

Programación en PHP

Daniela Juliana Sánchez Bayona- 1151947

Brayan Jesús García Urbina -1151936

Jesús Sebastián Bermon Bermudez-1151900

Historia de PHP

PHP tal y como se conoce hoy en día es en realidad el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios Common Gateway Interface (CGI) escritos en el lenguaje de programación C. Originalmente utilizado para rastrear visitas de su currículum online, llamó al conjunto de scripts "Personal Home Page Tools", más frecuentemente referenciado como "PHP Tools".



Versión Vigente de PHP

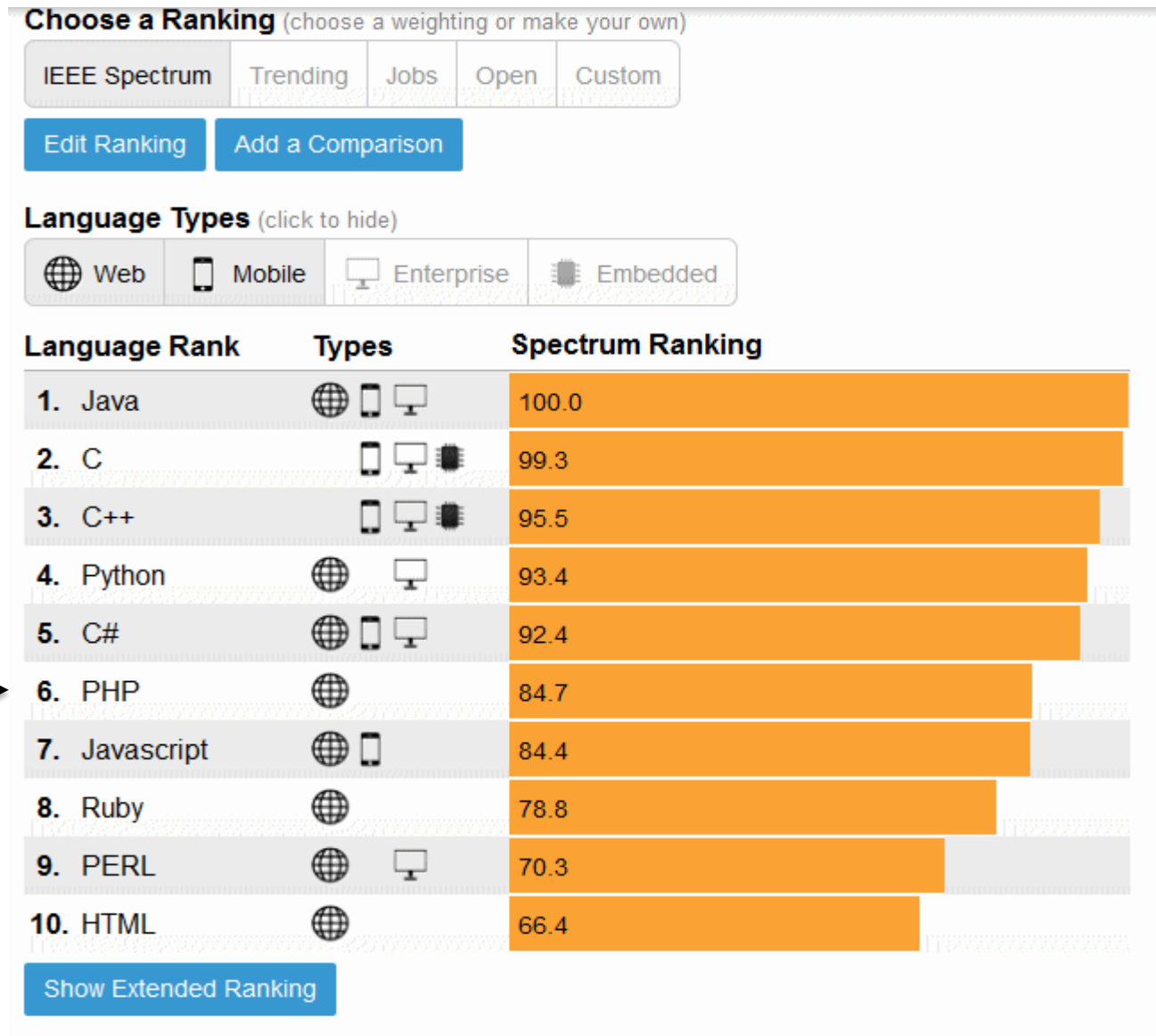
Ultima versión estable: 7.3.11 (24 de octubre de 2019)(25 días))

	Significado
Rojo	Versión antigua, sin soporte
Amarillo	Con soporte para errores de seguridad
Verde	Con soporte para errores generales y de seguridad
Azul	Lanzamiento futuro
Rosa	Lanzamiento cancelado

