

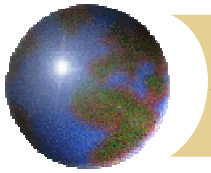
Desarrollo de Portales WEB *Lenguaje PHP*

Gonzalo Anchante Hurtado
Docente

INSTITUTO SUPERIOR TECNOLÓGICO



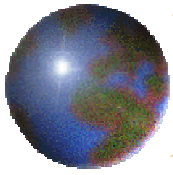
R.M. N° 420 - 94 - ED



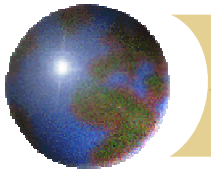
Crear una constante

- Una constante es objeto que almacena un valor de forma permanente, no cambia el valor de su contenido. Se declara de la forma siguiente:

```
<?php  
define("CONSTANTE", "hello world.");  
echo CONSTANTE;  
?>
```



Funciones de fecha



Función date

- Las funciones **date()** y **gmdate()** sirven para dar el formato deseado a una cadena conteniendo una fecha. Si no se pasa ninguna cadena a estas funciones, se da el formato al tiempo actual (calculado internamente con **time()**).

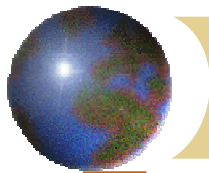
La única diferencia entre ambas es que **gmdate()** indica la hora del meridiano de Greenwich.



Función date

Trabajando con fecha de día.

modificadores	ejemplo	resultado
d - día del mes, número, dos dígitos	<code><?php \$ahora_es = date(d); ?></code>	<code>\$ahora_es = 23</code>
j - día del mes, número, dos dígitos sin cero inicial	<code><?php \$ahora_es = date(j); ?></code>	<code>\$ahora_es = 23</code>
D - día semana abreviado letras	<code><?php \$ahora_es = date(D); ?></code>	<code>\$ahora_es = Fri</code>
l - día de la semana, en texto, completo (ele)	<code><?php \$ahora_es = date(l); ?></code>	<code>\$ahora_es = Friday</code>
w - día de la semana, en numero (comienza en 0 = domingo)	<code><?php \$ahora_es = date(w); ?></code>	<code>\$ahora_es = 5</code>



Función date

```
<?php
```

```
$ahora=date (d) ;
```

```
echo "$ahora<br>";
```

```
?>
```

23

Fri

```
<?php
```

```
$ahora=date (D) ;
```

```
echo "$ahora<br>";
```

Friday

```
$ahora=date (l) ;
```

```
echo $ahora;
```

Terminado

```
?>
```



Función date

Trabajando con fecha de mes.

modificadores	ejemplo	resultado
m - mes, número, dos dígitos	<code><?php \$ahora_es = date(m); ?></code>	<code>\$ahora_es = 03</code>
n - mes, número, dos dígitos sin cero inicial	<code><?php \$ahora_es = date(n); ?></code>	<code>\$ahora_es = 3</code>
M - mes, tres letras	<code><?php \$ahora_es = date(M); ?></code>	<code>\$ahora_es = Mar</code>
F - mes, texto completo	<code><?php \$ahora_es = date(F); ?></code>	<code>\$ahora_es = March</code>
t - total dias en el mes	<code><?php \$ahora_es = date(t); ?></code>	<code>\$ahora_es = 31</code>



Función date

Trabajando con fecha de mes.

```
<?php
```

```
$ahora=date (M) ;
```

```
echo "MES CORTO: $ahora<br>" ;
```

```
$ahora=date (F) ;
```

```
echo "MES COMPLETO: $ahora<br>" ;
```

```
$ahora=date (t) ;
```

```
echo "TOTAL DE DIAS DEL MES: $ahora<br>" ;
```

```
?>
```




Función date

Trabajando con fecha de Año.

modificadores	ejemplo	resultado
Y - año, cuatro cifras	<code><?php \$ahora_es = date(Y); ?></code>	<code>\$ahora_es = 2007</code>
y - año, dos cifras	<code><?php \$ahora_es = date(y); ?></code>	<code>\$ahora_es = 07</code>
L - bisiesto?	<code><?php \$ahora_es = date(L); ?></code>	<code>\$ahora_es = 0</code> 0 = no; 1 = si
z - día del año	<code><?php \$ahora_es = date(z); ?></code>	<code>\$ahora_es = 81</code>
W - semana del año	<code><?php \$ahora_es = date(W); ?></code>	<code>\$ahora_es = 12</code>



