Y-m-d');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_parse()

Hàm date_parse() sẽ lấy thông tin chi tiết của chuỗi thời gian được truyền vào.

Cú pháp:

```
date_parse( $str);
```

\$str là chuỗi thời gian được chấp nhận bởi hàm strtotime()

Ví dụ:

```
$date = date_parse("2020-04-30 10:00:00");  
$date = date_parse("2020-11-12 10:00:00.5 +2 month +1 week  
+5 hour");
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_parse_from_format()

Hàm date_parse_from_format() sẽ lấy thông tin của mốc thời gian truyền vào. Hàm sẽ căn cứ vào kiểu định dạng của chuỗi thời gian để lấy thông tin phù hợp.

Cú pháp:

```
date_parse_from_format ($str, $format);
```

Trong đó:

- \$str là chuỗi thời gian cần lấy thông tin.
- \$format là định dạng của \$str.

Ví dụ:

```
$date = "24/08/2020 13:35:22";  
echo "<pre>";  
print_r(date_parse_from_format("d/m/Y H:i:s", $date));  
echo "</pre>";
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_offset_get()

Hàm date_offset_get() sẽ trả về múi giờ tính theo giây dựa theo giờ UTC

Cú pháp:

```
date_offset_get( $object);
```

\$object là đối tượng thời gian (Được tạo từ hàm date_create() hoặc date_create_from_format())

Ví dụ:

```
$date = date_create();  
echo date_offset_get($date)/3600;
```


XỬ LÝ DATETIME TRONG PHP

► Hàm date_modify()

Hàm date_modify() sẽ thay đổi mốc thời gian của đối tượng thời gian. Cụ thể, hàm sẽ thêm bớt một khoảng thời gian nào đó vào một mốc thời gian.

Cú pháp:

```
date_modify( $object, $modify);
```

Trong đó:

- \$object là đối tượng thời gian.
- \$modify là khoảng thời gian bạn muốn thêm hoặc bớt so với mốc thời gian của \$object.

Ví dụ:

```
$date = date_create();  
date_modify($date, '+41 day');  
echo date_format($date, 'd-m-Y');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_interval_format()

Hàm date_interval_format() sẽ định dạng một khoảng thời gian theo một định dạng nhất định.

Cú pháp:

```
date_interval_format ($str);
```

Ví dụ:

```
$interval = date_interval_create_from_date_string('34 day');  
echo date_interval_format($interval, '%d days');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_interval_create_from_date_string()`

Hàm `date_interval_create_from_date_string()` sẽ tạo đối tượng khoảng thời gian (`DateInterval`) từ một chuỗi liên quan

Cú pháp:

```
date_interval_create_from_date_string( $string);
```

\$string là chuỗi thời gian

```
$interval = date_interval_create_from_date_string('3 day');  
echo '<pre>';  
print_r($interval);  
echo '</pre>';
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_format()

Hàm date_format() sẽ định dạng một đối tượng thời gian dựa vào định dạng được truyền vào.

Cú pháp:

```
date_format( $object, $format);
```

Trong đó:

- \$object là đối tượng thời gian cần định dạng.
- \$format là định dạng muốn định dạng cho \$object.

XỬ LÝ DATETIME TRONG PHP

► Hàm date_diff()

Hàm date_diff() sẽ so sánh hai đối tượng thời gian với nhau

Cú pháp:

```
date_diff($date1, $date2);
```

Trong đó:

\$date1, \$date2 là hai đối tượng thời gian cần so sánh

Ví dụ:

```
$datetime1 = date_create('2010-04-30');  
$datetime2 = date_create('2020-04-30');  
$interval = date_diff($datetime1, $datetime2);  
echo $interval->format('%R%y years');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_default_timezone_set()`

Hàm `date_default_timezone_set()` sẽ đặt giá trị timezone mặc định cho hệ thống, tất cả các hàm về xử lý thời gian sẽ sử dụng timezone này.

Cú pháp:

```
date_default_timezone_set ($timezone);
```

\$timezone²: Chuỗi timezone. Xem cú pháp timezone tại đây:

<https://www.php.net/manual/en/timezones.php>

XỬ LÝ DATETIME TRONG PHP

► Hàm `date_default_timezone_get()`

Hàm `date_default_timezone_get()` sẽ lấy timezone mặc định của hệ thống

Cú pháp:

```
date_default_timezone_get();
```

Ví dụ: 📄

```
$timezone = date_default_timezone_get();  
echo $timezone;  
//Kết quả: Asia/Ho_Chi_Minh
```

XỬ LÝ DATETIME TRONG PHP

► Hàm date_create()

Hàm date_create() sẽ tạo một đối tượng thời gian

Cú pháp:

```
date_create( $time);
```

\$time là chuỗi quy định thời gian, mặc định \$time sẽ là thời gian hiện tại

```
$date = date_create('30-04-2020');  
echo "<pre>";  
print_r($date);  
echo "</pre>";
```


XỬ LÝ DATETIME TRONG PHP

► Hàm `date_create_from_format()`

Hàm `date_create_from_format()` sẽ tạo một đối tượng thời gian dựa vào chuỗi thời gian và định dạng tương ứng được truyền vào

Cú pháp:

```
date_create_from_format( $format, $string);
```

Trong đó:

- `$format` là định dạng thời gian giống với định dạng của chuỗi `$string`.
- `$string` là chuỗi thời gian.

XỬ LÝ DATETIME TRONG PHP

► Hàm date_add()

Hàm date_add() sẽ cộng thêm một khoảng thời gian nhất định vào một đối tượng thời gian nào đó.

Cú pháp:

```
date_add( $object, $interval);
```

Trong đó:

- \$object là đối tượng thời gian, \$object là kết quả trả về của hàm date_create().
- \$interval² là khoảng thời gian cần thêm, Thực chất \$interval cũng là một đối tượng DateInterval.

Ví dụ:

```
$date = date_create('2020-04-30');  
date_add($date, date_interval_create_from_date_string('10 days'));  
echo date_format($date, 'Y-m-d');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `timezone_open()`

Hàm `timezone_open()` sẽ khởi tạo một đối tượng `timezone`

Cú pháp

```
timezone_open( $timezone);
```

Ví dụ

```
$timezone = timezone_open('Europe/London');
```

XỬ LÝ DATETIME TRONG PHP

► Hàm `timezone_name_get()`

Hàm `timezone_name_get()` sẽ lấy tên của múi giờ từ đối tượng `DateTimeZone` truyền vào.

Cú pháp:

```
timezone_name_get ($object);
```

\$object: Đối tượng `DateTimeZone`

XỬ LÝ DATETIME TRONG PHP

► Hàm `timezone_location_get()`

Hàm `timezone_location_get()` sẽ lấy thông tin về múi giờ từ đối tượng `DateTimeZone` được truyền vào. Thông tin sẽ bao gồm mã quốc gia, kinh độ, vĩ độ và chú thích.

Cú pháp:

```
timezone_location_get ($object);
```

`$object` là đối tượng `DateTimeZone` nó sẽ là kết quả trả về của hàm `timezone_open()`

Ví dụ:

```
$timezone = date_default_timezone_get();  
$arr = timezone_location_get(timezone_open($timezone));  
echo "<pre>";  
print_r($arr);  
echo "</pre>";
```

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern, layered effect on the right side of the image.

Happy Coddling!