

Contenido

| | |
|--|----|
| INSTALACIÓN DE PHP | 2 |
| Windows..... | 2 |
| ELECCIÓN DEL ENTORNO DE DESARROLLO (IDE) | 5 |
| Sublime Text. | 5 |
| Uso de plantillas o Snippets. | 5 |
| ATOM..... | 6 |
| PhpStorm..... | 7 |
| HOLA MUNDO | 8 |
| CREANDO UN SERVIDOR, EN LA CONSOLA. | 9 |
| VARIABLES | 10 |
| Consideraciones. | 10 |
| Concatenación. | 10 |
| El uso de comillas simples y comillas dobles. | 11 |
| Operadores..... | 12 |
| PHP Y HTML 1 (FUNDAMENTOS) | 12 |
| Utilizando la URL para asignar valores a una variable PHP (\$_GET) | 16 |

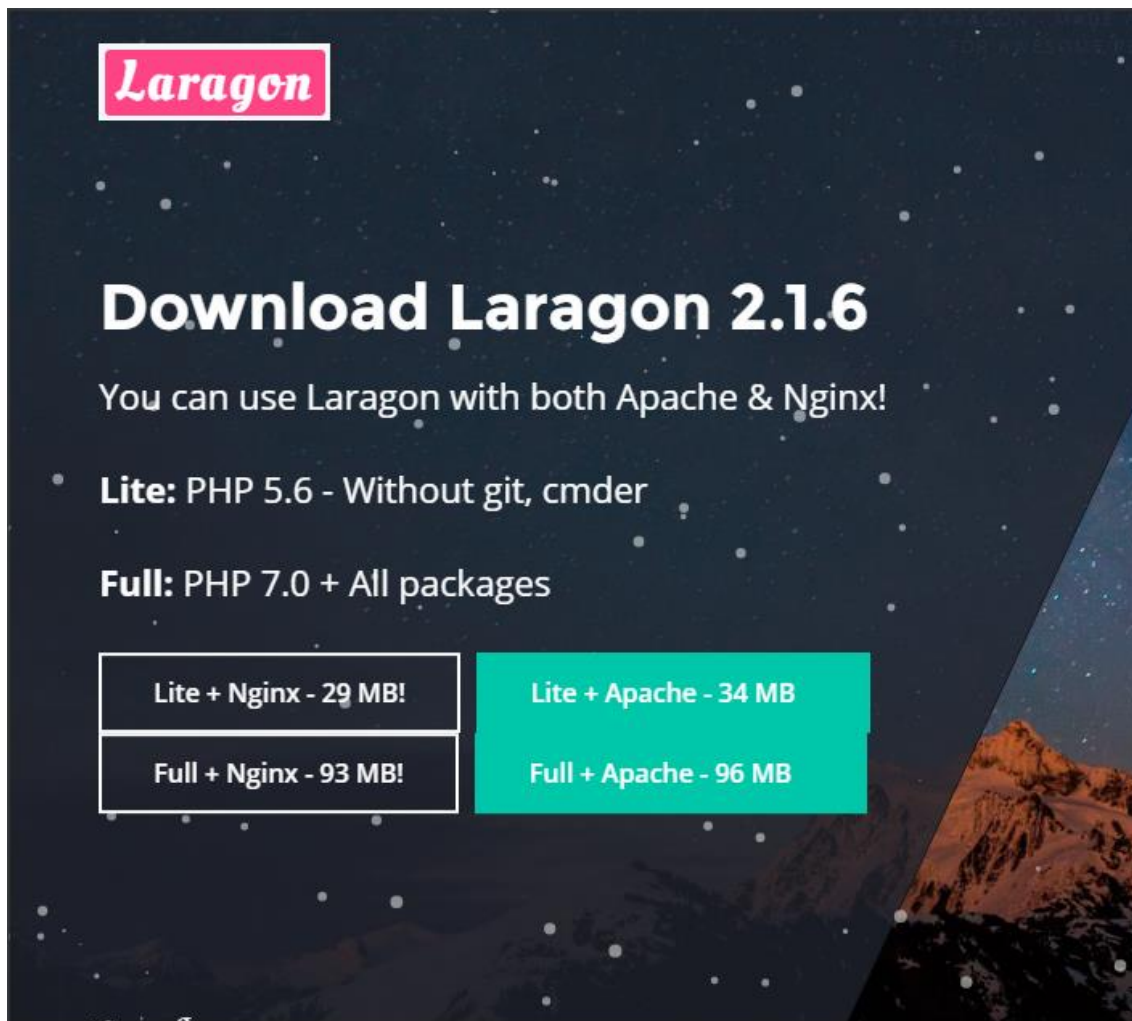
INSTALACIÓN DE PHP

Windows

Para utilizar php en Windows a modo de desarrollo se debe simplemente instalar una aplicación que contenga esta herramienta. Para este caso se utilizará Laragon.

