



Laboratorio II / Laboratorio de Software II 2024

Clase Teórica 01: Intro PHP, require Variables, Constantes, Operadores



Docente: Myriam Ruiz
Licenciada en Informática
Profesora en Computación y Matemática
Programadora Universitaria



Requisitos para Cursar Laboratorio II

- Para Cursar se requiere tener **Regular o Aprobada:**

Programador Universitario	Lic. en Informática	Ing. en Informática
1) Laboratorio I	1) Laboratorio I	1) Laboratorio de Software I 2) Elementos de Comp. y Lógica

❑ No se aceptan condicionales ni vocacionales.

CÁTEDRA LABORATORIO II:



Myriam Ruiz

Prof. Adjunta
(teoría)

Juan Pablo Pepe

Jefe de Trabajos Prácticos



César Flores

Auxiliar Docente Graduado

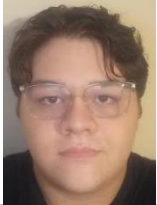
Daniel Singh

Auxiliar Docente Graduado



Martín Brandalisi

Ayudante Estudiantil



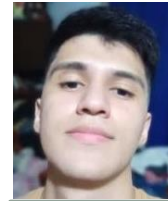
Bruno Ojeda

Ayudante Estudiantil



Tomás Rivera

Ayudante Estudiantil



Camila Sánchez

Ayudante Estudiantil

CÁTEDRA:
LABORATORIO II:

Auxiliares



Diego
Aredes



Santiago
Barrionuevo



Fernando
Flores



Gonzalo
Ibrahim



Joaquín
Lechesí



Matías
Medina



Matías
Prado



Iván
Risso



Franco
Rodríguez



Gabriel
Rojas



Sofía
Sandoval

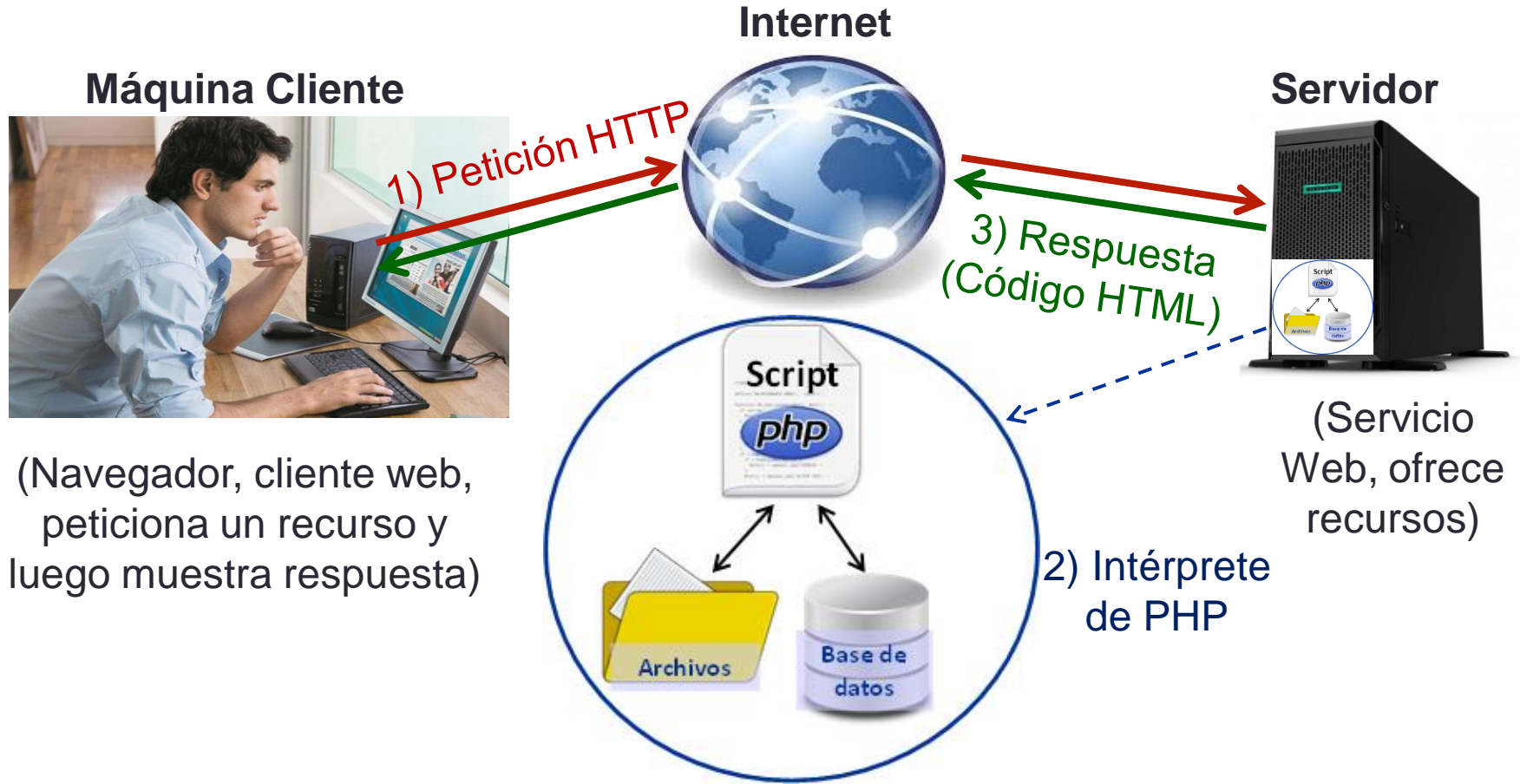


Nicolás
Velasco



Santiago
Zelaya

Proceso para visualizar una web



Qué es PHP?

- Surgió en 1994 como ***P**ersonal **H**ome **P**age (PHP) Tools*, con el propósito de facilitar el diseño de páginas web de carácter dinámico.



Rasmus Lerdorf

- Actualmente el nombre PHP es un acrónimo recursivo en inglés de ***P**HP **H**ypertext **P**reprocessor* (Preprocesador de Hipertexto PHP).

Características de PHP

- Características:

- ✓ Lenguaje del lado del Servidor (El código **se procesa en el Servidor** y no en la máquina cliente, no sucede en donde está el navegador del visitante del sitio)
- ✓ Es Multiplataforma (Corre en diversos sistemas Operativos: Windows, Linux, macOS, etc.)
- ✓ Es Software de código abierto y distribución libre
- ✓ Es Interpretado (no se compila, no se genera un código ejecutable)

Características de PHP

- Características:

- ✓ Soporta un amplio abanico de bases de datos (PostgreSQL, SQLite, MariaDB, MySQL, MongoDB, etc.).



Características de PHP

- Características:

- ✓ Dispone de abundante documentación y soporte.

