

FECHAS Y FUNCIONES EN PHP

FECHAS CON PHP



- Función TIME y DATE
- Estas dos funciones nos permitirán mostrar o capturar la fecha y hora, lo cual tiene una gran cantidad de aplicaciones. Por ejemplo, supón que tienes una tienda on-line y para cada operación de venta tienes que registrar la fecha y hora de la venta. Estas funciones resultarán útiles para ello.

- `date_default_timezone_get()`
- `date_default_timezone_set()`
- `date()`
- `time()`
- `strtotime()`
- `setlocale()`
- `strftime()`
- `mktime()`
- `getdate()`

date_default_timezone_get()



- Cuando trabajamos con fechas en PHP siempre hay que tener en cuenta si la zona horaria del sistema es correcta. Para obtener dicha información tenemos la función `date_default_timezone_get()` que nos devolverá una cadena de texto con dicha zona.

```
<?php
    $zonahoraria = date_default_timezone_get();
    echo 'Zona horaria predeterminada: ' . $zonahoraria;
?>
```

date_default_timezone_set()



- Esta función de PHP establece la zona horaria por defecto en todas las funciones de fecha que usemos en nuestro script PHP.

```
<?php
    //Establecemos zona horaria por defecto
    date_default_timezone_set('America/Lima');
    //preguntamos la zona horaria
    $zonahoraria = date_default_timezone_get();
    echo 'Zona horaria predeterminada: ' . $zonahoraria;
?>
```

date()

- Básicamente se puede decir que nos devuelve formateada la fecha local del sistema para obtener el día, mes, semana, año y hora actual.

```
<?php
// Día del mes con 2 dígitos, y con ceros iniciales, de 01 a 31
echo date("d");
// Día del mes, sin ceros iniciales, de 1 a 31
echo date("j");
// Día de la semana en inglés, con 3 letras, de Mon a Sun
echo date("D");
// Día de la semana en inglés, de Sunday a Saturday
echo date("l");
// del día de la semana, desde 1 (lunes) hasta 7 (domingo)
echo date("N");
// Sufijo del día del mes con 2 caracteres --> st, nd, rd o th
echo date("S");
// Número entero que representa el día de la semana, de 0 (dom) a 6 (sab)
echo date("w");
// Día del año, de 0 a 365
echo date("z");
?>
```

time()

- Obtener la hora actual con PHP time() es muy fácil, esta función de PHP nos devuelve en segundos el tiempo transcurrido desde la época Unix (1 de Enero de 1970 00:00:00 GMT).
- La función PHP date() devuelve la hora de una manera más 'reconocible', pero time() al trabajar con números enteros nos facilita mucho más el trabajo y operaciones entre fechas almacenadas en BBDD.

```
<?php
    date_default_timezone_set('America/Lima');

    echo date("m/d/Y h:i:s a", time());
?>
```

strtotime()



- La función de PHP strtotime() convierte una cadena con formato de fecha en otra con formato de fecha Unix.

strtotime()



```
<?php
    date_default_timezone_set('America/Lima');

    echo "Fecha actual Unix con parámetro 'now' -->" . strtotime("now") . "<br>";
    echo "Fecha pasada '15 May 2015' -->" . strtotime("15 May 2015") . "<br>";
    echo "Fecha actual + 1 hora -->" . strtotime("+1 hours") . "<br>";
    echo "Fecha actual + 1 día -->" . strtotime("+1 day") . "<br>";
    echo "Fecha actual + 1 semana -->" . strtotime("+1 week") . "<br>";
    echo "Fecha actual + 1 mes -->" . strtotime("+1 month") . "<br>";
    echo "Fecha actual + 1 año -->" . strtotime("+1 year") . "<br>";
    echo "Fecha actual + 1 año + 1 mes + 1 semana + 1 día + 1 hora -->" .
    strtotime("+1 year +1 month +1 week +1 day +1 hours") . "<br>";
    echo "Próximo lunes -->" . strtotime("next monday") . "<br>";
    echo "El pasado lunes -->" . strtotime("last monday") . "<br>";
    echo "Próxima semana -->" . strtotime("next week") . "<br>";
    echo "Próximo mes -->" . strtotime("next month") . "<br>";
    echo "Próximo año -->" . strtotime("next year") . "<br>";
?>
```

setlocale()

- Con la función de PHP setlocale() podemos definir o establecer información local en nuestro servidor. Dicha información puede ser el lenguaje, moneda, hora u otras informaciones específicas de la zona geográfica con la que trabajamos en nuestros script.

```
<?php
    date_default_timezone_set('America/Lima');

    //Establecer la información de hora local en castellano de España
    setlocale(LC_TIME,"es_ES");
    echo strftime("%A %d de %B de %Y");

?>
```

strftime()



- la función de PHP strftime() para el formateo local de fechas y horas. La información local se puede establecer en el script con la función setlocale().

strftime()