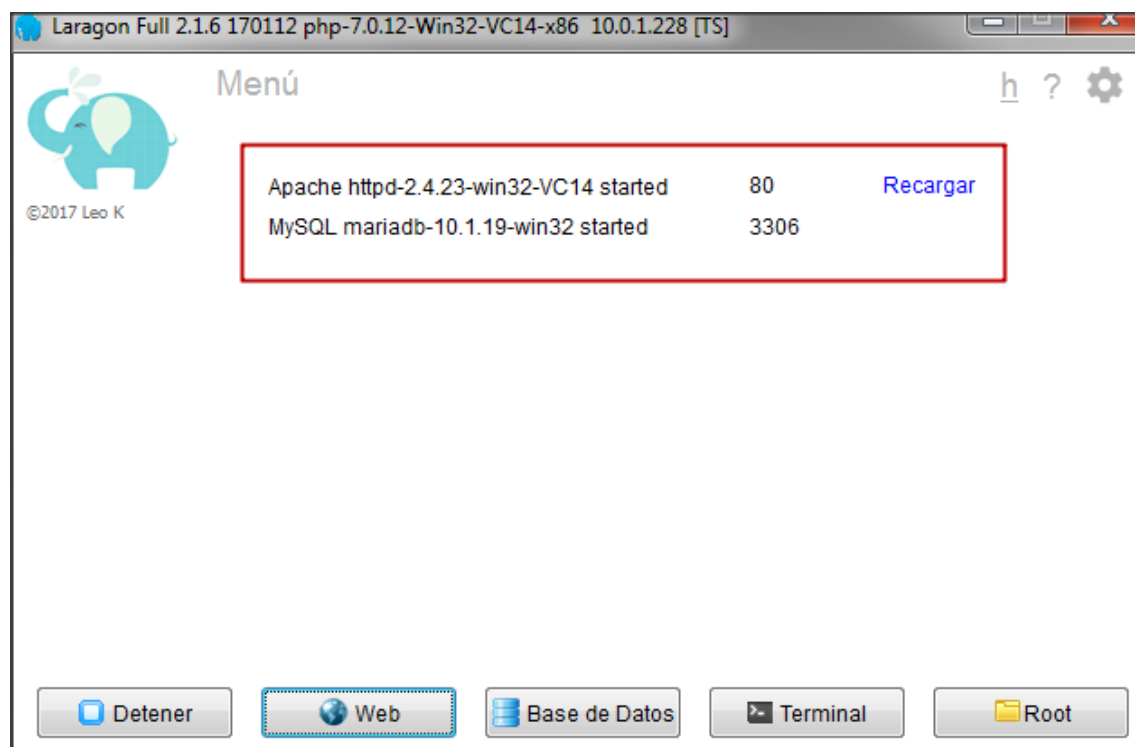
Para descargar esta aplicación se debe ingresar a:

<https://laragon.org/download.html>

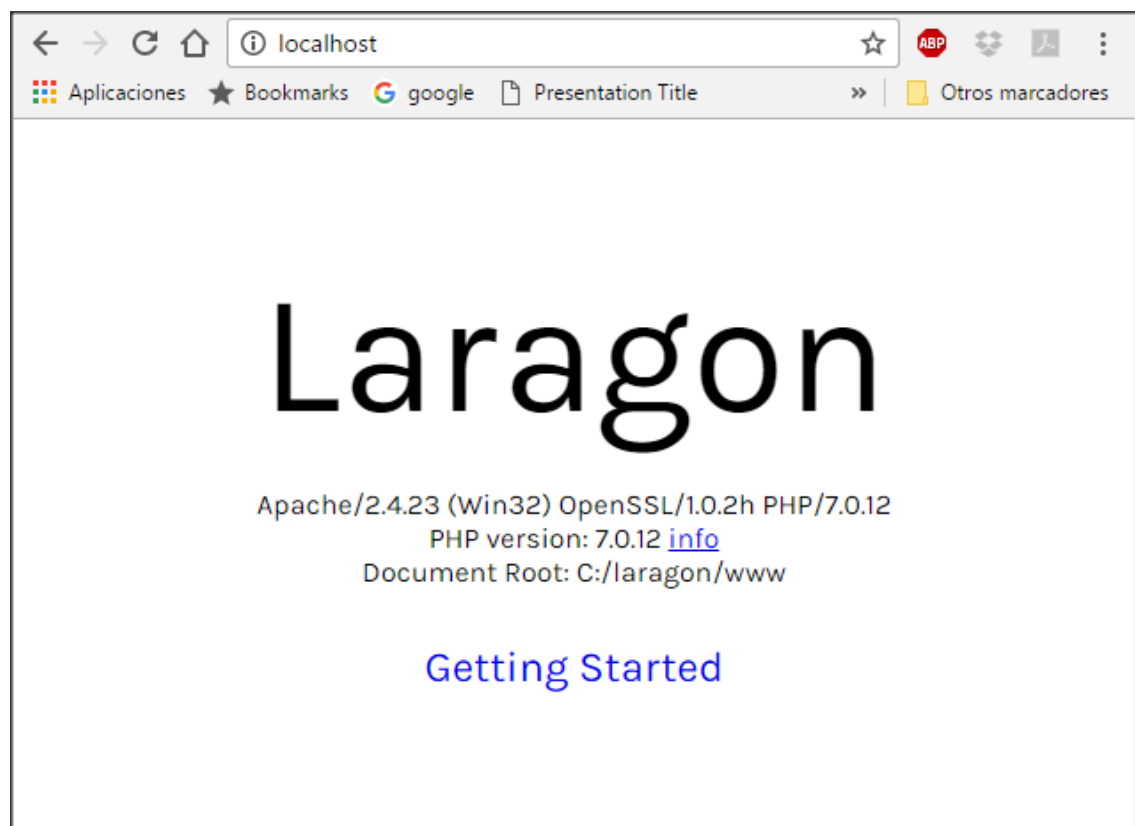


La que se ocupará será full + apache.

Una vez que los servicios están corriendo, se puede comenzar a correr código PHP.