Función date

Trabajando con fecha de Año.

```
<?php
```

```
$ahora=date(Y);  
echo "El año completo es: $ahora";  
echo "<br>";
```

```
$ahora=date(y);  
echo "El año compacto a dos cifra es: $ahora";  
echo "<br>";
```

```
$ahora=date(L);  
echo "Es un año bisiesto? : $ahora";  
echo "<br>";
```

```
$ahora=date(z);  
echo "Nos encontramos en el día del año número: : $ahora";  
echo "<br>";
```

```
$ahora=date(W);  
echo "La semana del año es: $ahora";  
echo "<br>";
```

```
?>
```



Función date

Trabajando con horas.

modificadores	ejemplo	resultado
h - hora, de "01" a "12"	<code><?php \$ahora_es = date(h); ?></code>	<code>\$ahora_es = 10</code>
g - sin ceros, de "1" a "12"	<code><?php \$ahora_es = date(g); ?></code>	<code>\$ahora_es = 10</code>
H - hora, de "00" a "23"	<code><?php \$ahora_es = date(H); ?></code>	<code>\$ahora_es = 10</code>
G - sin ceros, de "0" a "23"	<code><?php \$ahora_es = date(G); ?></code>	<code>\$ahora_es = 10</code>
a - "am" o "pm" A - "AM" o "PM"	<code><?php \$ahora_es = date(a); ?></code>	<code>\$ahora_es = am</code>



Función date

Trabajando con horas.

```
<?php
```

```
$ahora=date(h);
```

```
echo $ahora;
```

```
echo "<br>";
```

```
$ahora=date(g);
```

```
echo $ahora;
```

```
echo "<br>";
```

```
$ahora=date(H);
```

```
echo $ahora;
```

```
echo "<br>";
```

```
$ahora=date(G);
```

```
echo $ahora;
```

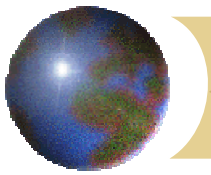
```
echo "<br>";
```

```
$ahora=date(a);
```

```
echo $ahora;
```

```
echo "<br>";
```

```
?>
```



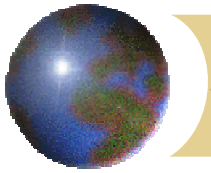
Función date

Jugando con fechas: minutos

modificadores	ejemplo	resultado
i - de "00" a "59"	<code><?php \$ahora_es = date(i); ?></code>	<code>\$ahora_es = 52</code>

Jugando con fechas: segundos

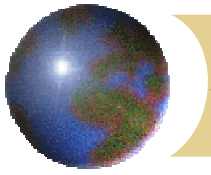
modificadores	ejemplo	resultado
s - de "00" a "59"	<code><?php \$ahora_es = date(s); ?></code>	<code>\$ahora_es = 21</code>



Función date

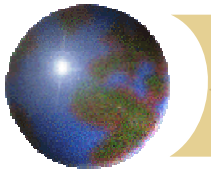
Jugando con fechas: Completas

modificadores	ejemplo	resultado
r - formato RFC 2822	<code><?php \$ahora_es = date(r); ?></code>	<code>\$ahora_es = Fri, 23 Mar 2007 10</code>



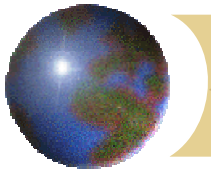
Consistencia de fecha

- ❖ **checkdate(*nMes nDia nAño*)** nos permite chequear si una fecha dada es posible: es decir, que no se trata del día 32 del mes 13.
- ❖ **checkdate** tiene en consideración años bisiestos, y es válido para un rango de años desde 0 a 32767



Y para mas flexibilidad, strftime()

- ❖ la función **date()**, tan flexible como es, presenta el detalle: no admite *localización* a idiomas distintos del inglés.