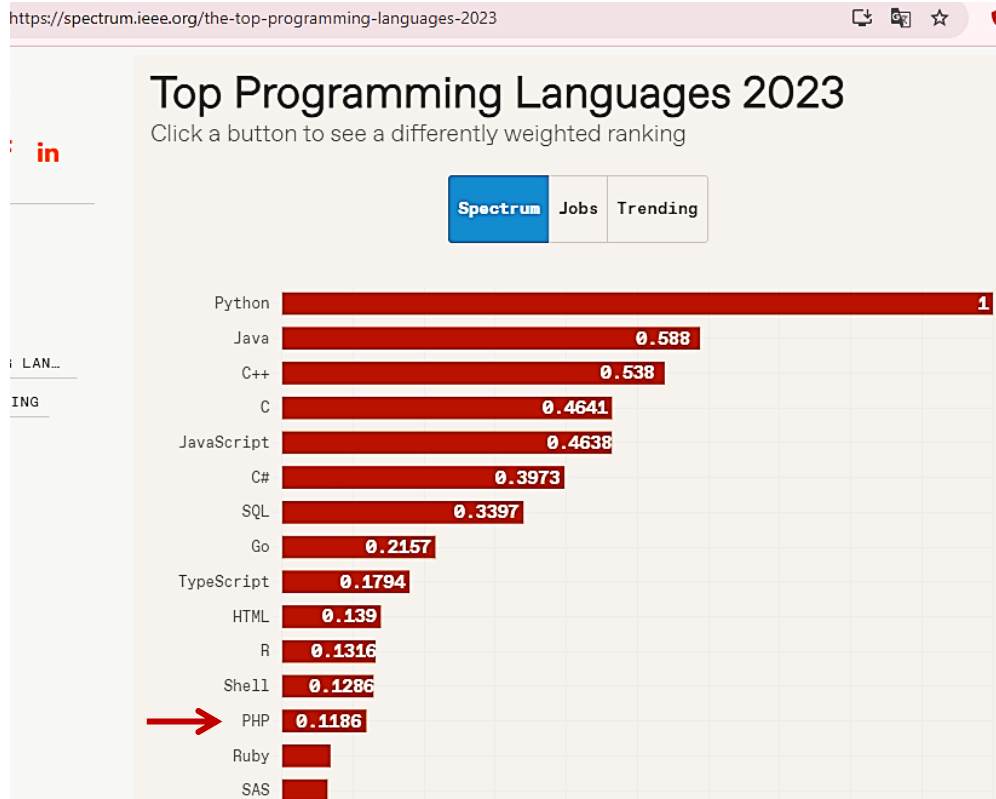
Sitio oficial: <https://www.php.net/manual/es/>

- ✓ Sencillo de instalar

- ✓ Curva de aprendizaje baja (se demora poco en aprender a usarlo)

Que tanto se usa PHP para Web a nivel mundial?

- Según un informe del 2023 basado en Mediciones de 9 fuentes distintas:



Fuente: <https://spectrum.ieee.org/the-top-programming-languages-2023>

Qué permite hacer PHP?

- Permite:
 - ✓ Generar HTML (que es lo que el navegador interpreta)
 - ✓ Creación de imágenes, PDF y cualquier tipo de texto.
 - ✓ Interactuar con Base de datos para tener un sitio dinámico (Similar a lo visto en Wordpress)
 - ✓ Subir Archivos al servidor
 - ✓ Procesar datos provenientes de formularios
 - ✓ Enviar mails
 - ✓ Etc.

Qué necesitamos para trabajar con PHP?

- Un Servidor Web
- Un Gestor de Base de Datos
- Un Intérprete de PHP
- Un IDE * o un editor de texto: Visual Studio Code, Sublime text, Netbeans, Atom, etc.

* Integrated Development Environment (entorno de desarrollo integrado)

Lo que usaremos en Laboratorio II

- Un Servidor Web, un Gestor de base de Datos y un Intérprete de PHP 8.x: **XAMPP** (Apache y MariaDB)

<https://www.apachefriends.org/es/index.html>

(Para ver como instalar, ver videos en Plataforma)

- Un IDE: **Visual Studio Code**

<https://code.visualstudio.com/>

Cómo usar PHP? – Etiquetas delimitadoras

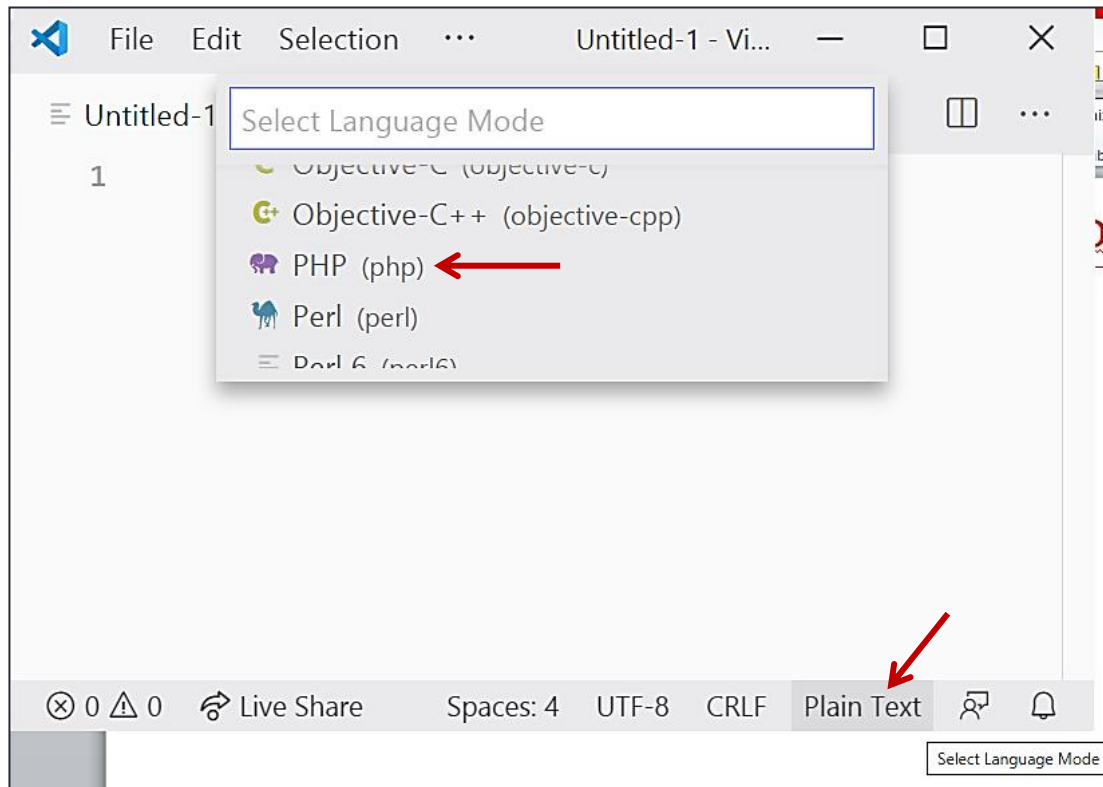
- Todo el código PHP debe estar entre las etiquetas:

<?php

?>

- La extensión de los archivos que contienen php deberá ser **.php** para poder ser interpretado por el servidor, aún si tiene también código html

En el IDE Visual Studio Code, elegir tipo php



Cómo usar PHP? – Sangría

- Para una lectura más clara, se recomienda indentar el código (dejar sangría) usando 4 espacios en blanco (no usar Tab, porque cada IDE lo interpreta diferente)

```
<?php
|   echo '<p>Hola mundo</p>';
?>
```



Cómo usar PHP? –Cuál es el objetivo?