Versión	Fecha de lanzamiento	Fin de soporte	Notas
1.0	08-06-1995		Llamada oficialmente "Personal Home Page Tools (PHP Tools)". Es la primera versión en usar el nombre "PHP". ¹⁹
2.0	01-11-1997		Considerada por sus creadores la "herramienta más rápida y sencilla" para crear páginas web dinámicas. ¹⁹
3.0	06-06-1998	20-10-2000	Zeev Suraski y Andi Gutmans reescribieron la base para esta versión. ¹⁹
4.0	22-05-2000	23-01-2001	Se añadió un sistema de parsing de dos fases llamado motor Zend. ²⁰
4.1	10-12-2001	12-03-2002	Se introducen 'superglobales' (\$_GET, \$_POST, \$_SESSION, etc.) ²⁰
4.2	22-04-2002	06-09-2002	Se deshabilitan register_globals por defecto. Datos recibidos por la red no son insertados en el espacio de nombres global, cerrando posibles agujeros de seguridad en las aplicaciones. ²⁰
4.3	27-12-2002	31-03-2005	Se introduce CLI, y CGI. ^{20 21}
4.4	11-07-2005	07-08-2008	Se añaden páginas man para phpize y php-config. ²⁰
5.0	13-07-2004	05-09-2005	Motor Zend II con un nuevo modelo de objetos. ²²
5.1	24-11-2005	24-08-2006	Mejoras de rendimiento con la introducción de variables de compilador para el nuevo motor de PHP. ²²
5.2	02-11-2006	06-01-2011	Activada extensión de filtro por defecto. Soporte JSON nativo. ²²
5.3	30-06-2009	14-08-2014 ²³	Soporte para espacios de nombres, enlace estático en tiempo de ejecución, etiqueta de salto (GOTO limitada), Clausuras nativas, Soporte nativo para archivos PHP (phar), recolección de basura para referencias circulares, soporte mejorado para Windows, sqlite3, mysqlnd como reemplazo a libmysql como biblioteca para extensiones que funcionan con MySQL, fileinfo como reemplazo de mime_magic para mejor soporte MIME, extensión de Internacionalización, y etiqueta preg_replace_callback.

5.4	01-03-2012	15-09-2015	Soporte para <code>Trait</code> y sintaxis abreviada de array. Elementos eliminados: <code>register_globals</code> , <code>safe_mode</code> , <code>allow_call_time_pass_reference</code> , <code>session_register()</code> , <code>session_unregister()</code> y <code>session_is_registered()</code> . Servidor web incorporado. ²⁴ Varias mejoras a características existentes y al rendimiento, y requerimientos de memoria menores.
5.5	20-06-2013	20-06-2016	Nuevos generadores para bucles, <code>empty()</code> soporta expresiones. Se pierde el soporte para Windows XP y Windows Server 2003. ²⁵
5.6	20-08-2014	31-12-2018	Constantes con expresiones escalares, listas de argumentos de longitud variable y exponencial mediante el operador <code>**</code> . ²⁶
6			El desarrollo de PHP 6 ha sido retrasado porque los desarrolladores decidieron que el enfoque actual para tratar cadenas Unicode no es correcto, y están considerando formas alternas para la siguiente versión. ²⁷ Las mejoras planeadas para PHP 6 fueron añadidas en su lugar en PHP 5.3.0 (Soporte para espacios de nombre, enlace estático en tiempo de ejecución, funciones lambda, clausuras, goto) y 5.4.0 (traits, revinculación de clausura). Se abandonó la versión 6 de PHP siendo la siguiente versión la 7. ²⁸
7.0	03-11-2015 ²⁹	03-12-2018	Mejoras de rendimiento, ³⁰ declaración de tipos de retorno en funciones ³¹
7.1	01-12-2016 ³² 33	01-12-2019	Tipo de retorno vacío, ³⁴ Clase constante modificadores de visibilidad, ³⁵ Tipos nullable, ³⁶ iterable de tipo seudo, ³⁷ Captura múltiples tipos de excepciones. ³⁸
7.2	30-11-2017	30-11-2020	Nuevo tipo de retorno "object", ³⁹ integración de Libsodium como una extensión del núcleo ⁴⁰

Posición dentro del Ranking



Utilidad del lenguaje

Funcionamiento

- Enviamos una petición al servidor, ejemplo www.ibrugor.com/blog/index.php
- C v
- El servidor recibe la petición y busca la página a entregar.
- Si la página contiene la extensión “.php”, el intérprete de PHP la procesa.
- El servidor ejecuta el código PHP de la página y prepara el resultado final, el HTML.
- Se envía la página HTML al cliente final.

PHP se utiliza para generar páginas web dinámicas. Por lo tanto, podemos vincularlo a una base de datos para que su contenido sea cambiante. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, ingreso de información, etc.