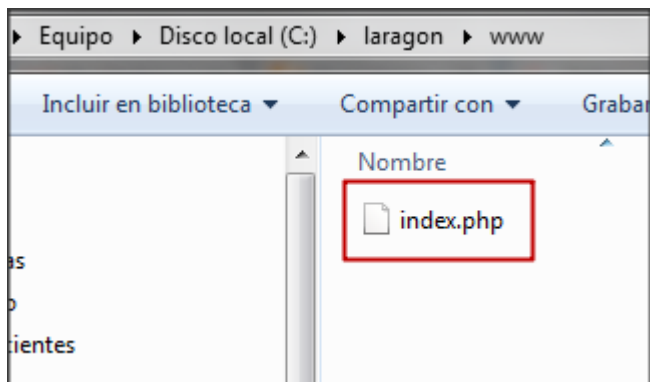
La dirección: `http://localhost/` simplemente `localhost`, nos lleva a la siguiente web:



Esta web se encuentra en el siguiente Path:

C:\laragon\www

El código PHP que muestra la web de localhost es la siguiente:



Si los servicios no estuvieran andando, entonces no se podría mostrar la web que figura en el ejemplo.

Existen más instaladores de servidores tales como MAMP o XAMP, etc. Básicamente son lo mismo instalan una variedad de herramientas con las que se puede desarrollar en PHP.

Si entramos en el terminal de Laragon podremos ver lo siguiente:

```
C:\laragon\www
λ php -v
PHP 7.0.12 (cli) (built: Oct 13 2016 11:04:23) ( ZTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies

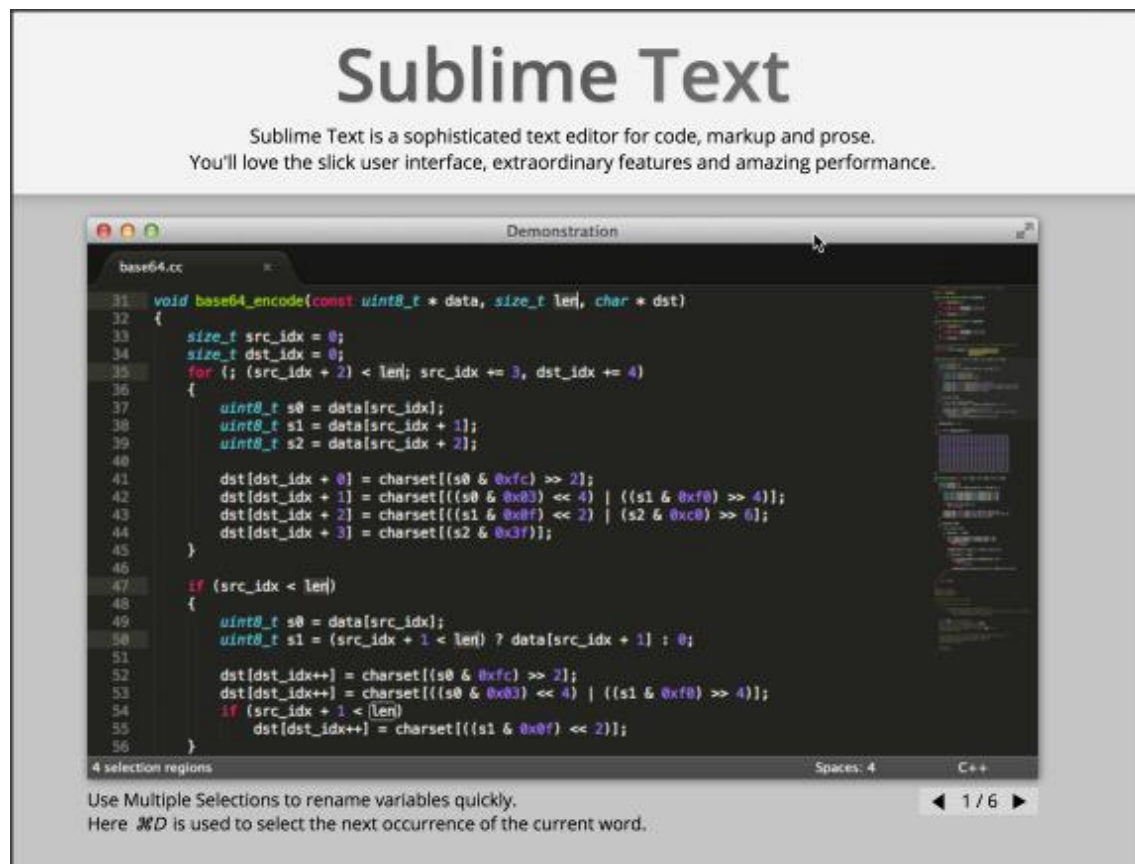
C:\laragon\www
λ |
```

Esto nos indica la versión de PHP que tenemos instalada.

ELECCIÓN DEL ENTORNO DE DESARROLLO (IDE)

Sublime Text.

Esta herramienta no es precisamente gratuita, pero se puede utilizar de manera completa, lo que si es que algunas veces cuando se guarde, nos pedirá que lo compremos.



