Tema 1. FONAMENTS DEL LLENGUATGE PHP

La sintaxis de PHP és molt semblant a altres llenguatges de programació molt estesos com ara C, JAVA, Perl o, inclús, el llenguatge de script JavaScript.
PHP ens proporciona un conjunt d'eines compactes pròpies que milloren la interacció entre els usuaris i les pàgines HTML.

1.1. FORMAT DEL CODI PHP

En aquest apartat veurem les bases sintàctiques de la programació de PHP.

1.1.1. Delimitadors

PHP està molt relacionat amb l'HTML, tant és així que el codi PHP apareix normalment inserit dins d'un document HTML. El document PHP, una vegada interpretat correctament en el servidor, genera una pàgina HTML, que és enviada al client.

Per a diferenciar els dos llenguatges dins del mateix document, s'utilitzen etiquetes d'inici i final de codi PHP. Les etiquetes més habituals per a delimitar els blocs de codi PHP són:

```
<?php
        instruccions php
?>
```

Existeixen altres possibles formats d'etiquetes, menys utilitzats que els anteriors, si bé no totes aquestes opcions estan disponibles en el sistema per defecte. La seva utilització serà correcta depenent de les característiques de configuració seleccionades durant el procés d'instal·lació de l'intèrpret de PHP.

```
<?
        instruccions php
?>
```

NOTA: *Ha d'estar activada la directiva short_open_tag al fitxer de configuració php.ini.*

```
<%
        instruccions php
%>
```

NOTA: *Ha d'estar activada la directiva aspt_tags al fitxer de configuració php.ini. Aquest tipus d'etiqueta són les utilitzades per ASP.*

Finalment, es pot introduir el codi PHP dins del document HTML mitjançant l'etiqueta **<script>**:

```
<script language="php">
        instruccions php
</script>
```

Per fer les nostres pràctiques utilitzarem les etiquetes **<?php… …?>**.

1.1.2. Extensió dels fitxers en PHP

La extensió dels fitxers que s'utilitza en PHP es té en compte per part del servidor Web ja que decideix si el document sol·licitat ha de ser processat per l'intèrpret PHP o no en funció de la extensió.

Les extensions que podem trobar són:

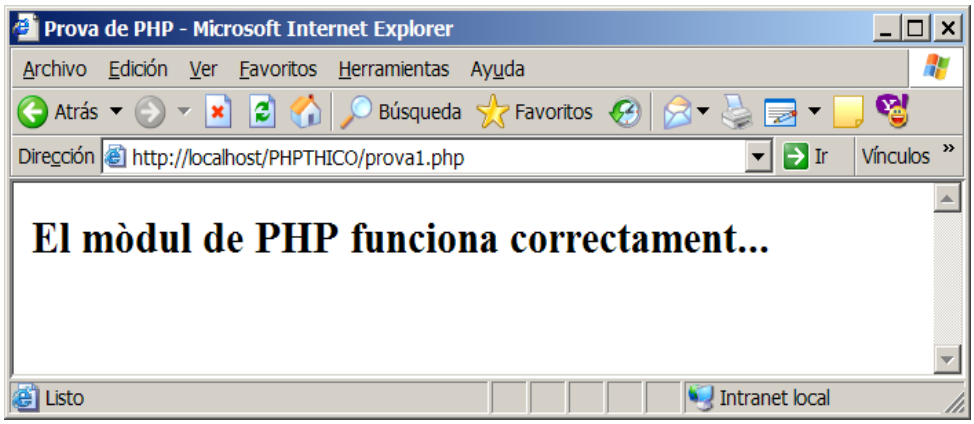
.php3	Codi PHP 3.x
.php4	Codi PHP 4.x
.php	Indica codi PHP
.phps	Utilitzada per veure la sintaxis del codi resaltat en colors
.phtml	Extensió en desús

En general, PHP 3.0 és compatible amb PHP 4.0, per tant no té molta importància posar una extensió o l'altra, ja que l'intèrpret les processa de igual forma.

Per veure el correcte funcionament de l'intèrpret anem a crear la primera pàgina PHP:

```
<html>
<head>
      <title>Prova de PHP</title>
</head>
<body align="center">
<?php
  echo "<h2>El mòdul de PHP funciona correctament...</h2>";
?>
</body>
</html>
```

Una vegada interpretat correctament pel processador PHP el document anterior donaria el següent resultat:



1.1.3. Comentaris

Per incloure comentaris dins del codi, PHP ofereix la possibilitat de fer-ho de 3 formes diferents:

```
<? php
 //Comentari inicial
 echo “Primer tipus de comentari”;    //Comentari de tipus C, C++
 # Comentari final tipus shell de UNIX
?>
```

La doble // o # ens permeten comentar una sola línia de codi.

```
<?php
    /* Comentari inicial */
    echo “Segon tipus de comentari”; /* Comentari de tipus C, C++ */
   /*
      Comentari
      Final
    */
?>
```

Aquest segon tipus de comentari és de tipus multilínia, és a dir, ens permet comentar vàries línies de codi font.

1.1.4. Final de línia

PHP ignora qualsevol caràcter d’espai present en el codi, incloent espais en blanc, tabuladors i salts de línia, excepte si es troben dins d’una cadena de text.

El final de sentència es marca en totes les instruccions amb el **;** o bé aprofitant l’etiqueta de tancament **?>**, ja que es considera que el final de la inclusió de codi limita la introducció de noves instruccions. Per tant, els codis següents són correctes:

```
<?php
  echo “Finalitzat amb l’etiqueta de tancament”
?>
```

```
<?php
  echo “Finalitzat amb punt i coma”;
?>
```

1.2. SINTAXIS BÀSICA

Per començar a programar en PHP és necessari conèixer la seva sintaxi: tipus de variables, definició de constants i tipus i ús d'operadors.

1.2.1. Variables

1.2.1.1. Declaració de variables

En PHP no cal declarar les variables abans de la seva utilització. Les variables es creen en el moment de ser utilitzades per primera vegada. Per a la seva inicialització s'utilitza l'operador d'igualtat (**=**). A partir d'aquest moment es pot recuperar el seu contingut només indicant el nom de la variable.

Per una altra banda, les variables no tenen associat el tipus d'informació que emmagatzemen. De fet, una variable podrà emmagatzemar durant tot el seu temps de vida diferents tipus d'informació.

1.2.1.2. Nombrar les variables

Tots els noms de les variables comencen per **$** seguit, al menys, per una lletra (de la a a la z o de la A a la Z) o d'un guió baix (**_**), per després continuar amb qualsevol combinació de lletres (en majúscules o minúscules), de dígits (del 0 al 9) i de guions baixos.

Podem veure alguns exemples vàlids i invàlids:

Vàlids	Invàlids	Motiu
$Valor_actual	$Valor actual	Conté un espai
$NumeroDeDades	$#Dades	Conté un caràcter no vàlid #
$N	$2Saldos	Comença per un número
$n	$Prova,valor	Conté un caràcter no vàlid ,

L'intèrpret PHP distingeix entre majúscules i minúscules, els noms de variables amb diferències entre majúscules i minúscules en caràcters iguals composen noms de variables diferents; per tant, els identificadors **$n** i **$N** són variables diferents.

1.2.1.3. Variables predefinides

PHP ofereix una gran quantitat de variables predefinides a qualsevol script que s'executi al seus sistema. Per veure aquestes variables es pot utilitzar la funció **phpinfo()**. Aquestes variables guarden informació relativa de l'entorn d'execució de l'intèrpret i del propi PHP.