- Al final de cuentas, lo que haremos con php será generar un código html que el navegador pueda interpretar.
 - Existen 2 formas de trabajar para lograrlo:
 1. Incorporar código php dentro de la estructura html.
 2. En un archivo php, incorporar el contenido de archivos html o php con etiquetas html (**recomendado**).
- * En ambos casos el archivo tendrá extensión .php

Cómo usar PHP? – Cómo incorporar html

- Las partes “fijas” de un sitio, las pondremos en archivos separados, y las incorporaremos cuando las necesitemos, mediante la instrucción:
- `require_once(ubicación/archivo.extensión);`
- Permite controlar que un archivo se incorpore de manera obligatoria, una única vez.
- Se pueden incorporar tanto archivos .html como .php

Cómo usar PHP? – Cómo incorporar html

- Si quisiéramos que un archivo se pueda incorporar varias veces en un mismo .php, se puede usar:
- `require(ubicación/archivo.extensión);`
- **Importante:** Todo código php que muestre algo en el navegador, **debe generar la estructura html completa**, vista en Laboratorio I, mediante los require.

Cómo usar PHP? – Cómo incorporar html o php

Para entenderlo, deben pensar que cada require produce que el código del archivo externo se copie en el archivo donde están los require

Cómo separar el código html?

- Dejamos en archivos separados, dentro de la carpeta html, aquellas partes del html que son iguales en todas las páginas. Cuáles serían?

Zona de menú
Zona de Encabezado
Contenido Principal
Pie

Cómo separar el código html?

- Ejemplo de html fragmentado (encabezado.html):

D: > xampp21 > htdocs > sitio1 > html > <> encabezado.html >  html >  body

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Sitio de Prueba 1</title>
7      <link rel="stylesheet" href="css/estilo.css">
8  </head>
9  <body>
10
```

Cómo separar el código html?

- Ejemplo de html fragmentado (menu.html):

```
D: > xampp21 > htdocs > sitio1 > html > <> menu.html > section.contenedor > mai
1  <section id="contenedor-menu">
2
3      <a href="#contenedor-menu" id="abrir"> ▾ </a>
4      <a href="#" id="cerrar"> ▴ </a>
5      <nav >
6          <ul>
7              <li><a href="html/pagina1.html">Enlace 1</a></li>
8              <li><a href="html/pagina2.html">Enlace 2</a></li>
9          </ul>
10     </nav>
11 </section>
12 <header> <h1>Título Principal</h1></header>
13 <section class="contenedor">
14     <main>
15         <section>
16             <article class="art1">
17
```

Unión de todas las partes en el .php

D: > xampp21 > htdocs > sitio1 > 🐘 index.php

```
1  <?php
2      require_once('html/encabezado.html');
3      require_once('html/menu.html');
4
5      echo '<h2>Esto está escrito en el archivo php</h2>';
6
7      require_once('html/contenido.html');
8      require_once('html/pie.html');
9  ?>
10
```


Dónde ubicar y cómo visualizar los archivos php?

- Guardar todos los archivos en la carpeta:

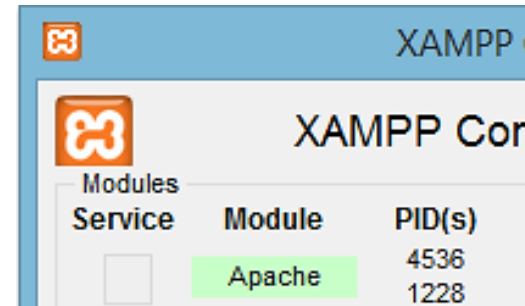
`c:\xampp\htdocs`

Ejemplo: `c:\xampp\htdocs\ejemplos-clase01\index.php`

- En el navegador escribir:

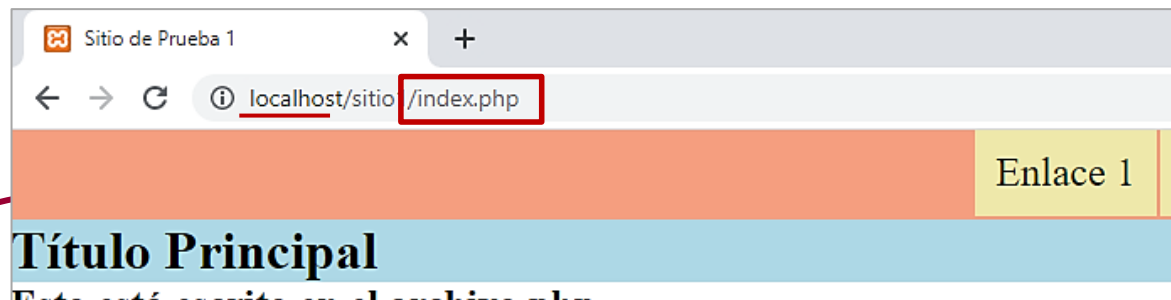
`localhost/ejemplos-clase01/index.php`

(Ojo! Xampp debe estar encendido,
con Apache en verde)



El código PHP se ve en el navegador?

Página
visualizada en el
navegador



Hacer clic
derecho y elegir
ver código
fuente de la
página

A screenshot of the browser's 'View Source' page. The address bar shows 'view-source:localhost/sitio1/index.php'. The code is displayed with line numbers 1 through 14. The code includes a DOCTYPE declaration, HTML lang attribute, head section with meta tags, and a body section with a menu container. A red box highlights the text 'En el navegador NO se ve el código PHP'.

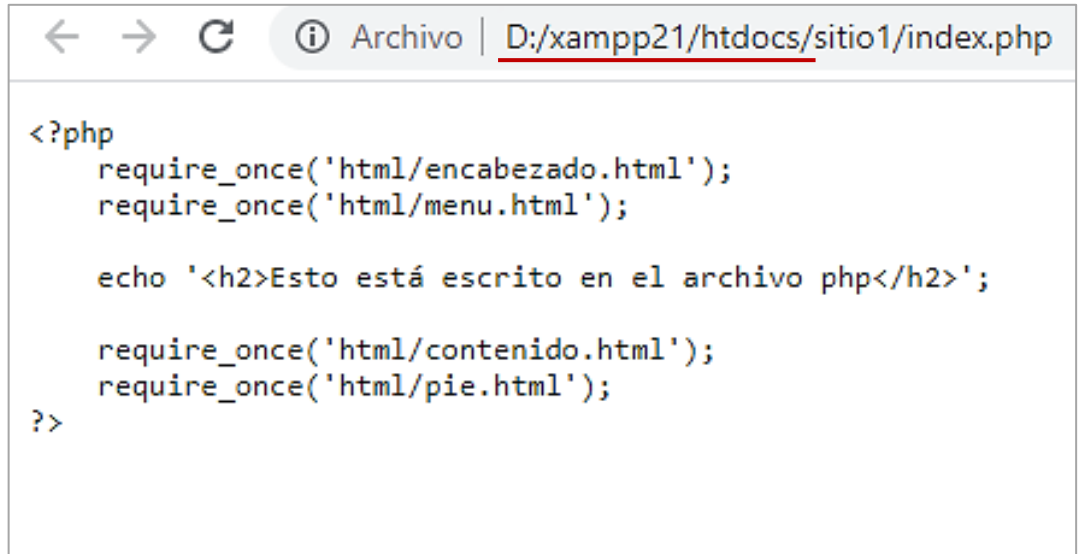
En el navegador NO se ve
el código PHP

Código generado
para el sitio

Qué pasa si no abrimos desde localhost?

- Si se abre una página .php como un sitio html, sin indicar que está dentro del localhost que pasará?
- El código no pasa por el intérprete PHP del servidor y por tanto no devuelve un código html que el navegador pueda mostrar como una web.

Así luce si no
abrimos
empezando por
localhost

A screenshot of a web browser window. The address bar shows the file path 'D:/xampp21/htdocs/sitio1/index.php'. The page content is raw PHP code, indicating that the browser is not interpreting the file as PHP. The code includes file inclusion statements and an echo statement.

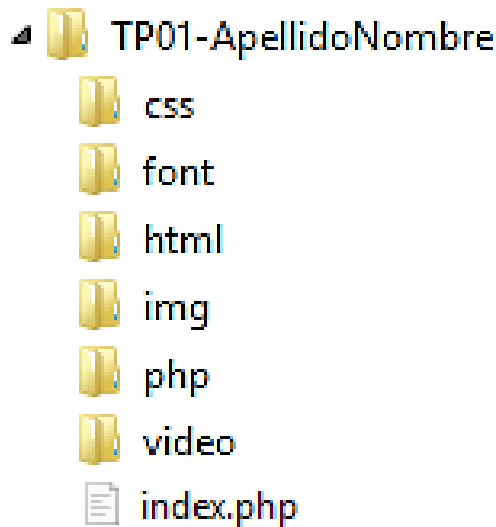
```
<?php
    require_once('html/encabezado.html');
    require_once('html/menu.html');

