**Funciones**

**date()** - Permite recurperar el año dia, mes, etc del momento en que se hace la consulta pg.97

En el manual php online podemos encontrar clasificadas por tematica todas las funciones existentes **www.php.com**

**Ejemplo Funcion para calcular factorial:**

function factorial($numero)  
{  
$resultado = 1;  
for ($x = $numero; $x>0; $x--) {  
$resultado = $resultado \* $x;  
}

return $resultado;  
}

echo "El factorial de 3 es ".factorial(3)."<br>";  
echo "El factorial de 4 es ".factorial(4)."<br>";  
echo "El factorial de 5 es ".factorial(5)."<br>";

dentro de una funcion podemos definir una variable como global para que pueda ser reconocida fuera de la funcion.

la declaracion static delante de una variable hace que la funcion retenga su valor en memoria en cada ejecucion

**include() y require()**

llamadas a archivos, include es mas tolerante a fallos y permite continuar la ejecucion del programa aunque se haya encontrado un fallo. Require debe utilizarse para aplicaciones que necesiten obligatoriamente de algun archivo critico y no pueda continuarse la ejecucion sin el.

**include\_once () y require\_once()**

hacen lo mismo que include o require solo que ignoran si se intenta cargar el archivo mas de una vez en el script que tenemos (esto es util para proyectos grandes que requieran de multiples cargas)

**Otro ejemplo**

function capitales ($Pais,$Capital = "Madrid", $habitantes = "muchos") {  
 return ("La capital de $Pais es $Capital y tiene $habitantes habitantes <br>");

}

echo capitales("España");  
echo capitales("Portugal","Lisboa");  
echo capitales("Francia","Paris","muchisimos");

La salida es

La capital de España es Madrid y tiene muchos habitantes  
La capital de Portugal es Lisboa y tiene muchos habitantes  
La capital de Francia es Paris y tiene muchisimos habitantes

**func\_num\_args():** Devuelve el numero de argumentos que recibe la funcion desde que es llamada

**func\_get\_arg(5):** Devuelve el quinto argumento pasado

**func\_get\_args():** Devuelve un array con todos los argumentos pasados a la funcion

**Ejemplo**

function concatenar () {  
$resultado = " ";  
$numero\_argumentos = func\_num\_args();  
$array\_parametros = func\_get\_args();  
for ($x = 0; $x <= $numero\_argumentos; $x++);  
$resultado = $resultado.$array\_parametros[$x  
}  
return $resultado."<br>";  
}

echo concatenar ("Hola","Mundo");  
echo concatenar("Esto"," es"," una"," prueba"," de"," la potencia", " de este", " metodo"

**LLamadas por referencia (a funciones y variables ver pags. 112 y 113) - prepend the & operator**

Se puede cambiar arguemntos en la definicion de la funcion

function elevado (&$numbero, &$indice) {  
instrucciones  
}

El operador & hace que las variables puedan ser modificadas dentro de la funcion y guarde este numero fuera de ella

**Funciones variables**

Uno de los trucos de PHP es usar variables para almacenar partes del nombre de una funcion asi

function saludo\_maniana(){  
 return ("Buenos Dias");  
}  
function saludo\_tarde(){  
 return ("Buenas Tardes");  
}  
function saludo\_noche(){  
 return ("Buenas Noches");  
}  
$horario = tarde

$funcion\_variable = "saludo\_".$horario  
echo $funcion\_variable ()

Devuelve Buenas tardes