La URL en donde se encuentra la descarga es:

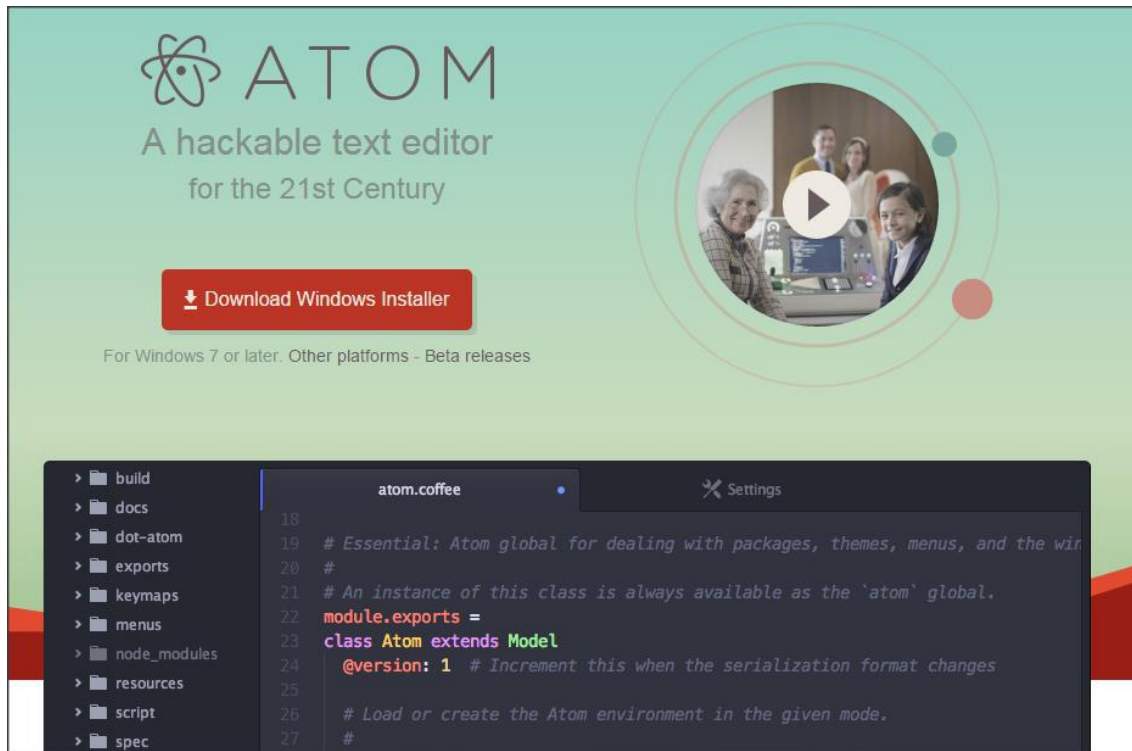
<https://www.sublimetext.com/>

Uso de plantillas o Snippets.

<http://sublimetext.info/docs/es/extensibility/snippets.html>

ATOM

Es completamente gratuito ya que está construido por la comunidad y está creciendo muy rápido.



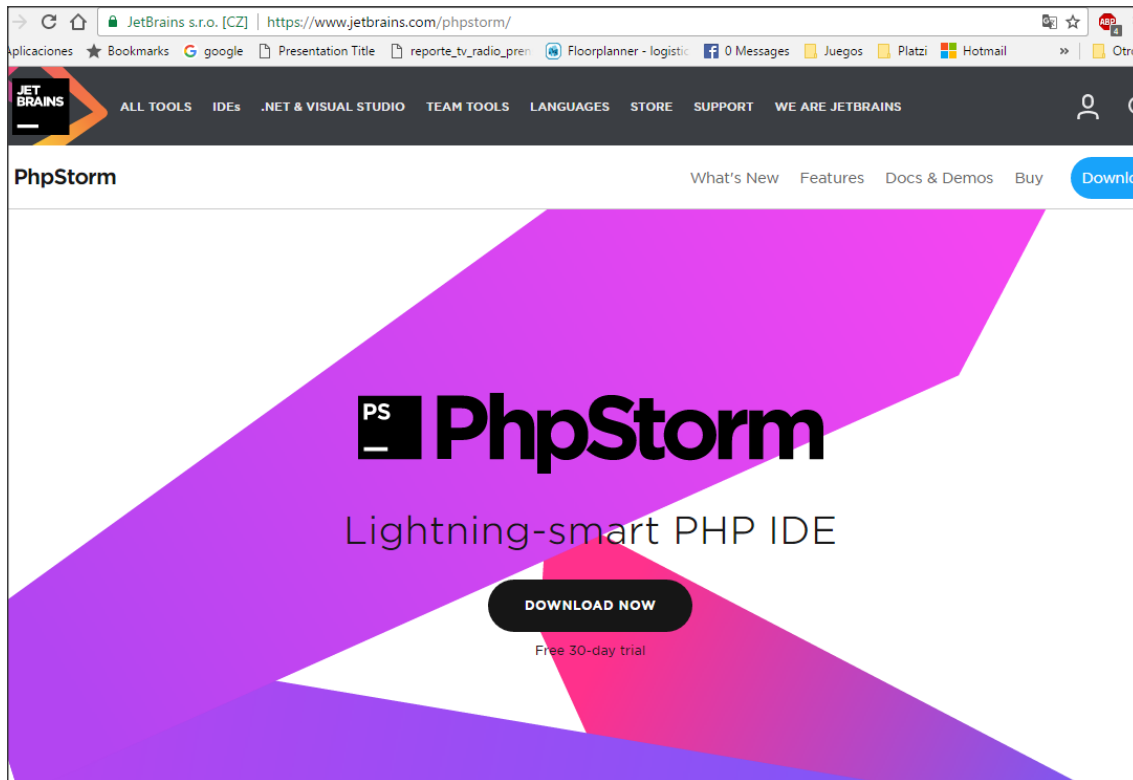
Link de la descarga está en:

<https://atom.io/>

PhpStorm

Es completamente pagado, sólo se puede usar gratis por 30 días en modo trial.