    echo '<h2>Esto está escrito en el archivo php</h2>';

    require_once('html/contenido.html');
    require_once('html/pie.html');
?>
```

Hacer siempre la jerarquía de carpetas

- Todas las carpetas que hagamos en los prácticos deberán estar contenidas dentro de **htdocs**, teniendo la jerarquía de carpetas ya vista en Laboratorio I adentro, como se ve en la figura.
- Sólo la página principal (**index.php**) va fuera de las carpetas.



Comentarios

- Que se haga costumbre comentar el código siempre

// este es un comentario de una línea

/* esto es un comentario
que puede
ocupar varias líneas */

Cuando escribí este código,
sólo Dios y yo sabíamos
como trabajaba



Ahora solo Dios lo sabe

*Lo que pasa cuando no
comentamos el código*

Uso de punto y coma ;

- Se recomienda escribir **una instrucción por línea**
- Todas las instrucciones de código PHP deben finalizar con **punto y coma**.
- Sino van a estar así....



Uso de punto y coma ;

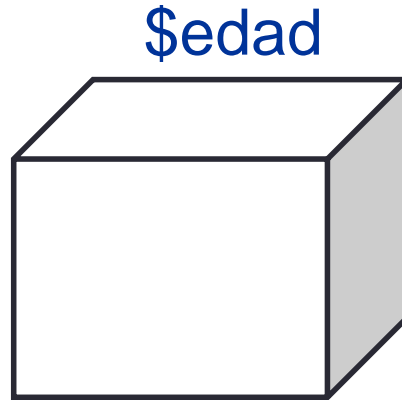
- O así... ¿Qué le falta a mi código que no anda?
Llevo noches sin dormir y ya no se que hacer. Dios,
por favor, si me estás escuchando dame una señal!!

La señal:



Variables: Qué son?

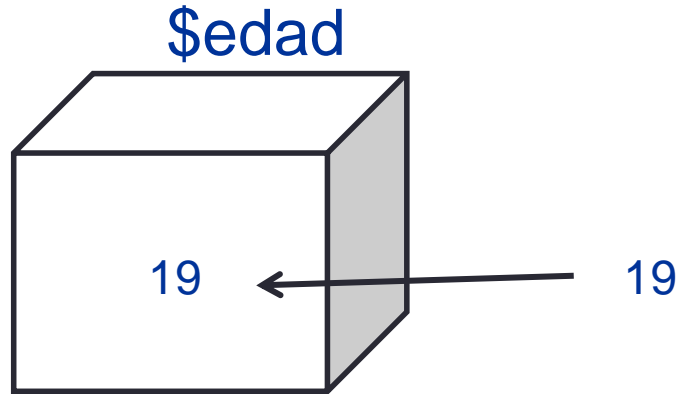
- Una variable es una posición de memoria, en la que se almacena un dato, que tiene asociado un nombre y un tipo de dato



19

Variables: Qué son?

- Una variable es una posición de memoria, en la que se almacena un dato, que tiene asociado un nombre y un tipo de dato



¿Qué tipo es?

Variables: Cómo escribirlas

- Deben comenzar siempre por \$, ejemplo: \$edad
- Luego pueden continuar con underscore (guión bajo) o letra. Ejemplo: \$_edad, \$edad
- A continuación puede contener más letras o números. Ejemplo: \$edad1
- Son sensibles a mayúsculas y minúsculas: No es lo mismo \$SUELDO1, \$sueldo1, \$Sueldo1

Variables: Cómo No escribirlas

- No se pueden usar símbolos **?**, **%**, **#**, *****, **&**, **@**, **!**, **(**, **)**, **<**, **>**, **+**, **-**, **/**, o espacios en blanco y no puede empezar con números
- Son nombres **inválidos**: ciudad, \$#orden, \$%descuento, \$fecha de nacimiento, \$123
- **\$this** no se puede usar como nombre porque está reservado para el lenguaje

Variables: Cómo escribirlas

- Son válidos estos nombres?

- \$_apellido
- \$telefono
- \$_123

- \$celular2
- \$dirección
- \$año

Si todas son válidos, pero no se recomienda usar acentos ni ñ.

Variables: Cómo escribirlas

- Que se observa cuando se usa un nombre indebido para las variables?