Aquestes variables es poden classificar en dos grans grups: per una banda estan les que l'entorn li pasa a l'intèrpret de PHP i per una altra, les que l'intèrpret posa a disposició del programador.

Podem veure algunes de les variables d'entorn més utilitzades:

Variable	Significat
SERVER_NAME	Indica el nom de l'equip servidor sobre el que s'executa l'script
SERVER_PORT	Indica el port de l'equip servidor que utilitza el servidor Web per a la comunicació
SERVER_SOFTWARE	Indica quin programari s'està utilitzant a l'equip servidor
REMOTE_PORT	Conté el port que utilitza el peticionari per a comunicar-se amb el servidor Web
REMOTE_ADDR	Conté l'adreça remota des d'on es realitza la petició
DOCUMENT_ROOT	Indica el directori arrel del document sota el qual s'executa l'script
HTTP_REFERER	Conté l'adreça de la pàgina(si existeix) des de la que el navegador ha saltat a la pàgina actual

Aquestes variables s'importen a l'espai de noms globals de PHP des de l'entorn on s'estigui executant l'intèrpret de PHP.

En la següent taula podem veure algunes de les variables que PHP ofereix al programador per facilitar-li la feina:

Variable	Significat
argv	array amb els argument que es passen a l'script
argc	indica el nombre d'arguments que es passen a l'script
PHP_SELF	indica el nom del fitxer que conté l'script d'execució
$_COOKIES	array associatiu amb les variables passades mitjançant cookies
$_GET	array associatiu amb les variables passades mitjançant el mètode GET
$_POST	array associatiu amb les variables passades mitjançant el mètode POST
$_ENV	array associatiu amb les variables passades des de l'entorn
$_SERVER	array associatiu amb les variables passades des del servidor
$_SESSION	array associatiu amb les variables de sessió
$_FILES	array associatiu amb informació relativa als fitxers rebuts

1.2.2. Tipus de dades

PHP suporta 3 tipus de dades simples: **integer**, **float** i **string**; i dos tipus de dades compostes: **array** i **object**. A més fa ús d'un tipus lògic o **boolean** encara que no apareix definit com a tal a la sintaxis del llenguatge.

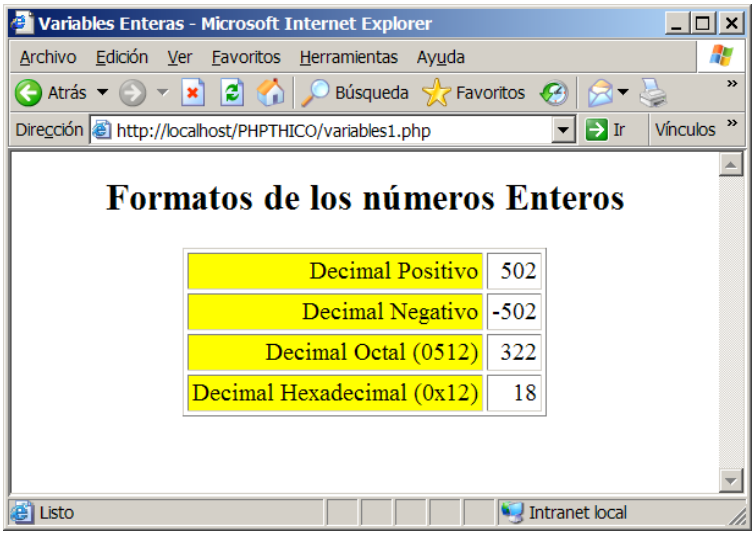
1.2.2.1. Enters

Les variables de tipus **integer** poden contenir números enters que varien entre un rang de -2 bilions i + 2 bilions i es poden especificar utilitzant diferents sintaxis. Els enters es poden representar en format decimal (base 10), octal (base 8) o hexadecimal (base 16).

Format	Valor
Decimal	-33 2139
Octal	071 03664
Hexadecimal	0x7b8 0x395

```
<HTML>
 <HEAD>
  <TITLE>Variables Enteras</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Formatos de los números Enteros</H2>
   <TABLE BORDER="1" CELLPADDING="2" CELLSPACING="2">
     <TR ALIGN="right">
       <TD BGCOLOR="yellow">Decimal Positivo</TD>
       <TD>
        <?php
            $num = 502;   //número decimal
          echo $num;    //mostramos el valor de $num
        ?>
       </TD>
     </TR>
     <TR ALIGN="right">
       <TD BGCOLOR="yellow">Decimal Negativo</TD>
       <TD>
        <?php
            $num = -502;  //número decimal negativo
          echo $num;    //mostramos el valor de $num
        ?>
       </TD>
     </TR>
     <TR ALIGN="right">
       <TD BGCOLOR="yellow">Decimal Octal (0512)</TD>
       <TD>
```

```
        <?php
            $num = 0502;  //número octal
          echo $num;    //mostramos el valor de $num
        ?>
        </TD>
      </TR>
      <TR ALIGN="right">
        <TD BGCOLOR="yellow">Decimal Hexadecimal (0x12)</TD>
        <TD>
        <?php
            $num = 0x12;  //número hexadecimal
          echo $num;    //mostramos el valor de $num
        ?>
        </TD>
      </TR>
    </TABLE>
  </CENTER>
 </BODY>
</HTML>
```



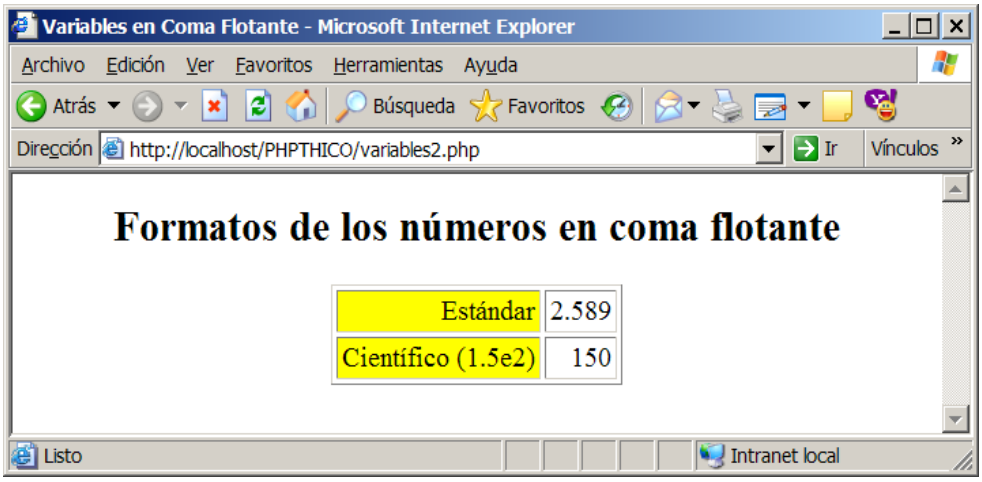
Com es pot veure, la funció echo mostra per defecte sempre la informació en decimal, encara que internament s'hagi emmagatzemat en la variable amb un format diferent.

1.2.2.2. Nombres en coma flotant

Amb les variables de tipus **float** es representen números en coma flotant, o, el que és igual, números amb valors decimals. Són nombres decimals que es poden expressar en forma estàndard o amb notació científica.