Y para mas flexibilidad, strftime()

```
<?php
```

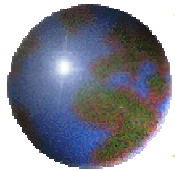
```
    setlocale(LC_ALL, "en");  
    echo strftime("Today is %A, %e %B %Y.");  
    echo "<br>";  
    setlocale(LC_ALL, "sp");  
    echo strftime("Hoy es %A, %d de %B de %Y.");
```

```
?>
```

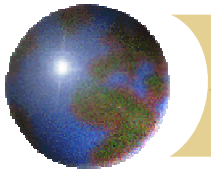


Cuidado !!

- ⊕ No olvides que PHP *se ejecuta en el servidor* y no en tu computador. Por lo tanto, **fecha y hora locales** se refieren *al lugar donde está instalado el servidor*.

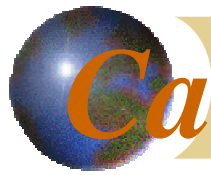


Funciones Numéricas



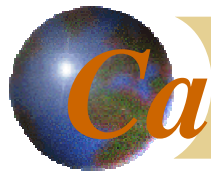
Cambio de base

- ⊕ PHP tiene funciones que permiten obtener **una cadena de caracteres** con la expresión en una **nueva base** de un número expresado en **otra base cualquiera**



Cambio de base

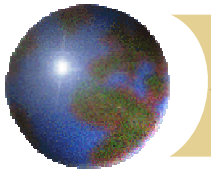
Asignación de valores en distintas bases				
Valor de la variable	Base	Nueva base	Sintaxis	Expresión
\$a=1234	10	8	decoct(\$a)	2322
\$a=1234	10	16	dechex(\$a)	4d2
\$a=1234	10	2	decbin(\$a)	10011010010
\$a=1234	8	10	octdec(\$a)	668
\$a=1234	16	10	hexdec(\$a)	4660
\$a=1010011	2	10	bindec(\$a)	83
\$a=1234	7	14	base_convert(\$a,7,14)	254
\$a=1234	5	18	base_convert(\$a,5,18)	ae
\$a=1234	18	5	base_convert(\$a,18,5)	202123



Cambio de base

```
<?php
$num=1234;
echo decoct ($num) ;
$num=1234;
echo "<br>";
echo base_convert ($num, 7, 14) ;

?>
```



Formato de presentación de números

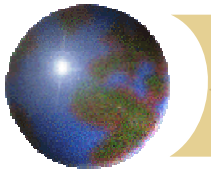
- En PHP es posible establecer el formato de la presentación de los valores numéricos utilizando alguna de estas funciones.

number_format (*número*)

Presenta la parte entera del número (sin decimales) y utiliza como separador de miles una coma (,).

number_format (*número* , *número de cifras decimales*)

Presenta el número de cifras decimales que se indiquen y utiliza como separador decimal un punto (.) y el separador de miles es una coma (,).

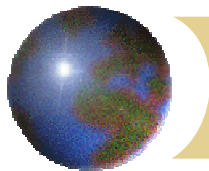


Formato de presentación de números

number_format (número , núm decimales , "sep. decimal" , "sep. miles")

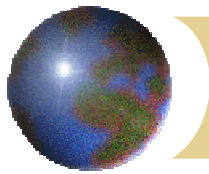
Establece el número de cifras decimales del formato así como el carácter que se establezca como separador de decimales y como separadores de miles.

¡¡Cuidado...!! No te olvides de escribir los caracteres de separación *entre comillas*..



Formato de presentación de números

Formatos de presentación de número					
Valor inicial	Nro de decimales	Sep . dec.	Sep. miles	Sintaxis	Resultado
\$a=1234567.234	0		,	<code>number_format(\$a)</code>	1,234,567
\$a=1234567.234	2	.	,	<code>number_format(\$a,2)</code>	1,234,567.23
\$a=1234567.234	1	,	.	<code>number_format(\$a ,2 , "," , ".")</code>	1.234.567,2
\$a=1234567.234	1	'	esp	<code>number_format(\$a ,2 , " ' " , " ")</code>	1 234 567'2



Formato de presentación de números

```
<?php
```

```
$num=1512.252;
```

```
echo number_format ($num, 2) ;
```

```
echo "<br>" ;
```

```
$num=22.257;
```

```
echo number_format ($num, 2) ;
```

```
echo "<br>" ;
```

```
$num=50.896;
```

```
echo number_format ($num, 2) ;
```

```
echo "<br>" ;
```

```
?>
```