```
<?php
```

```
    $2daDireccion = 'Av. Independencia 1800';
```

```
    echo $2daDireccion; // se Escribe el contenido
```

```
?>
```

Parse error: syntax error, unexpected '2' (T_LNUMBER), expecting variable (T_VARIABLE) or '{' or '\$' in C:\xampp19\htdocs\EjemplosClase01\02-NombresVariables\index.php on line 9

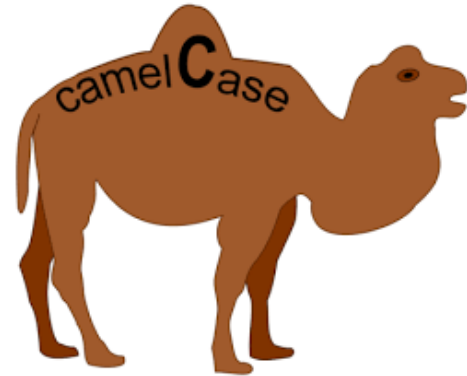
Variables: Cómo escribirlas

Parse error: syntax error, unexpected '2' (T_LNUMBER), expecting variable (T_VARIABLE) or '{' or '\$' in C:\xampp19\htdocs\EjemplosClase01\02-NombresVariables\index.php on line 9

Error de análisis: error de sintaxis, '2' inesperado (Tipo número), se esperaba tipo variable (Tipo variable) o '{' o '\$' en

Variables: Cómo escribirlas

- Se recomienda escribir las variables con más de una palabra con la forma de escritura conocida como **camelCase**: La primera letra de la 2da palabra y cada nueva palabra es con mayúsculas. Lo demás debe estar en minúsculas
- Ejemplo: \$telefonoFijoCasa



Variables: Tipos de Datos en PHP I

- PHP No Requiere, Ni Soporta definición explícita de tipo al crear una variable (no es como C)
- Presenta tipado dinámico (la variable toma un tipo de dato dependiendo de que lo se guardó en ella)
 - Ej:

```
<?php  
    $sueldo = 8500.45; // Qué tipo cree es?  
?>
```


Variables: Tipos de Datos en PHP II

- Tipos de datos Soportados
 - ✓ **integer**: Número Entero, valores posibles $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
 - ✓ **double**: Número con decimales, número real, valores posibles $\mathbb{R} = \{..., -1.005, -1.01, 0, 1, 1.01, 1.02,\}$
 - ✓ **string**: Tira de caracteres, valores posibles: "Hola", 'Luis Pérez', "Av. Independencia 1800", "25", 'La sumas', '4850.75'

Variables: Tipos de Datos en PHP II

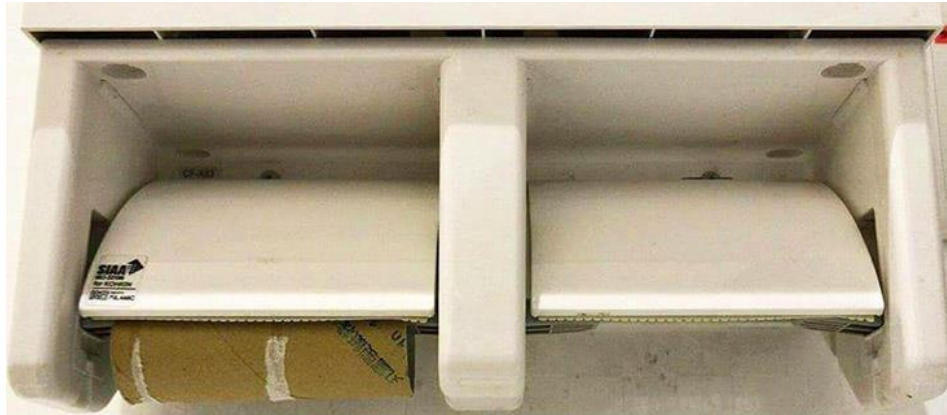
- Tipos de datos Soportados
 - ✓ **boolean**: Valores posibles **true** y **false** (verdadero y falso)
 - ✓ **array**: Arreglo o vector o matriz de una fila (profundizaremos en la siguiente clase)

Variables: Tipos de Datos en PHP III

- Tipos especiales
 - ✓ **NULL**: Valor nulo, significa ausencia de valor. Se da cuando no se ha asignado valor o se ha destruido la variable.
 - ✓ **resource**: para indicar los recursos externos que no se utilizan de manera nativa en PHP (al manipular archivos o bases de datos, por ejemplo).

Variables: Tipos de Datos en PHP IV

- Es lo mismo cero que **NULL**?



Cero Vs **NULL**

ESCRIBIR (mostrar en la página)

- Una de las formas de ESCRIBIR en PHP:

```
echo ( string $arg1 [, string $... ] ) : void
```

- Puede recibir más de un parámetro separado por comas y no devuelve nada.
- El código html que queramos mostrar deberá ir siempre entre comillas

Cómo mostrar comillas dentro de las comillas?

- Anteponer la barra invertida \ a las comillas para poder mostrarlas

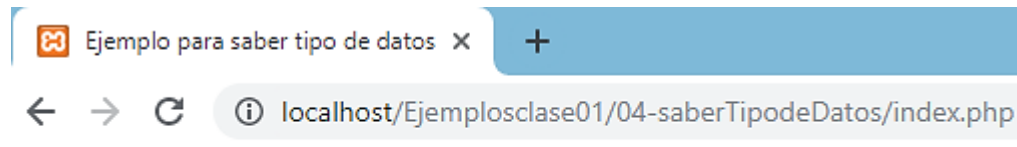
```
<?php
    require('html/header.html');
    $libro = 'La Voz Ausente';
    echo '<p>El Libro más vendido se llama \'' . $libro . '\'' de Gabriel Rolón</p>';
    require('html/footer.html');
?>
```

El libro más vendido se llama 'La Voz Ausente' de Gabriel Rolón

Cómo saber el Tipo de Dato en PHP?

- Se puede saber qué tipo de dato tomó una variable
`gettype($variable)`

```
$direccion = 'Av. Independencia 1800';  
$tipo = gettype($direccion);  
echo $direccion . ' es de tipo ' . $tipo;
```



Av. Independencia 1800 es de tipo string

Qué pasa al operar con distintos tipos de datos?

- Existen conversiones implícitas de tipos de datos que PHP maneja automáticamente.

```
<?php
    $num1 = '5.7';
    $num2 = 3;
    $suma = $num1 + $num2;
    echo '<p>Suma string con entero = ' . $suma . '</p>';

    $num3 = '5';
    $suma = $num1 + $num3;
    echo '<p>Suma string con string = ' . $suma . '</p>';

    $num4 = null;
    $suma = $num3 + $num4;
    echo '<p>Suma null con string = ' . $suma . '</p>';
?>
```

Suma string con entero = 8.7

Suma string con string = 10.7

Suma null con string = 5

Se puede usar una variable no inicializada?

- Qué pasa si tratamos de operar o mostrar una variable no inicializada?