Valors
3405.673
-1.958
8.3200e+11
8.3200e11

```
<HTML>
 <HEAD>
  <TITLE>Variables en Coma Flotante</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Formatos de los números en coma flotante</H2>
   <TABLE BORDER="1" CELLPADDING="2" CELLSPACING="2">
     <TR ALIGN="right">
       <TD BGCOLOR="yellow">Estándar</TD>
       <TD>
        <?php
            $num = 2.589; //formato estándar
          echo $num;    //mostramos el valor de $num
        ?>
       </TD>
     </TR>
     <TR ALIGN="right">
       <TD BGCOLOR="yellow">Científico (1.5e2)</TD>
       <TD>
        <?php
            $num = 1.5e2; //formato científico
          echo $num;    //mostramos el valor de $num
        ?>
       </TD>
     </TR>
   </TABLE>
  </CENTER>
 </BODY>
</HTML>
```



1.2.2.3. Cadenes

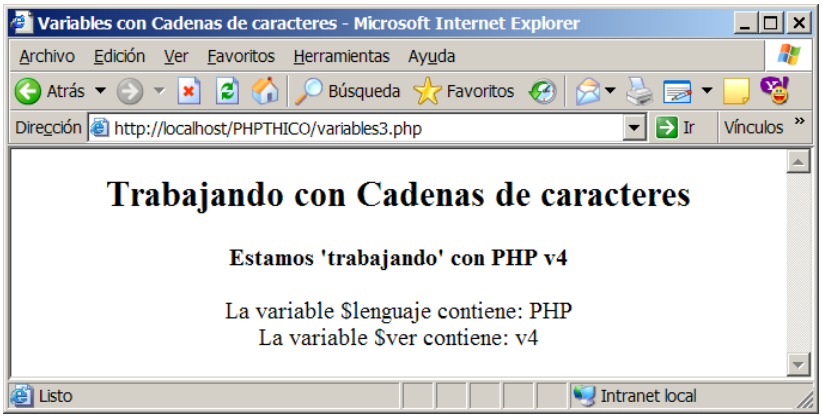
Amb el tipus **string** es representen cadenes de caràcters. Una cadena està formada per zero o més caràcters tancats entre doble cometes “” o entre cometes simples ‘’. Sempre s’ha d’utilitzar el mateix tipus de cometa per completar una cadena.

Cadenes
“hola”
‘hola’
“#12-6”
“Quina passada!”
‘inclou “dobles” cometes’
“inclou \”dobles\” cometes”

Com succeeix al penúltim exemple, en alguns casos es pot barrejar l’ús dels dos tipus d’entrecomillats, principalment per a inserir una cadena literal dins d’una altra. A l’últim exemple s’utilitza el caràcter \ per a poder introduir cometes dobles dins d’un text que està entrecomillat amb cometes dobles.

Quan s’utilitzen cometes dobles, es poden incloure noms de variables que seran avaluades (es substituiran pels seus respectius valors) a l’hora de mostrar la informació. Si posem noms de variables dins d’una cadena amb cometes simples, la variable no serà avaluada.

```
<HTML>
 <HEAD>
  <TITLE>Variables con Cadenas de caracteres</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Trabajando con Cadenas de caracteres</H2>
   <?php
    $lenguaje="PHP";
    $ver="v4";
    echo "<B>Estamos 'trabajando' con $lenguaje $ver </B><BR><BR>";
    echo 'La variable $lenguaje contiene: ';
    echo "$lenguaje <BR>";
    echo 'La variable $ver contiene: ';
    echo $ver;
   ?>
  </CENTER>
 </BODY>
</HTML>
```

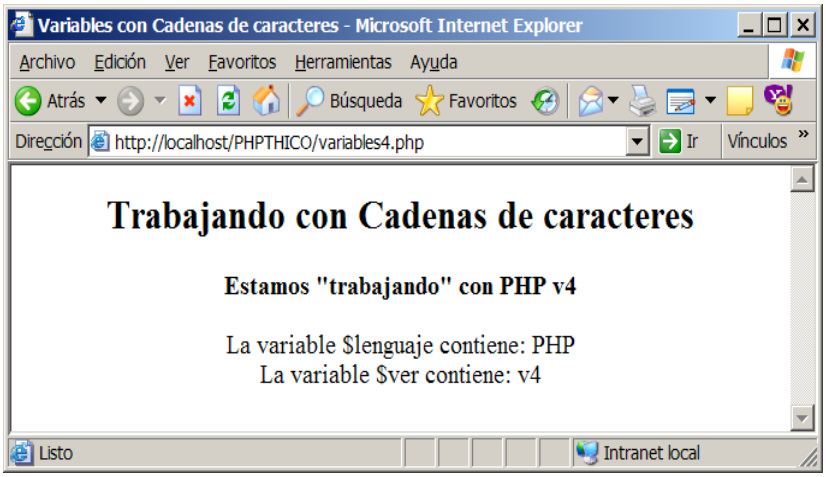


Les cadenes de caràcters, a més de text normal i variables, poden contenir caràcters especials com ara:

Codi d'escapament	Significat
\b	espai cap endarrera (backspace)
\f	canvi de pàgina (form feed)
\n	canvi de línia (line feed)
\r	retorn de carro (carriage return)
\t	tabulació
\\	barra invertida (backslash)
\'	cometa simple
\''	cometa doble
\$	caràcter $

Amb aquests caràcters passa el mateix que amb les variables, si la cadena que conté el caràcter especial utilitza cometes simples, aquests no s'avaluaran i es mostraran directament.

```
<HTML>
 <HEAD>
  <TITLE>Variables con Cadenas de caracteres</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Trabajando con Cadenas de caracteres</H2>
   <?php
    $lenguaje="PHP";
    $ver="v4";
    echo "<B>Estamos \"trabajando\" con $lenguaje $ver </B><BR><BR>";
    echo "La variable \$lenguaje contiene: $lenguaje <BR>";
    echo "La variable \$ver contiene: $ver";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



1.2.2.4. Arrays

Els arrays o matrius són estructures que permeten l'emmagatzematge d'un conjunt de dades sota un mateix nom. És un conjunt ordenat d'elements identificats per un índex. Es poden construir tants índexs com es vulguin, encara que l'ús habitual dels arrays és en forma de matriu unidimensional. La longitud de l'array es modifica de forma dinàmica sempre que s'afegeix un nou element.

En el cas de PHP, els arrays poden estar composats d'elements de diferents tipus i la primera posició és la 0.

Existeixen uns arrays especials de PHP anomenats **associatius** on l'índex és un valor de tipus **string**, de forma que cada posició està definida per una clau i un valor, podent accedir al contingut (valor) mitjançant la clau.

Es poden veure dos estructures d'array com un conjunt d'elements cadascun dels quals té associada una posició o clau:

Cougar	Ford		2500	V6	182
0	1	2	3	4	5

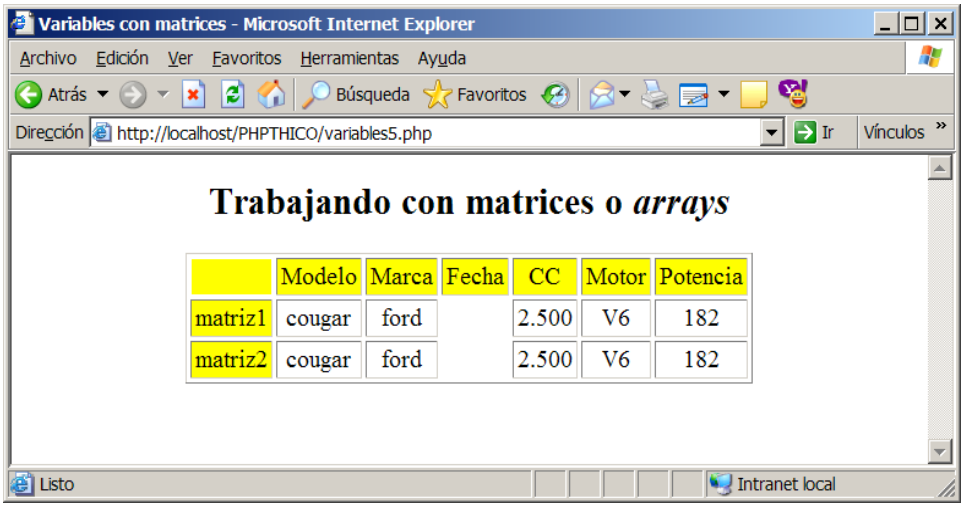
Cougar	Ford		2500	V6	182
model	marca	data	cc	motor	potencia

En els dos arrays els índexs són diferents i poden existir posicions o claus sense contingut associat.