A diferencia de las otras dos herramientas, este IDE está dedicado a PHP por lo cual ya viene listo para utilizar.



HOLA MUNDO

La herramienta a utilizar será Sublime Text.

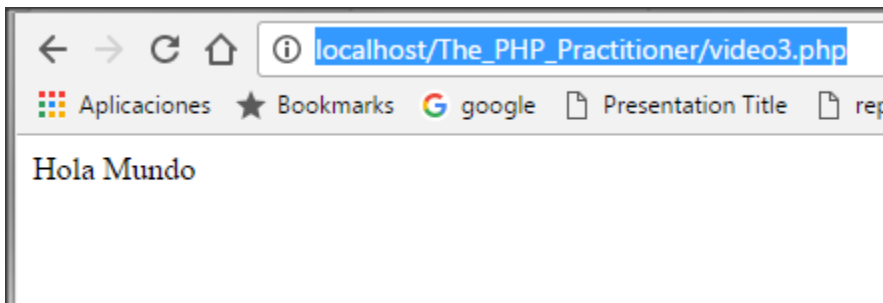
```
<?php  
echo "Hola Mundo";
```

Esto lo guardamos en:

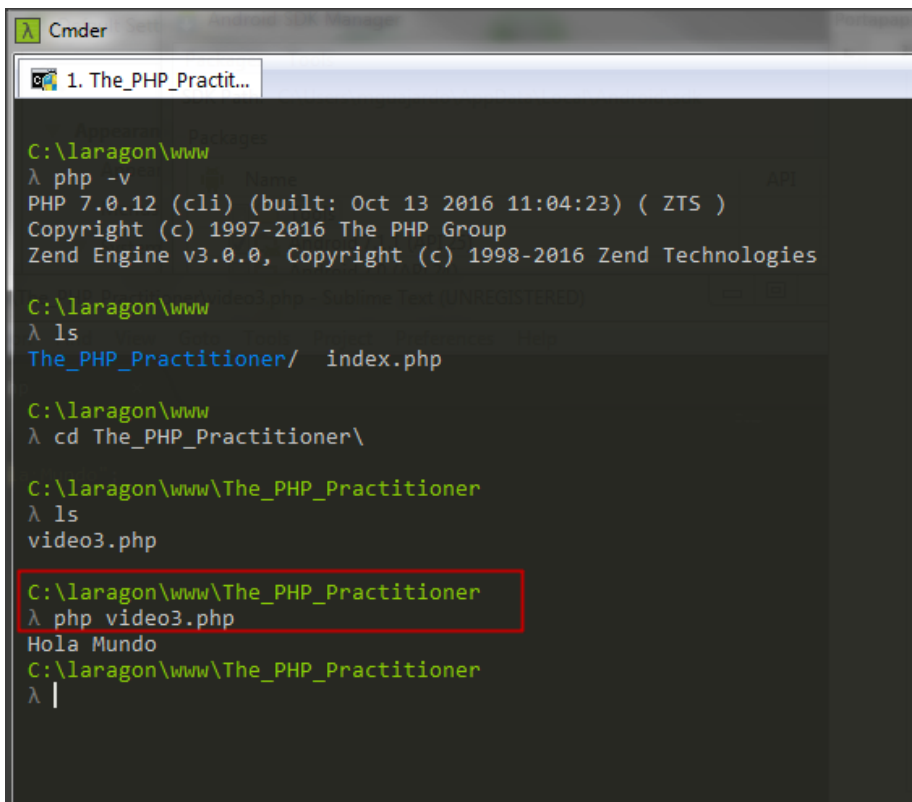
```
C:\laragon\www\The_PHP_Practitioner
```

Lo que arrojará el siguiente resultado.

http://localhost/The_PHP_Practitioner/video3.php



Además se puede ejecutar desde el Cmder de Laragon.