```
<?php
    require('html/header.html');

    $apellido = 'Pérez'; // inicializamos $apellido
    echo '<p>Apellido: ' . $apellido . '</p>';
    echo '<p>Edad: ' . $edad . '</p>'; // $edad no se inicializó

    require('html/footer.html');
?>
```

Se puede usar una variable no inicializada?

- Qué pasa si tratamos de operar o mostrar una variable no inicializada?



Es buena práctica siempre inicializar las variables a utilizar

Formas de tener datos para trabajar (LEER)

- Existen 3 formas básicas en las que nuestro script PHP recibe un dato (LEER en Pseudocódigo)
 - Asignar **manualmente** el dato: `$num1 = 15;`
 - Generar un valor **aleatorio**:
`$num2 = mt_rand($valorInicial, $valorFinal);`
(por defecto devuelve valores enteros)
 - Recibir los datos de un **formulario** (veremos en otra clase)

Formas de tener datos para trabajar

- Ejemplos: Genere aleatoriamente la nota **entera** de un alumno.

```
<?php
    require('html/header.html');

    $inicio = 0;
    $fin = 10;
    $nota = mt_rand($inicio, $fin);
    echo '<p>La nota es: ' . $nota . '</p>';

    require('html/footer.html');
?>
```




Formas de tener datos para trabajar

- Ejemplo: Genere aleatoriamente la nota **con 2 decimales** de un alumno. (usamos 100 para 2 decimales, 10 para uno, etc.).

```
<?php
    require('html/header.html');

    $inicio = 0;
    $fin = 10;
    $nota = mt_rand($inicio * 100, $fin * 100) / 100;
    echo '<p>La nota es: ' . $nota . '</p>';

    require('html/footer.html');
?>
```



La nota es: 5.28



- La parte decimal se separan con . (esa es la "coma")

Cómo establecer el Tipo de Dato en PHP?

- Se puede establecer un tipo de dato nuevo a una variable existente. Existen dos formas:
 - `settype($variable, "tipo")`
 - `$variable = (tipo) $variable // se conoce como Casting`
- Esto puede resultar necesario cuando la realidad que se intenta modelar no tiene el tipo con el trabajamos, por ejemplo, si un script debe devolver una cantidad de personas y devuelve un valor con decimales

Cómo establecer el Tipo de Dato en PHP?

2374.25pesos es de tipo string

2374.25 es de tipo double

2374.25 es de tipo double

```
<?php
    $precio = '2374.25pesos';
    echo '<p>' . $precio . ' es de tipo ' . gettype($precio) . '</p>';
    $precio2 = (float)$precio;
    echo '<p>' . $precio2 . ' es de tipo ' . gettype($precio2) . '</p>';
    settype($precio, 'float');
    echo '<p>' . $precio . ' es de tipo ' . gettype($precio) . '</p>';
?>
```

Operadores Aritméticos

- Suma: $\$c = \$a + \$b$
- Resta: $\$c = \$a - \$b$
- Producto: $\$c = \$a * \$b$
- Cociente: $\$c = \$a / \$b$
- Cociente entero: $\$c = \text{intdiv}(\$a, \$b)$
- Resto de división entera: $\$c = \$a \% \$b$
- Potencia: $\$c = \$base ** \$exponente$
- Negación: $\$c = -\a

Operadores Aritméticos - Ejemplos

```
<?php
    $dividendo = 7;
    $divisor = 2;
    $cociente = $dividendo / $divisor;
    $cocienteEntero = intval($dividendo, $divisor);
    $resto = $dividendo % $divisor;
    echo '<p>Dividendo = ' . $dividendo . '</p>';
    echo '<p>Divisor = ' . $divisor . '</p>';
    echo '<p>Cociente = ' . $cociente . '</p>';
    echo '<p>Cociente entero = ' . $cocienteEntero . '</p>';
    echo '<p>Resto = ' . $resto . '</p>';
?>
```

Dividendo = 7

Divisor = 2


Cociente = 3.5

Cociente entero = 3

Resto = 1

Operadores Aritméticos - Ejemplos

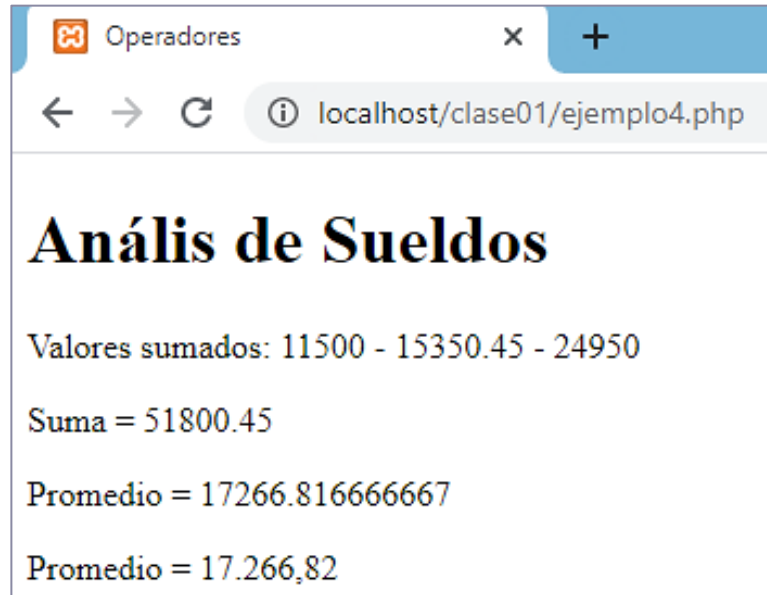
- Ingrese tres sueldos, entre 10000 y 25000, calcule la suma total y el promedio. (LEER con asignación manual)



```
<?php
    $sueldo1 = 11500;
    $sueldo2 = 15350.45;
    $sueldo3 = 24950;
    $suma = $sueldo1 + $sueldo2 + $sueldo3;
    $promedio = $suma / 3;
    echo '<p>Valores sumados: ', $sueldo1, ' - ', $sueldo2, ' - ', $sueldo3, '</p>';
    echo '<p>Suma = ' . $suma . '</p>';
    echo '<p>Promedio = ' , $promedio, '</p>'; // con muchos decimales
    echo '<p>Promedio = ' . number_format($promedio,2, ',', '.') . '</p>';
?>
```


Operadores Aritméticos - Ejemplos

- Ingrese tres sueldos, entre 10000 y 25000, calcule la suma total y el promedio. (LEER con asignación manual)



Operadores Aritméticos - Ejemplos

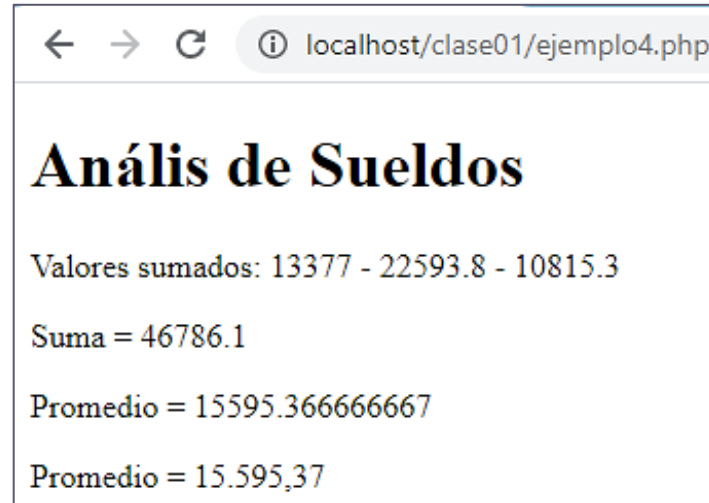
- Ingrese tres sueldos, entre 10000 y 25000, calcule la suma total y el promedio. (LEER con asignación aleatoria)