```
<HTML>
 <HEAD>
  <TITLE>Variables con matrices</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Trabajando con matrices o <I>arrays</I></H2>
   <?php
    $matriz1[0]="cougar";
    $matriz1[1]="ford";
    // la tercera posición del array esta vacía
    // por eso le asignamos una cadena sin contenido
    $matriz1[2]="";
    $matriz1[3]="2.500";
    $matriz1[4]="V6";
    $matriz1[5]=182;
    $matriz1[]=182;
    // para añadir el último elemento a una matriz
    // no es necesario poner el número de índice

    // creamos la matriz asociativa
    $matriz2['modelo']="cougar";
    $matriz2['marca']="ford";
    // para marca una posición sin contenido también
```

```
    // se puede utilizar <null>
    $matriz2['fecha']=null;
    $matriz2['cc']="2.500";
    $matriz2['motor']="V6";
    $matriz2['potencia']=182;
  ?>
  <TABLE BORDER="1" CELLPADDING="2" CELLSPACING="2">
    <TR ALIGN="center" BGCOLOR="yellow">
      <TD></TD>
      <TD>Modelo</TD>
      <TD>Marca</TD>
      <TD>Fecha</TD>
      <TD>CC</TD>
      <TD>Motor</TD>
      <TD>Potencia</TD>
    </TR>
    <TR ALIGN="center">
      <TD BGCOLOR="yellow">matriz1</TD>
       <?php
         echo "<TD> $matriz1[0] </TD>";
         echo "<TD> $matriz1[1] </TD>";
         echo "<TD> $matriz1[2] </TD>";
         echo "<TD> $matriz1[3] </TD>";
         echo "<TD> $matriz1[4] </TD>";
         echo "<TD> $matriz1[5] </TD>";
       ?>
    </TR>
    <TR ALIGN="center">
      <TD BGCOLOR="yellow">matriz2</TD>
       <?php
         echo "<TD>". $matriz2['modelo'] ."</TD>";
         echo "<TD>". $matriz2['marca'] ."</TD>";
         echo "<TD>". $matriz2['fecha'] ."</TD>";
         echo "<TD>". $matriz2['cc'] ."</TD>";
         echo "<TD>". $matriz2['motor'] ."</TD>";
         echo "<TD>". $matriz2['potencia'] ."</TD>";
       ?>
    </TR>
  </TABLE>
 </CENTER>
</BODY>
</HTML>
```



1.2.2.5. Objectes

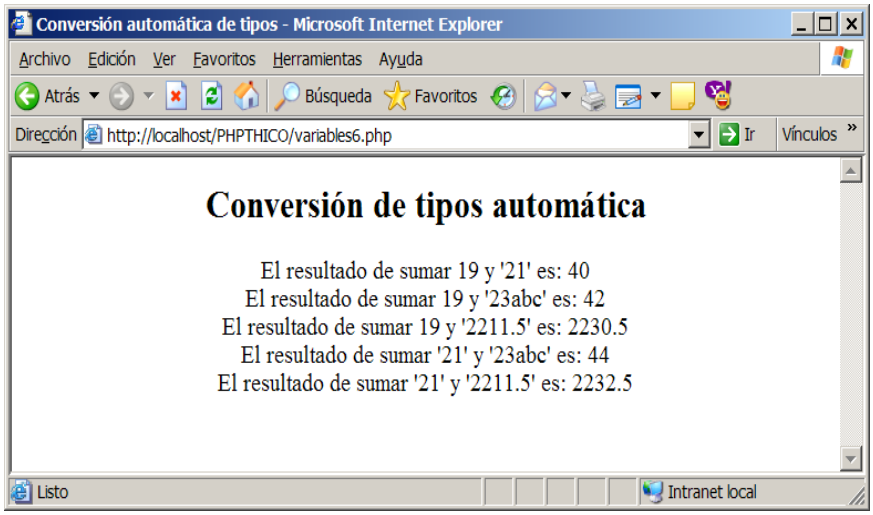
Un objecte és una estructura que manté en sí mateixa tant les seves característiques bàsiques (anomenades propietats), com les possibles funcionalitats (anomenades mètodes).

1.2.2.6. Conversió de tipus

En PHP, el tipus de variable es determina pel context. Així, a una variable que se li assigna un valor de tipus enter serà de tipus **integer**; si seguidament se li assigna una cadena de caràcters, la variable serà de tipus **string**.

Quan s'opera amb variables de diferent tipus, l'intèrpret de PHP tendeix a homogeneitzar els diferents tipus en funció de la operació que es vol realitzar i dels operands que formen part d'ella. Per veure un exemple amb l'operador +:

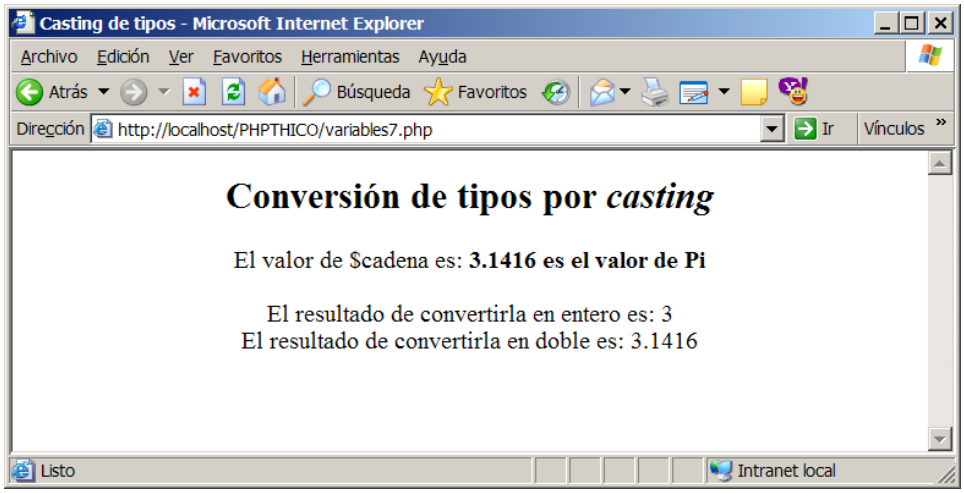
```
<HTML>
 <HEAD>
  <TITLE>Conversión automática de tipos</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Conversión de tipos automática</H2>
   <?php
    $numero=19;
    $cadena1="21";
    $cadena2="23abc";
    $cadena3="2211.5";
    $suma=$numero+$cadena1;
    echo "El resultado de sumar $numero y '$cadena1' es: $suma <BR>";
    $suma=$numero+$cadena2;
    echo "El resultado de sumar $numero y '$cadena2' es: $suma <BR>";
    $suma=$numero+$cadena3;
    echo "El resultado de sumar $numero y '$cadena3' es: $suma <BR>";
    $suma=$cadena1+$cadena2;
    echo "El resultado de sumar '$cadena1' y '$cadena2' es: $suma <BR>";
    $suma=$cadena1+$cadena3;
    echo "El resultado de sumar '$cadena1' y '$cadena3' es: $suma <BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



L'intèrpret intentarà convertir les cadenes en valors numèrics per a que es puguin sumar correctament, si una cadena no es pot convertir a un valor numèric se li assigna el valor 0.