DÍA

- **%a**: Representación abreviada el día. De 'dom' a 'sáb'.
- **%A**: Representación completa del día. De 'domingo' a 'sábado'.
- **%d**: Día del mes incluyendo ceros iniciales. De '01' a '31'.
- **%e**: Día del mes sin ceros iniciales. De '1' a '31'.
- **%j**: Día del año con ceros iniciales. De '001' a '366'.
- **%u**: Día de la semana ISO-8601. De '1' a '7'.
- **%w**: Día de la semana. De '0' a '6'.

strftime()



SEMANA

- **%U:** Número de la semana, comenzando la primera semana en el primer domingo.
- **%V:** Número de la semana ISO-8601, la primera semana debe contener al menos 4 días y el lunes como primer día. De '01' a '53'.
- **%W:** Representación numérica de la semana del año, comenzando con el primer domingo como la primera semana.

MES

- **%b:** Nombre del mes abreviado. De 'ene' a 'dic'.
- **%B:** Nombre del mes completo. De 'enero' a 'diciembre'.
- **%h:** Igual que **%b**.
- **%m:** Representación numérica del mes con ceros iniciales. De '01' a '12'.

strftime()



AÑO

- **%C:** Representación del número de siglo. Año dividido entre 100 cogiendo la parte entera.
- **%G:** Representación numérica del año en 4 dígitos formato ISO-8601. Por ejemplo '2015'
- **%g:** Representación numérica del año en 2 dígitos formato ISO-8601. Por ejemplo '15' para '2015'.
- **%Y:** Representación numérica del año en 4 dígitos. Por ejemplo '2015'.
- **%y:** Representación numérica del año en 2 dígitos. Por ejemplo '15' para 2015.

HORA

- **%H:** Representación numérica de la hora en 2 dígitos con formato 24h. De '00' a '23'.
- **%I:** Representación numérica de la hora en 2 dígitos con formato 12h. De '01' a '12'.
- **%M:** Representación numérica de los minutos en 2 dígitos. De '00' to '59'.
- **%P:** Representación de 'AM' o 'PM' dependiendo la hora dada. Por ejemplo 'AM' para las 00:51.
- **%p:** Representación de 'am' o 'pm' dependiendo la hora dada. Por ejemplo 'am' para las 00:51.
- **%S:** Representación numérica de los segundos en 2 dígitos. De '00' to '59'.

strftime()

```
<?php
    date_default_timezone_set('America/Lima');

    //Establecer la información local en castellano de España
    setlocale(LC_TIME,"es_ES");

    echo strftime("Hoy es %A y son las %H:%M");
    echo strftime("El año es %Y y el mes es %B");

?>
```

Hoy es Thursday y son las 18:25
El año es 2017 y el mes es June

mktime()

- la función de PHP mktime(), su funcionalidad es la de obtener una fecha Unix a partir de los parámetros pasados.

```
<?php
    date_default_timezone_set('America/Lima');

    //Establecer la información local en castellano de España
    setlocale(LC_TIME,"es_ES");

    echo "Ayer en formato Unix es: " .
    mktime(0, 0, 0, date("m") , date("d")-1, date("Y"));

    echo "Mañana en formato Unix es: " .
    mktime(0, 0, 0, date("m") , date("d")+1, date("Y"));

    echo "El mes que viene en formato Unix es: " .
    mktime(0, 0, 0, date("m")+1, date("d"), date("Y"));
?>
```


getdate()

- La función de PHP `getdate()` para obtener la información de fecha y hora actual del sistema, también se puede pasar como parámetro una fecha con formato timestamp.
- `getdate()` devuelve un array asociativo con la información en formato timestamp.

```
<?php
    date_default_timezone_set('America/Lima');

    //Establecer la información local en castellano de España
    setlocale(LC_TIME,"es_ES");

    $fechaactual = getdate();
    print_r($fechaactual);
    echo "Hoy es: $fechaactual[weekday], ".
        "$fechaactual[mday] de $fechaactual[month] de $fechaactual[year]";
?>
```

getdate()

```
Array ( [seconds] => 16 [minutes] => 38 [hours] => 18 [mday] => 29 [wday] => 4  
[mon] => 6 [year] => 2017 [yday] => 179 [weekday] => Thursday [month] => June [0] => 1498779496 )
```

Los valores del array son los siguientes:

- **[seconds]** – Segundos -> 0-59
- **[minutes]** – Minutos -> 0-59
- **[hours]** – Horas -> 0-23
- **[mday]** – Día del mes -> 1-31
- **[wday]** – Día de la semana -> 0(Domingo) – 6(Sábado)
- **[mon]** – Mes -> 1-12
- **[year]** – Año -> 4 dígitos
- **[yday]** – Día del año -> 0-365
- **[weekday]** – Día de la semana en formato texto -> Sunday-Saturday
- **[month]** – Nombre del mes -> January-December
- **[0]** – Segundos Unix -> -2147483648 hasta 2147483647

FUNCIONES EN PHP

- Una función es un conjunto de instrucciones que a lo largo del programa van a ser ejecutadas multitud de veces. Es por ello, que este conjunto de instrucciones se agrupan en una función. Las funciones pueden ser llamadas y ejecutadas desde cualquier punto del programa.

```
function nombre (parámetro1, parámetro2, ..., parámetroN) {  
    instrucción1  
    instrucción2  
    .  
    .  
    .  
    instrucciónN  
}
```

JQUERY

Página oficial



Your donations help fund the continued development and growth of jQuery.