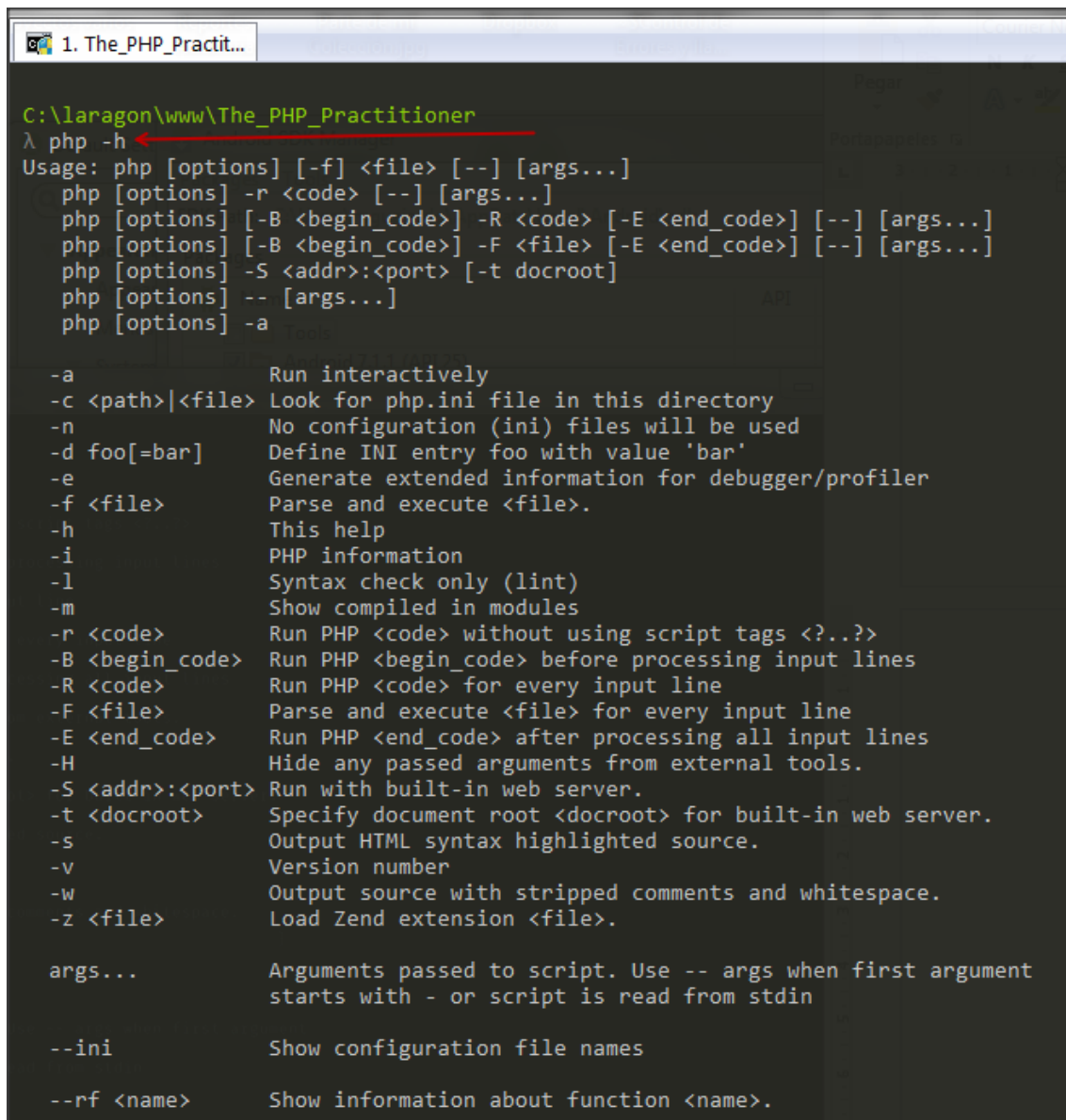


echo — Muestra una o más cadenas

<https://secure.php.net/manual/es/function.echo.php>

CREANDO UN SERVIDOR, EN LA CONSOLA.

Php se puede utilizar además desde la consola.



```
C:\laragon\www\The_PHP_Practitioner
λ php -h
Usage: php [options] [-f] <file> [--] [args...]
    php [options] -r <code> [--] [args...]
    php [options] [-B <begin_code>] -R <code> [-E <end_code>] [--] [args...]
    php [options] [-B <begin_code>] -F <file> [-E <end_code>] [--] [args...]
    php [options] -S <addr>:<port> [-t docroot]
    php [options] -- [args...]
    php [options] -a

-a          Run interactively
-c <path>|<file> Look for php.ini file in this directory
-n          No configuration (ini) files will be used
-d foo=<bar> Define INI entry foo with value 'bar'
-e          Generate extended information for debugger/profiler
-f <file>   Parse and execute <file>.
-h          This help
-i          PHP information
-l          Syntax check only (lint)
-m          Show compiled in modules
-r <code>   Run PHP <code> without using script tags <?...?>
-B <begin_code> Run PHP <begin_code> before processing input lines
-R <code>   Run PHP <code> for every input line
-F <file>   Parse and execute <file> for every input line
-E <end_code> Run PHP <end_code> after processing all input lines
-H          Hide any passed arguments from external tools.
-S <addr>:<port> Run with built-in web server.
-t <docroot> Specify document root <docroot> for built-in web server.
-s          Output HTML syntax highlighted source.
-v          Version number
-w          Output source with stripped comments and whitespace.
-z <file>   Load Zend extension <file>.

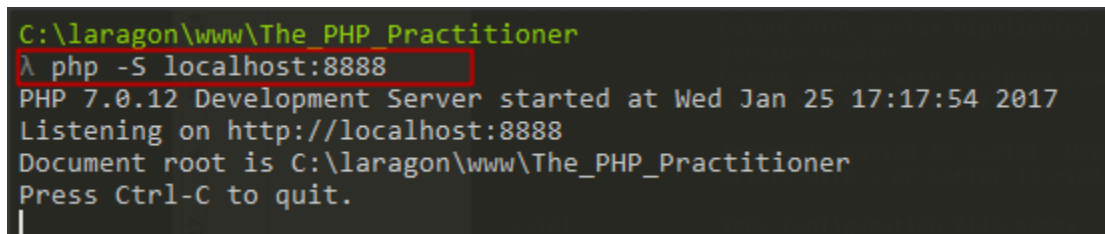
args...    Arguments passed to script. Use -- args when first argument
           starts with - or script is read from stdin

--ini       Show configuration file names

--rf <name> Show information about function <name>.
```

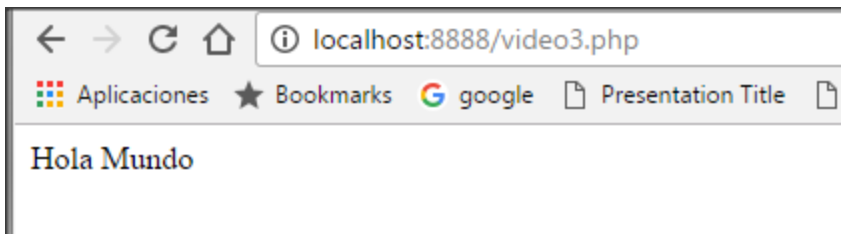
Se puede ver que con el comando `php -h` aparecen muchas opciones para la utilización de PHP desde la consola.