A més de la conversió automàtica realitzada per l'intèrpret de PHP, és possible convertir les variables d'un tipus a un altre quan el programador vulgui. És similar a la conversió de tipus que es fa en el llenguatge C i s'anomena **casting**. Per a això s'escriu entre parèntesis el tipus desitjat (integer, float, string, boolean, array, object) abans de la variable que es vol convertir.

```
<HTML>
 <HEAD>
  <TITLE>Casting de tipos</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Conversión de tipos por <I>casting</I></H2>
   <?php
    $cadena="3.1416 es el valor de Pi";
    echo "El valor de \$cadena es: <B> $cadena </B><BR><BR>";
    $aux=(integer)$cadena;
    echo "El resultado de convertirla en entero es: $aux <BR>";
    $aux=(double)$cadena;
    echo "El resultado de convertirla en doble es: $aux <BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```

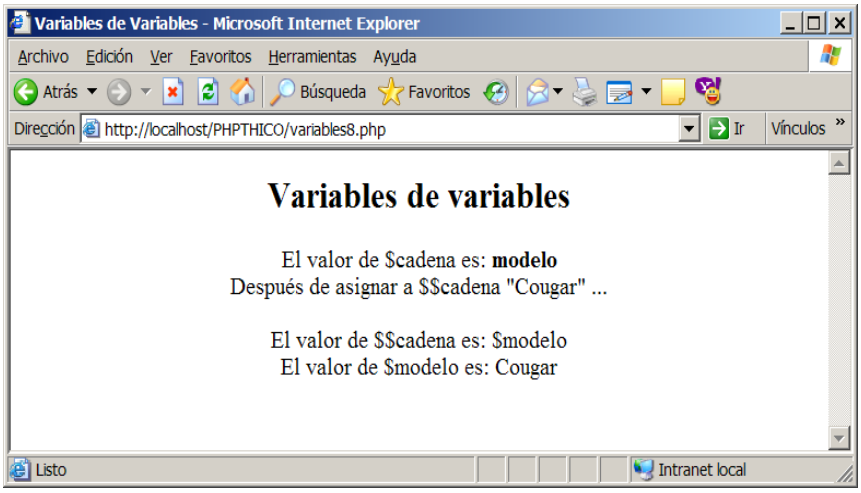


1.2.3. Altres components associats a les variables

1.2.3.1. Variables de variables

Consisteix en reutilitzar el valor de les variables com a nom d'altres variables al mateix temps. Les variables de variables es poden establir i utilitzar dinàmicament, de fet, habitualment s'utilitzen en scripts on es genera codi de forma dinàmica.

```
<HTML>
 <HEAD>
  <TITLE>Variables de Variables</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Variables de variables</H2>
   <?php
    $cadena="modelo";
    echo "El valor de \$cadena  es: <B> $cadena </B><BR>";
    echo 'Después de asignar a $$cadena "Cougar" ...<BR><BR>';
    $$cadena="Cougar";
    echo "El valor de \$\$cadena es: $$cadena <BR>";
    echo "El valor de \$modelo  es: $modelo <BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



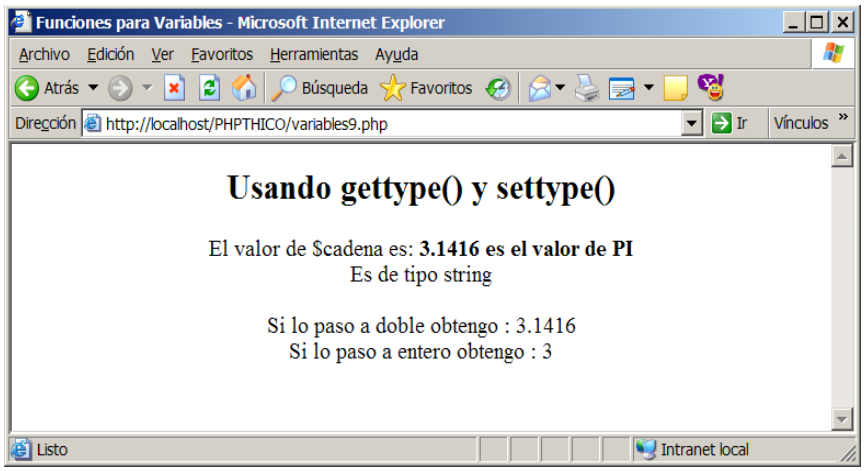
1.2.3.2. Funcions per a variables

PHP proporciona un conjunt de funcions de gran utilitat a l'hora de tractar les variables:

- **gettype(**variable**)**: retorna el tipus de dada passat com a paràmetre, pot retornar: *integer, float, string, array, class, object* i *unknown type*.

- **settype(**variable, tipus**)**: posa el tipus de dada a guardar en una variable. Té dos arguments: el nom de la variable i el tipus que es vol establir. Amb aquesta funció es poden realitzar conversions d'un tipus de dada a un altre. Retornarà *true* si ha tingut èxit, en cas contrari retornarà *false*.

```
<HTML>
 <HEAD>
  <TITLE>Funciones para Variables</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Usando gettype() y settype() </H2>
   <?php
    $cadena="3.1416 es el valor de PI";
    echo "El valor de \$cadena  es: <B> $cadena </B><BR>";
    echo "Es de tipo ".gettype($cadena)."<BR><BR>";
    settype($cadena,"double");
    echo "Si lo paso a doble obtengo : $cadena <BR>";
    settype($cadena,"integer");
    echo "Si lo paso a entero obtengo : $cadena <BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```

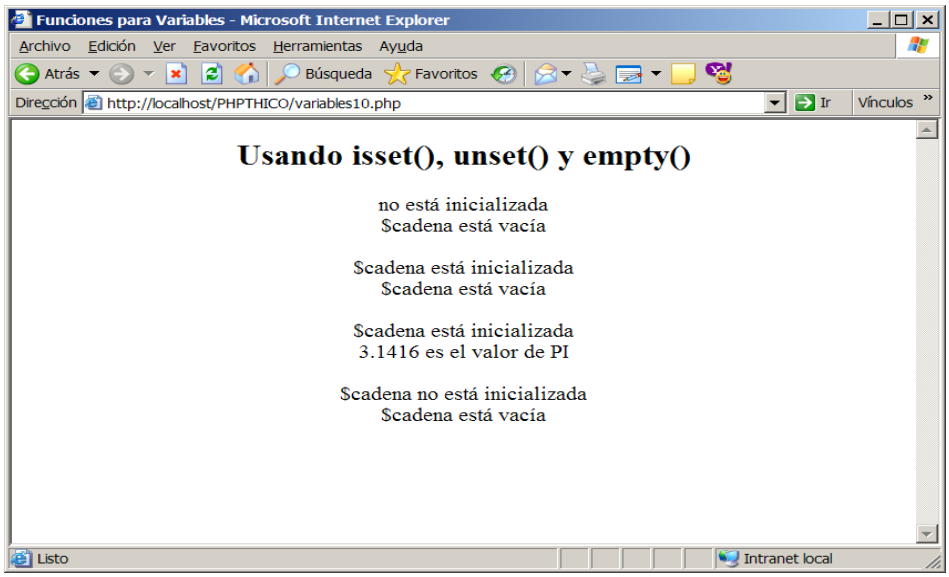


- **isset(**variable**)**: indica si una variable ha sigut inicialitzada amb un valor, en aquest cas retorna *true*, en cas contrari retorna *false*.

- **unset(**variable**)**: destrueix una variable alliberant els seus recursos. S'indica com a paràmetre la variable a destruir.

- **empty(**variable**)**: retorna *true* si la variable encara no ha estat inicialitzada, o bé té un valor igual a 0 o és una cadena buida, en cas contrari retorna *false*.