SUPPORT THE PROJECT

[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)

Search



Lightweight Footprint

Only 32kB minified and gzipped. Can also be included as an AMD module



CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation



Cross-Browser

[Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more](#)



Download jQuery
v3.2.1

[View Source on GitHub →](#)

[How jQuery Works →](#)

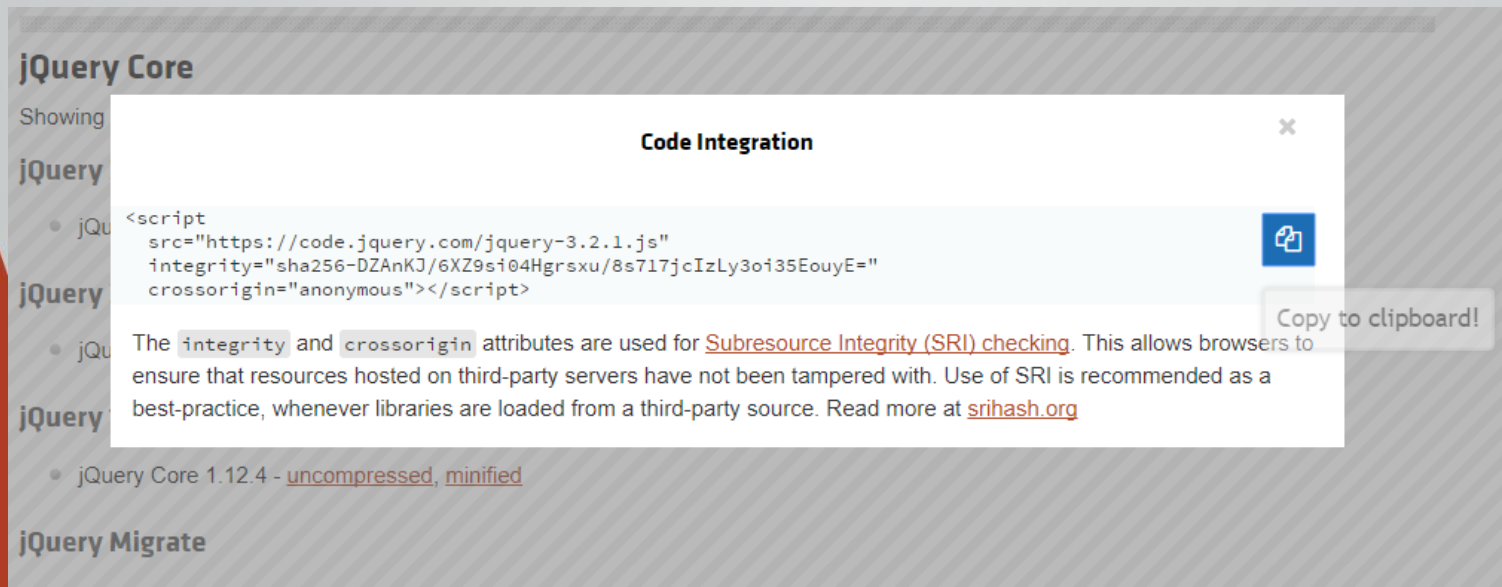
What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)

- `<script src="https://code.jquery.com/jquery-3.2.1.js" integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jclzLy3oi35EouyE=" crossorigin="anonymous"></script>`

A screenshot of the jQuery Core website with a "Code Integration" dialog box open. The dialog shows the jQuery script tag with integrity and crossorigin attributes. A "Copy to clipboard!" button is visible next to the code. Below the code, a text block explains that these attributes are used for Subresource Integrity (SRI) checking to ensure resources have not been tampered with. The background shows the jQuery Core navigation menu and a list of links including "jQuery Core 1.12.4 - uncompressed, minified" and "jQuery Migrate".

jQuery Core

Showing

jQuery

- jQuery
- jQuery
- jQuery
- jQuery

jQuery Migrate

Code Integration [X]

```
<script
src="https://code.jquery.com/jquery-3.2.1.js"
integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jclzLy3oi35EouyE="
crossorigin="anonymous"></script>
```

[Copy icon]

Copy to clipboard!

The `integrity` and `crossorigin` attributes are used for [Subresource Integrity \(SRI\) checking](#). This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source. Read more at [srihash.org](#)

- jQuery Core 1.12.4 - [uncompressed](#), [minified](#)

Práctica de selectores y jquery

- Esto se realizará con supervisión del ponente.