```
<?php
    $sueldo1 = mt_rand(10000*10, 25000*10)/10; // aleatorio con 1 decimal
    $sueldo2 = mt_rand(10000*10, 25000*10)/10;
    $sueldo3 = mt_rand(10000*10, 25000*10)/10;
    $suma = $sueldo1 + $sueldo2 + $sueldo3;
    $promedio = $suma / 3;
    echo '<p>Valores sumados: ', $sueldo1, ' - ', $sueldo2, ' - ', $sueldo3, '</p>';
    echo '<p>Suma = ' . $suma . '</p>';
    echo '<p>Promedio = ' , $promedio, '</p>'; // con muchos decimales
    echo '<p>Promedio = ' . number_format($promedio,2, ',', '.') . '</p>';
?>
```

Operadores Aritméticos - Ejemplos

- Ingrese tres sueldos, entre 10000 y 25000, calcule la suma total y el promedio. (LEER con asignación aleatoria)



A screenshot of a web browser window. The address bar shows 'localhost/clase01/ejemplo4.php'. The page content displays the title 'Análisis de Sueldos' in a large, bold, black serif font. Below the title, there are four lines of text in a smaller, black serif font: 'Valores sumados: 13377 - 22593.8 - 10815.3', 'Suma = 46786.1', 'Promedio = 15595.366666667', and 'Promedio = 15.595,37'.

```
< > ↻ ⓘ localhost/clase01/ejemplo4.php
```

Análisis de Sueldos

Valores sumados: 13377 - 22593.8 - 10815.3

Suma = 46786.1

Promedio = 15595.366666667

Promedio = 15.595,37

Operadores de Asignación I

- `$a = 5` `//` operando de la izquierda `$a` se establece con el valor de la expresión de la derecha `5`
- `$a += 4` `//` es como `$a = $a + 4`
- `$b = 'Hola';`
`$b .= ' Lorena';` `//` es como `$b = $b . ' Lorena'`
`echo $b;` `//` Devuelve `Hola Lorena`

Operadores de Asignación II

- $\$a \text{ -= } 4$ // es como $\$a = \$a - 4$
- $\$a \text{ *= } \b // es como $\$a = \$a * \$b$
- $\$a \text{ /= } \b // es como $\$a = \$a / \$b$
- $\$a \text{ %= } \b // $\$a = \$a \% \$b$

Operadores de incremento

- **\$a++** // Post-incremento: Retorna \$a, y luego incrementa \$a en uno.
- **++\$a** // Pre-incremento: Incrementa \$a en uno, y luego retorna \$a.

Operadores de incremento – Ejemplo I

```
<?php
    $a = 3;
    echo '<p>Resultado a): ' . $a++ . '</p>';
    echo '<p>Resultado b): ' . $a . '</p>';
    $a = 3;
    echo '<p>Resultado c): ' . ++$a . '</p>';
    echo '<p>Resultado d): ' . $a . '</p>';
?>
```

Resultado a): 3

Resultado b): 4

Resultado c): 4

Resultado d): 4

Operadores de incremento – Ejemplo II

```
<?php
    $a = 3;
    $b = $a++ + 2;
    echo $b; //    que tiene $b?
?>
```

5

Porque primero usa el valor de \$a y luego incrementa en 1

Operadores de decremento

- `--$a` Pre-decremento Decrementa \$a en uno, luego retorna \$a.
- `$a--` Post-decremento Retorna \$a, luego decrementa \$a en uno.

Constantes

- Las Constantes no pueden cambiar su valor durante la ejecución del script.
- Por defecto, una constante distingue mayúsculas y minúsculas.
- Por convención, los identificadores (nombres) de constantes siempre se declaran en mayúsculas.
- Para nombres compuestos utilizar guión bajo _
Ejemplo: LIMITE_INFERIOR

Diferencias entre constantes y variables

- Las constantes no llevan el signo (\$), como prefijo.
- Antes de PHP 5.3, las constantes solo podían ser definidas usando la función `define()`, y no por simple asignación con `const`.
- Las constantes no pueden ser redefinidas (asignadas con un nuevo valor) o eliminadas, una vez se han definido

Creación de Constantes

- `define('NOMBRE1', valor1);`
- `const NOMBRE2 = valor2; // desde PHP 5.3`

Observación: Si posee una versión vieja de xampp, es posible que `const` no funcione, entonces use `define`.

Creación de Constantes - Ejemplos

Uso de Constantes

Valor de PI: 3.1415926

Aceleración de la Gravedad: 9.8

```
<?php
    define('PI', 3.1415926);
    echo '<p>Valor de PI: ' . PI . '</p>';
    const ACELERACION_GRAVEDAD = 9.8; // en versiones nuevas de php
    echo '<p>Aceleración de la Gravedad: ' . ACELERACION_GRAVEDAD . '</p>';
?>
```

Creación de Constantes – Se pueden modificar?