```
<HTML>
 <HEAD>
  <TITLE>Funciones para Variables</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Usando isset(), unset() y empty()</H2>
   <?php
    $cadena; //definimos la variable pero no la inicializamos.
    echo '$cadena ';
    echo (isset($cadena))?'está ':'no está ';
    echo "inicializada<BR>";
    echo (empty($cadena))?'$cadena está vacía':$cadena;
    echo "<BR><BR>";
    $cadena="";
    echo '$cadena ';
    echo (isset($cadena))?'está ':'no está ';
    echo "inicializada<BR>";
    echo (empty($cadena))?'$cadena está vacía':$cadena;
    echo "<BR><BR>";
    $cadena="3.1416 es el valor de PI";
    echo '$cadena ';
    echo (isset($cadena))?'está ':'no está ';
    echo "inicializada<BR>";
    echo (empty($cadena))?'$cadena está vacía':$cadena;
    echo "<BR><BR>";
    unset($cadena);
    echo '$cadena ';
    echo (isset($cadena))?'está ':'no está ';
    echo "inicializada<BR>";
    echo (empty($cadena))?'$cadena está vacía':$cadena;
   ?>
  </CENTER>
 </BODY>
</HTML>
```



- **is_int(**variable**)**, **is_integer(**variable**)**, **is_long(**variable**)**: retornen *true* si la variable passada és de tipus **integer**; en cas contrari retorna *false*.

- **is_float(**variable**)**, **is_real(**variable**)**, **is_double(**variable**)**: retornen *true* si la variable passada és de tipus **float**; en cas contrari retorna *false*.

- **is_numeric(**variable**)**: retorna *true* si la variable passada és un número o una cadena numèrica; en cas contrari retorna *false*.

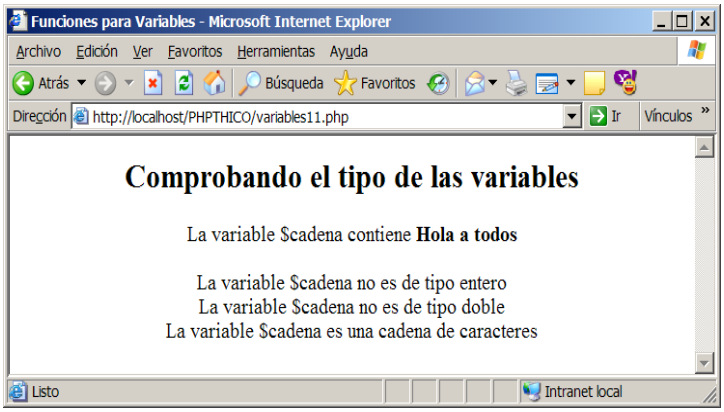
- **is_bool(**variable**)**: retorna *true* si la variable passada és de tipus lògic; en cas contrari retorna *false*.

- **is_array(**variable**)**: retorna *true* si la variable passada és de tipus *array*; en cas contrari retorna *false*.

- **is_string(**variable**)**: retorna *true* si la variable passada és de tipus *string*; en cas contrari retorna *false*.

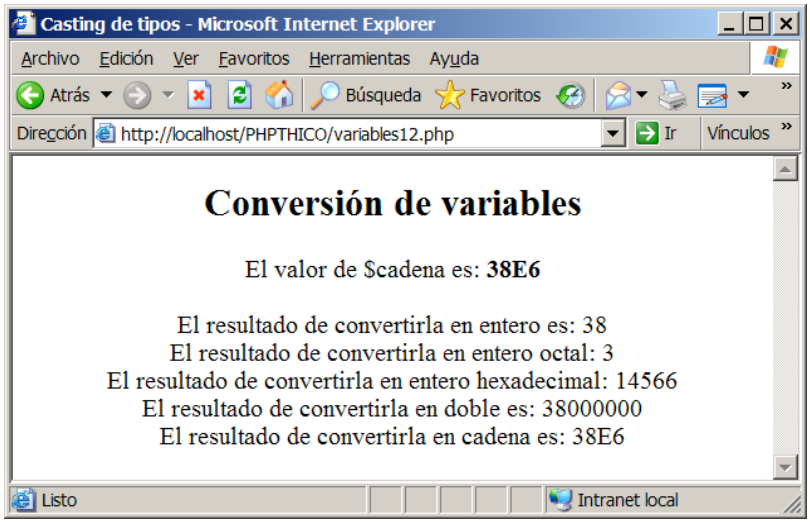
- **is_object(**variable**)**: retorna *true* si la variable passada és de tipus *object*; en cas contrari retorna *false*.

```
<HTML>
 <HEAD>
  <TITLE>Funciones para Variables</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Comprobando el tipo de las variables</H2>
   <?php
    $cadena="Hola a todos";
    echo 'La variable $cadena contiene <B>'.$cadena.'</B><BR><BR>';
    echo 'La variable $cadena ';
    echo (is_integer($cadena))?'es':' no es';
    echo ' de tipo entero <BR>';
    echo 'La variable $cadena ';
    echo (is_double($cadena))?'es':' no es';
    echo ' de tipo doble <BR>';
    echo 'La variable $cadena ';
    echo (is_string($cadena))?'es':' no es';
    echo ' una cadena de caracteres <BR>';
   ?>
  </CENTER>
 </BODY>
</HTML>
```



- **intval(**variable,base**)**, **floatval(**variable**)**, **strval(**variable**)**: aquestes funcions serveixen per fer conversions de tipus (casting), de forma que converteixen a **integer**, **float** o **string** respectivament, el valor de la variable passada com a paràmetre.