Utilizaremos un ejemplo, corriendo un servidor desde la consola.



```
C:\laragon\www\The_PHP_Practitioner
λ php -S localhost:8888
PHP 7.0.12 Development Server started at Wed Jan 25 17:17:54 2017
Listening on http://localhost:8888
Document root is C:\laragon\www\The_PHP_Practitioner
Press Ctrl-C to quit.
```

Luego, utilizando el navegador:



Se ha creado un servidor a partir de la consola.

VARIABLES

<https://secure.php.net/manual/es/language.variables.php>

En PHP las variables se representan con un signo de dólar \$ seguido por el nombre de la variable. **El nombre de la variable es sensible a minúsculas y mayúsculas.**

Los nombres de variables siguen las mismas reglas que otras etiquetas en PHP. Un nombre de variable válido tiene que empezar con una letra o un carácter de subrayado (underscore), seguido de cualquier número de letras, números y caracteres de subrayado.

```
<?php
$var = 'Roberto';
$Var = 'Juan';
echo "$var, $Var";      // imprime "Roberto, Juan"

$4site = 'aun no';      // inválido; comienza con un número
$_4site = 'aun no';     // válido; comienza con un carácter de
subrayado
$täyte = 'mansikka';    // válido; 'ä' es ASCII (Extendido) 228
?>
```

`$this` es una variable especial que no puede ser asignada. Existen varias funciones de manejo de variables.

<https://secure.php.net/manual/es/ref.var.php>

Consideraciones.

Concatenación.

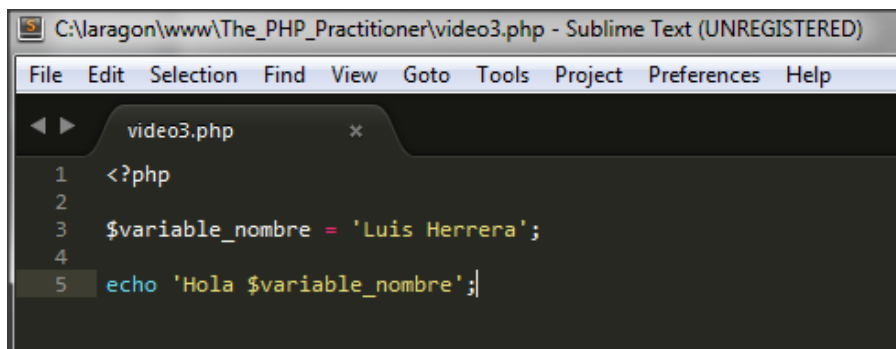
La concatenación es necesaria para unir cadenas de caracteres, su operador es un punto.

```
$var_1 = 'hola';
$var_2 = 'mundo';
$saludo = $var_1.' '.$var_2;
```

Esto da como resultado: hola mundo

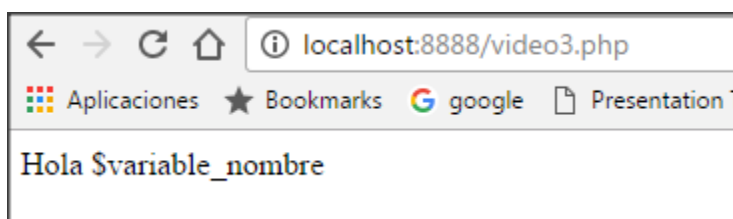
El uso de comillas simples y comillas dobles.

En PHP se puede representar una cadena de caracteres (String) utilizando `'` o `"`, la diferencia entre utilizar comillas simples y comillas dobles, se puede ver en la siguiente imagen.

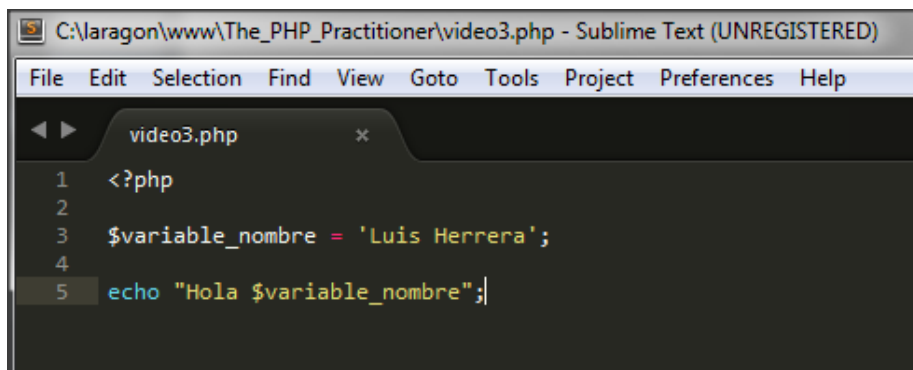


```
C:\laragon\www\The_PHP_Practitioner\video3.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
video3.php x
1 <?php
2
3 $variable_nombre = 'Luis Herrera';
4
5 echo 'Hola $variable_nombre';
```

Esto da como resultado.