```
<?php
    require('html/header.html');

    define('PI', 3.1415926);
    echo '<p>Valor de Pi= ' . PI;
    define('PI', 3.14);

    require('html/footer.html');
?>
```

Ejemplo de Constante

Valor de Pi= 3.1415926

Warning: Constant PI already defined in D:\xampp21\htdocs\ejemplos-clase01\sitio2\index.php on line 6

correo@gmail.com



Constante PI ya definida en

Ejemplo de uso de Constante

- Sea el valor de una pulgada 2,54 cm, realice un sitio que le permita generar las pulgadas de una pantalla de TV, entre 23 y 50 pulgadas, y las convierta a centímetros

```
<?php
    const PULG = 2.54;
    $diagonal = mt_rand(23, 50);
    $diagCm = $diagonal * PULG;
    echo '<p>Diagonal en ": ' . $diagonal . '</p>';
    echo '<p>Diagonal en cm: ' . number_format($diagCm, 2, ',', '.') . '</p>';
?>
```

Ejemplo de uso de Constante

- Sea el valor de una pulgada 2,54 cm, realice un sitio que le permita generar las pulgadas de una pantalla de TV, entre 23 y 50 pulgadas, y las convierta a centímetros



Dar formato para mostrar números

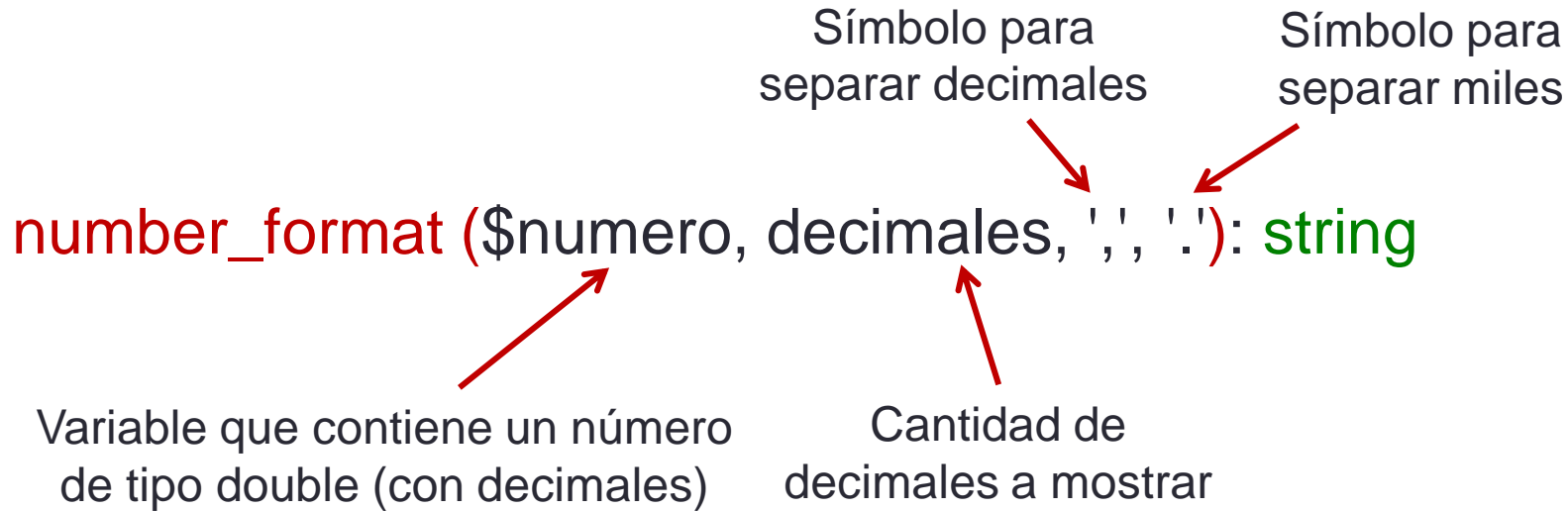
`number_format($numero, decimales, ',', '.')`: **string**

Variable que contiene un número de tipo double (con decimales)

Cantidad de decimales a mostrar

Símbolo para separar decimales

Símbolo para separar miles



The diagram illustrates the parameters of the `number_format` function. The function signature is shown in red and green text: `number_format($numero, decimales, ',', '.')`. Four red arrows point from descriptive text to the parameters:

- An arrow from "Variable que contiene un número de tipo double (con decimales)" points to `$numero`.
- An arrow from "Cantidad de decimales a mostrar" points to `decimales`.
- An arrow from "Símbolo para separar decimales" points to the comma `,`.
- An arrow from "Símbolo para separar miles" points to the period `.`.

 The return type `: string` is shown in green.

Dar formato para mostrar números

- **number_format** (\$numero, decimales, ',', '.'); // devuelve cadena con formato de número con miles y decimales

```
<?php
    define('PI', 3.14159);
    /* en versiones nuevas se puede usar
    const PI = 3.14159; */
    $radio = 20;
    $area = PI * $radio ** 2; // la potencia se resuelve primero
    echo '<p>Area = ' . $area . '</p>';
    $area2 = number_format($area, 2, ',', '.');
    echo '<p>Area2 = ' . $area2 . '</p>';
?>
```


Area = 1256.636

Area2 = 1.256,64

Cantidad de decimales

Creación de Tablas desde php

```
1  <?php
2      require('html/header.html');
3      $costoCortina = mt_rand(220,450);
4      $costoBaseGlobo = mt_rand(280,500);
5  ?>
6      <table>
7          <caption>Presupuesto de Cotillón</caption>
8          <thead>
9              <tr><th>Nombre</th><th>Precio Costo</th><th>Precio Venta</th></tr>
10         </thead>
11         <tbody>
12             <tr><td>Cortina roja</td>
13                 <td><?php echo $costoCortina; ?></td>
14                 <td><?php echo $costoCortina*2; ?></td>
15             </tr>
16             <tr><td>Base porta globo</td>
```



Creación de Tablas desde php

← → ↻ ⓘ localhost/ejemplo-tabla/ 🔍 ☆ ⚙

Ejemplo de tabla

Presupuesto de Cotillón

Nombre	Precio Costo	Precio Venta
Cortina roja	276	552
Base porta globo	325	650

correo@gmail.com

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sitio de Prueba de Tabla</title>
  <link rel="stylesheet" href="css/estilo.css">
</head>
<body>
  <header>
    <h1>Ejemplo de tabla</h1>
  </header>
  <main>
    <section>
      <article>

        <table>
          <caption>Presupuesto de Cotillón</caption>
          <thead>
            <tr><th>Nombre</th><th>Precio Costo</th><th>Precio Venta</th></tr>
          </thead>
          <tbody>
            <tr><td>Cortina roja</td>
              <td>276</td>
              <td>552</td>
            </tr>
            <tr><td>Base porta globo</td>
              <td>325</td>
              <td>650</td>
            </tr>
          </tbody>
        </table>

      </article>
    </section>
  </main>
</body>
</html>
```

Porcentaje

Obtención fórmula:

1500 --- 100%

\$x --- 21%

$$x = \frac{1500 * 21\%}{100\%}$$

$$x = \$1500 * \frac{21}{100}$$

$$x = 1500 * 21/100$$

Ejemplo de cálculo de porcentaje:

```
<?php
require_once('php/encabezado.php');
const IVA = 21;
$precio = 1500;
$ivaCalculado = $precio * IVA/100; // calcula el 21%
$totalConIva = $precio + $ivaCalculado;

echo '<h1>Precio con IVA</h1>';
echo '<p> Precio: ' . $precio . '</p>';
echo '<p> IVA del 21%: ' . $ivaCalculado . '</p>';
echo '<p> Precio con IVA: ' . $totalConIva . '</p>';

require_once('html/pie.php');

?>
```

Comprobación Teórica 1

<https://tinyurl.com/QR-Teo1>