```
<HTML>
 <HEAD>
  <TITLE>Casting de tipos</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Conversión de variables</H2>
   <?php
    $cadena="38E6";
    echo "El valor de \$cadena es: <B> $cadena </B><BR><BR>";
    $aux=intval($cadena);
    echo "El resultado de convertirla en entero es: $aux <BR>";
    $aux=intval($cadena,8);
    echo "El resultado de convertirla en entero octal: $aux <BR>";
    $aux=intval($cadena,16);
    echo "El resultado de convertirla en entero hexadecimal: $aux <BR>";
    $aux=doubleval($cadena);
    echo "El resultado de convertirla en doble es: $aux <BR>";
    $aux=strval($cadena);
    echo "El resultado de convertirla en cadena es: $aux <BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



1.2.4. Constants

Una constant és una variable que manté el mateix valor durant tota l'execució del programa. Es pot assegurar que la constant manté sempre un mateix valor; en cap part de l'script es pot canviar el seu valor una vegada s'ha definit.

1.2.4.1. Funcions per a constants

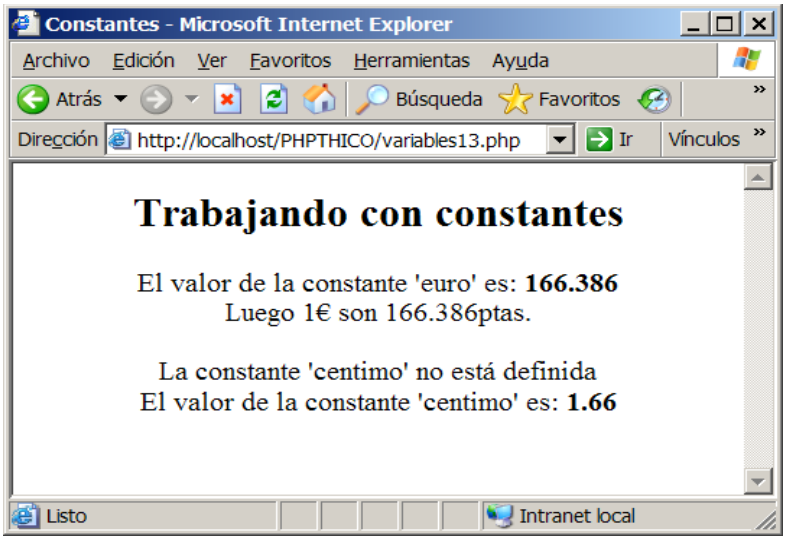
Per a treballar amb constants, PHP proporciona les següents funcions:

- **define(**constant, valor**)**: permet crear una constant assignant-li el seu nom i el seu valor mitjançant el paràmetre que rep.

- **defined(**constant**)**: retorna *true* si la constant passada com a argument està definida i, per tant, existeix; en cas contrari retorna *false*.

Els noms de constant no porten a davant el **$**, ja que no es tracta d'una variable.

```
<HTML>
 <HEAD>
  <TITLE>Constantes</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Trabajando con constantes</H2>
   <?php
    define("euro",166.386);
    echo "El valor de la constante 'euro' es: <B>". euro ."</B><BR>";
    echo "Luego 1€ son ".euro."ptas.<BR><BR>";
    echo "La constante 'centimo' ";
    echo (defined("centimo"))?"está".centimo:"no está";
    echo " definida<BR>";
    define("centimo",1.66);
    echo "El valor de la constante 'centimo' es: <B>". centimo ."</B><BR><BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```

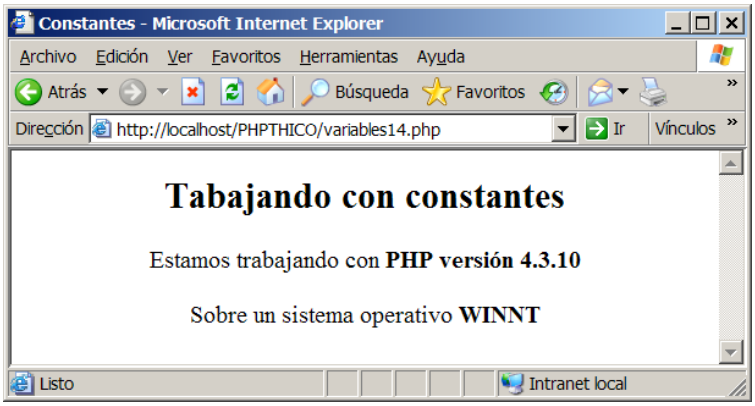


1.2.4.2. Constants predefinides

Existeixen un conjunt de constants predefinides i per tant no es poden definir constants amb el mateix nom d'aquestes constants. Algunes són:

Constants	Significat
PHP_VERSION	Cadena que representa la versió de l'intèrpret de PHP en ús.
PHP_OS	Cadena amb el nom del sistema operatiu on s'està executant l'intèrpret de PHP.
TRUE	Cert.
FALSE	Fals.
E_ERROR	Informació sobre errors diferents als d'interpretació dels quals no es possible recuperar-se.
E_PARSE	Informa que l'intèrpret ha trobat una sintaxis invàlida a l'arxiu de comandes. Finalitza l'execució.
E_NOTICE	Informa que s'ha produït alguna cosa incorrecta que pot provenir o no d'un error. L'execució continua.
E_WARNING	Indica un error que no impedeix que continuï l'execució.
E_ALL	Conjunt amb tots els errors que s'han produït.

```
<HTML>
 <HEAD>
  <TITLE>Constantes</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Tabajando con constantes</H2>
   <?php
    echo "Estamos trabajando con <B>PHP versión ". PHP_VERSION ."</B><BR><BR>";
    echo "Sobre un sistema operativo <B>". PHP_OS ."</B>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



1.2.5. Expressions

Les expressions són la base principal de PHP, que és, en sí, un llenguatge orientat a expressions, ja que casi tot ell, és una expressió.

Una expressió pot ser quelcom tan simple com un número o una variable, o pot incloure moltes variables, constants, operadors i funcions conjuntament.
Les expressions més bàsiques són les variables i les constants. Un altre tipus d'expressió el formen les expressions que s'avaluen a 0 o 1 corresponent-se amb el valor *false* o *true*.

1.2.6. Operadors

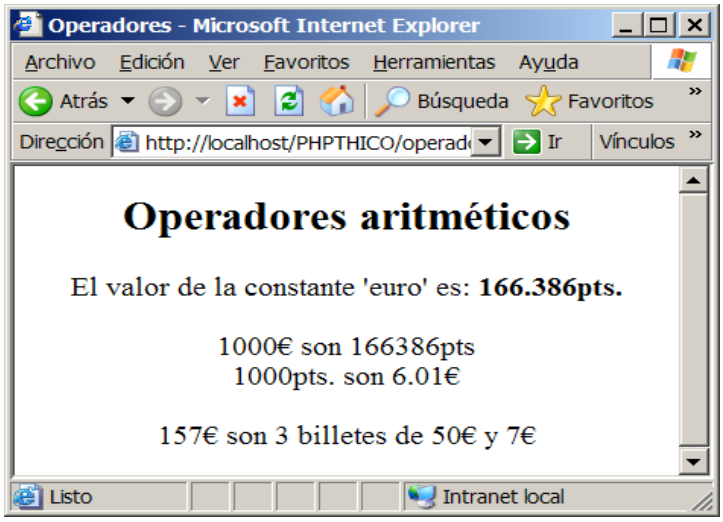
Els operadors s'utilitzen per a determinar un valor, o bé, per a obtenir un valor final a partir d'un o més operands.