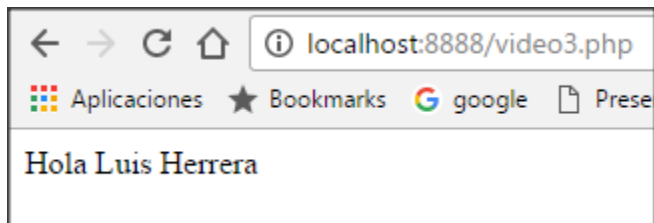


En cambio si lo hacemos con comillas dobles.

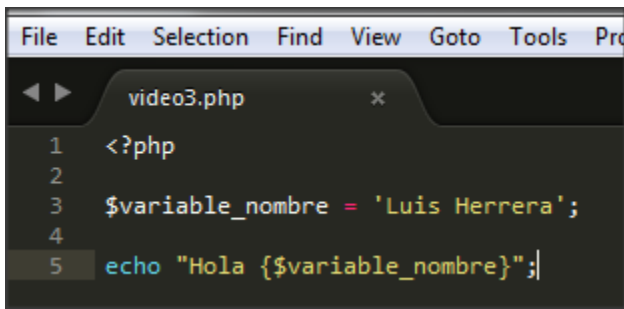


```
C:\laragon\www\The_PHP_Practitioner\video3.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
video3.php x
1 <?php
2
3 $variable_nombre = 'Luis Herrera';
4
5 echo "Hola $variable_nombre";
```

El resultado es el siguiente.



Nota: si se va a usar comillas dobles para usar variables, es conveniente usar llaves.



```
File Edit Selection Find View Goto Tools Pro
video3.php x
1 <?php
2
3 $variable_nombre = 'Luis Herrera';
4
5 echo "Hola {$variable_nombre}";
```

Operadores

Además de concatenar caracteres, PHP puede operar de diferentes formas sus variables.

<http://php.net/manual/es/language.operators.php>

PHP Y HTML 1 (FUNDAMENTOS)

Cuando se tiene un archivo que contiene solo PHP, no es necesario cerrar el TAG **<?php**, de esta forma **<?php ?>**.

Si dentro de un Código HTML por ejemplo, se inserta código php entonces se debe abrir y cerrar el tag.

La forma de incluir PHP en HTML puede variar dependiendo la complejidad del código PHP que se está utilizando.

Ej. 1

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>

<style type="text/css">
    header{
        background: #e3e3e3;
        padding: 2em;
        text-align: center;
    }
</style>

<body>
<header>
    <h1>
        <?php echo "hola mundo"; ?>
    </h1>
</header>
</body>
</html>
```

El resultado es el siguiente.



Ej. 2

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>

<style type="text/css">

  header{
    background: #e3e3e3;
    padding: 2em;
    text-align: center;
  }
</style>

<body>
<header><h1><?="hola mundo"; ?></h1></header>
</body>
</html>
```

Ej. 3

```
<?php
$saludo = "hola mundo";
?>

<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>

<style type="text/css">

    header{
        background: #e3e3e3;
        padding: 2em;
        text-align: center;
    }
</style>

<body>
<header><h1><?=$saludo; ?></h1><br>
    <h6><?php echo $saludo; ?></h6>
</header>
</body>
</html>
```



Se puede crear un archivo externo con las variables e importarlo para utilizarlo.

Ej. 4

Se crea un archivo llamado variables.php

```
<?php

$saludo = "hola mundo";
```

Se puede ver que al ser un archivo php puro, este no se cierra, solo posee el tag de apertura.

Luego en el archivo index...

```
<?php
include 'variables.php';

?>
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>

<style type="text/css">
    header{
        background: #e3e3e3;
        padding: 2em;
        text-align: center;
    }
</style>

<body>
<header><h1><?=$saludo; ?></h1><br>
    <h6><?php echo $saludo; ?></h6>
</header>
</body>
</html>
```

Include se utiliza para incluir el archivo que se utilizará para usar las variables, es como si estuviera en el mismo archivo index.

Utilizando la URL para asignar valores a una variable PHP (`$_GET`)

Podemos asignar valores a una variable en php agregando ese valor a la URL.

```
http://localhost:8888/php_y_html/index.php?var_1=valorDesdeLaURL
```

Al poner esto desde la URL, no hay cambio en el resultado de la web.



Para capturar el valor que entregamos en la URL, se debe hacer lo siguiente.

En el archivo variables.php

```
<?php
$saludo      = "hola mundo";
$var_desde_url = $_GET['var_1'];
```


Luego en el archivo index.php.

```
<?php
include 'variables.php';
?>
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>

<style type="text/css">

    header{
        background: #e3e3e3;
        padding: 2em;
        text-align: center;
    }
</style>

<body>
<header><h1><?=$var_desde_url; ?></h1><br>
</header>
</body>
</html>
```

Se obtiene el siguiente resultado.



El valor que obtendrá la variable es capturada en el arreglo \$_GET, este puede tener uno o varios valores, es una cadena de **clave->valor** que en este caso fue **var_1->valorDesdeURL**.

Desde la URL se puede escribir todo tipo de String para pasarlo a la variable y luego esto será interpretado por el servidor. El código HTML es una cadena de caracteres, por tanto, se puede pasar código HTML complejo y este tomará efecto en el comportamiento de la web. Para evitarlo se debe efectuar lo siguiente.

El archivo variables.php

```
<?php
$saludo = "hola mundo";
$var_desde_url = htmlspecialchars($_GET['var_1']);
```

Al agregar esta función, el html ingresado por el usuario no tendrá efecto de funcionalidad y solo se imprimirá como texto.

Antes.



Después