1.2.6.1. Operadors aritmètics

Aquests operadors funcionen igual que a l'aritmètica bàsica i es poden aplicar a variables i a constants numèriques:

Operador	Exemple	Descripció
+	$a + $b	Suma dos operands
-	$a - $b	Resta dos operands
*	$a * $b	Multiplica dos operands
/	$a / $b	Divideix dos operands
%	$a % $b	Resto de la divisió entera

```
<HTML>
 <HEAD>
  <TITLE>Operadores</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Operadores aritméticos</H2>
   <?php
    define("euro",166.386);
    echo "El valor de la constante 'euro' es: <B>". euro ."pts.</B><BR><BR>";
    echo "1000€ son ". (euro*1000) ."pts<BR>";
    echo "1000pts. son ". intval((1000/euro)*100)/100 ."€<BR><BR>";
    echo "157€ son ". intval(157/50) ." billetes de 50€";
    echo " y ". (157%50) ."€";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



1.2.6.2. Operadors d'assignació

L'operador d'assignació més utilitzat és **=**; la seva funció bàsica és assignar un valor a una variable. L'operador d'assignació és un operador binari que sempre pren la forma:

variable = expressió

Operador	Exemple	Descripció
=	$a = $b	$a pren el valor de $b
+=	$a += $b	Equival a $a = $a + $b
-=	$a -= $b	Equival a $a = $a - $b
*=	$a *= $b	Equival a $a = $a * $b
/=	$a /= $b	Equival a $a = $a / $b
%=	$a %= $b	Equival a $a = $a % $b
.=	$a .= $b	Equival a $a = $a . $b

```
<HTML>
 <HEAD>
  <TITLE>Operadores</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Operadores de asignación</H2>
   <?php
    define("euro",166.386);
    // en la variable $texto vamos a concatenar todo el texto a mostrar
    // por pantalla
    $texto = "El valor de la constante 'euro' es: <B>";
    $texto .= euro;
    $texto .= "pts.</B><BR><BR>";
    $texto .= "1000€ son ";
    $valor = euro*1000;       //calculamos el valor de 1000€
    $texto .= $valor;
    $texto .= "pts<BR>";
    $texto .= "1000pts. son ";
    $valor =1000/euro;       //obtenemos el valor de 1000pts en €
    $valor *= 100;            //lo multiplicamos por 100
    $valor = intval($valor); //eliminamos los decimales que no queremos
    $valor /= 100;           //lo dividimos por 100 para obtener el valor final
    $texto .= "€<BR><BR>";
    $texto .= "157€ son ";
    $valor = intval(157/50);
    $texto .= $valor ." billetes de 50€ y ". (157%50) ."€";
    echo $texto;
   ?>
  </CENTER>
 </BODY>
</HTML>
```

1.2.6.3. Operadors de cadenes

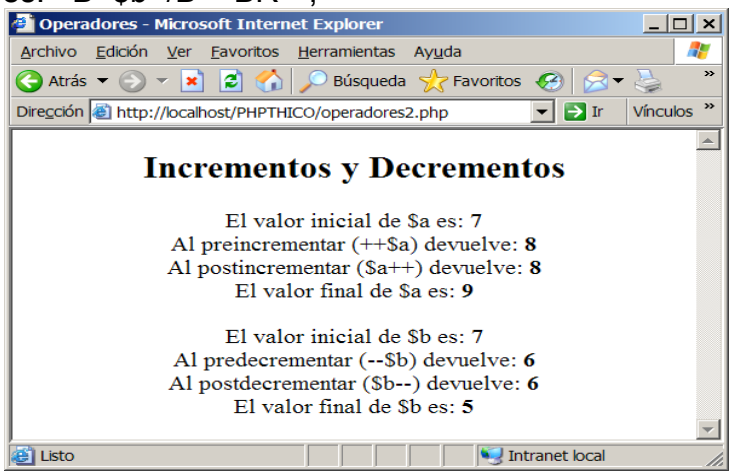
L'operador de concatenació de cadenes es fa mitjançant el . i també es pot utilitzar l'operador de concatenació i assignació vist a l'apartat anterior (.=).

1.2.6.4. Operadors d'increment i de decrement

PHP suporta operadors específics per a incrementar i decrementar el valor de les variables. La possibilitat d'aquests operadors és:

Operador	Exemple	Descripció
++	++$a	**Preincrement:** incrementa $a en 1 i després el retorna
	$a++	**Postincrement:** retorna $a i després l'incrementa en 1
- -	--$a	**Predecrement:** decrementa $a en 1 i després el retorna
	$a--	**Postdecrement:** retorna $a i després el decrementa en 1

```
<HTML>
 <HEAD>
  <TITLE>Operadores</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Incrementos y Decrementos</H2>
   <?php
    $a=7;
    echo "El valor inicial de \$a es: <B>$a</B><BR>";
    echo "  Al preincrementar (++\$a) devuelve: <B>".++$a."</B><BR>";
    echo "  Al postincrementar (\$a++) devuelve: <B>".$a++."</B><BR>";
    echo "El valor final de \$a es: <B>$a</B><BR><BR>";
    $b=7;
    echo "El valor inicial de \$b es: <B>$b</B><BR>";
    echo "  Al predecrementar (--\$b) devuelve: <B>".--$b."</B><BR>";
    echo "  Al postdecrementar (\$b--) devuelve: <B>".$b--."</B><BR>";
    echo "El valor final de \$b es: <B>$b</B><BR>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```



S'ha de destacar com els operadors de potsincrement i postdecrement retornen el mateix valor que reben i posteriorment el modifiquen.

1.2.6.5. Operadors de comparació

Els operadors de comparació s'utilitzen bàsicament per a comparar expressions. Les expressions que utilitzen operadors de comparació habitualment realitzen preguntes sobre els dos valors continguts als operands. La resposta a aquesta pregunta pot ser *true* o *false*.

Oper.	Exemple	Retorna true quan
==	$a == $b	Els operands són iguals
!=	$a != $b	Els operands són diferents
===	$a === $b	Els operands són idèntics: iguals i del mateix tipus
!==	$a !== $b	Els operands no són iguals o del mateix tipus
<	$a < $b	L'operand de l'esquerra és menor que el de la dreta
>	$a > $b	L'operand de l'esquerra és més gran que el de la dreta
<=	$a <= $b	L'operand de l'esquerra és menor o igual al de la dreta
>=	$a >= $b	L'operand de l'esquerra és més gran o igual al de la dreta

1.2.6.6. Operadors lògics

Els operadors lògics (o booleans) s'utilitzen conjuntament amb expressions que retornen valors lògics. La seva sintaxis és:

Oper.	Exemple	Retorna true quan
&&	$a && $b	$a i $b són tots dos *true*
and	$a and $b	
||	$a || $b	$a o $b són *true*
or	$a or $b	
!	! $a	$a és *false*, nega el valor lògic de la variable
xor	$a xor $b	$a és *true* o $b és *true* però no ho són els dos alhora

```
<HTML>
 <HEAD>
  <TITLE>Operadores</TITLE>
 </HEAD>
 <BODY>
  <CENTER>
   <H2>Operadores lógicos</H2>
   <?php
     $a=3;
       $b=7;
       $c=9;
     echo "<BR>Los tres números a comparar son: ";
       echo "<B>$a, $b </B>y<B> $c</B><BR><BR>";
       echo " y el mayor es el <B>";
       echo ($a>$b)&&($a>$c)?$a:"";
       echo ($b>$a)&&($b>$c)?$b:"";
       echo ($c>$a)&&($c>$b)?$c:"";
       echo "</B>";
   ?>
  </CENTER>
 </BODY>
</HTML>
```