Ventajas

1. Lenguaje totalmente libre y abierto
2. Curva de aprendizaje muy baja
3. Los entornos de desarrollo son de rápida y fácil configuración
4. Fácil despliegue: paquetes totalmente autoinstalables que integran PHP
5. Fácil acceso a bases de datos
6. Comunidad muy grande



Desventajas

- Debes saber cuando menos HTML para poder hacer un trabajo medianamente funcional.
- Si no lo configuras correctamente dejas abiertas muchas brechas de seguridad.
- Se necesita instalar un servidor web.



Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender..
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.



Clases y Objetos en PHP

Clase: es un término genérico que, como su nombre lo indica, representa una clase, tipo o conjunto. Una clase es una construcción que permite crear tipos personalizados propios mediante la agrupación de variables de otros tipos, métodos y eventos..

```
1 <?php
2
3 class Person
4 {
5
6 }
```

```
1 <?php
2
3 class Person
4 {
5     public $firstName = 'John';
6     public $lastName = 'Doe';
7     private $dateOfBirth = '1980-12-01';
8 }
```

Propiedades

- public quiere decir que la propiedad se puede acceder y modificar fuera del «molde» (esqueleto o contexto) de la clase.
- private indica que no es posible ni leer ni reescribir la propiedad fuera de la clase y esto nos trae un nuevo concepto muy importante llamado encapsulamiento.

Métodos en PHP

```
1 <?php
2
3 class Person
4 {
5     public $firstName = 'John';
6     public $lastName = 'Doe';
7     private $dateOfBirth = '1980-12-01';
8
9     public function fullName()
10    {
11        return "{$this->firstName} {$this->lastName}";
12    }
13
14    public function age()
15    {
16        // Calculate age using $this->dateOfBirth...
17        return $age;
18    }
19 }
```

- En este contexto un método es una función que en su interior tiene definido un conjunto de instrucciones.
- El método tiene un nombre para identificarlo.
- Podemos hacer que requiera datos de entrada de distinto tipo para ejecutarse. Y podemos hacer que el método devuelva como resultado un dato.
- Cuando necesitamos ejecutar las instrucciones que contiene el método, lo hacemos simplemente utilizando su nombre

CONSTRUCTORES

```
1 <?php
2
3 class Person
4 {
5     public $firstName = 'John';
6     public $lastName = 'Doe';
7     private $dateOfBirth = '1980-12-01';
8
9     public function __construct($firstName, $lastName, $dateOfBirth)
10    {
11        $this->firstName = $firstName;
12        $this->lastName = $lastName;
13        $this->dateOfBirth = $dateOfBirth;
14    }
15
16    public function fullName()
17    {
18        return "{$this->firstName} {$this->lastName}";
```

El **constructor** es un método especial dentro de una clase, se suele utilizar para darle un valor a los atributos del objeto al crearlo.

Es el primer método que se ejecuta al crear el objeto y se llama automáticamente al crearlo.

Este método puede recibir parámetros como cualquier otro método y para pasárselos tenemos que pasarle los parámetros al objeto.

El constructor **NO DEVUELVE** ningún dato.

Métodos Getter and Setter

```
public function getDateOfBirth()  
{  
    return $this->dateOfBirth;  
}  
  
public function setDateOfBirth($date)  
{  
    $this->dateOfBirth = $date;  
}  
}
```

- Los getters o setters son métodos comunes que nos permiten obtener o reemplazar el valor de una propiedad, respectivamente:
- Con getDateOfBirth obtenemos la fecha de nacimiento de la persona.
- Con setDateOfBirth reemplazamos la fecha de nacimiento de la persona.

Herencia

Se denomina herencia a la capacidad de una clase de emplear todos los métodos y propiedades de otra clase, lo cual se consigue en PHP mediante la palabra reservada `extends`.

```
class operacion
{
    protected $numero1;
    protected $numero2;
    public function __construct($numero1, $numero2) {
        $this->numero1 = $numero1;
        $this->numero2 = $numero2;
    }
    public function operacion() {
        return $this->numero1 + $this->numero2;
    }
}
```

```
$mi_suma = new operacion(8, 4);
echo $mi_suma->operacion();
```


Pero reescribiendo la clase, podemos conseguir que reste en vez de sumar. no es necesario declarar el método constructor en la clase hija, pues lo hereda de la clase padre.

```
class resta extends operacion
{
    public function operacion() {
        return $this->numero1 - $this->numero2;
    }
}
```

este objeto ahora daría 4:

```
$mi_resta = new resta(8, 4);
echo $mi_resta->operacion();
```

Creación de un Objeto

Para crear objetos o instancias de una clase usamos la palabra reservada **new** y el nombre de la clase. El resultado de ejecutar esta instrucción es que se reserva un espacio en memoria capaz de albergar a nuestro objeto y se devuelve una referencia a dicho espacio.

```
1 <?php
2
3 $john = new Person("John", "Doe", "1980-12-01");
```

Creación de una GUI en PHP

El código es interpretado por un servidor web con un modulo procesador de PHP, el mas común llamado Apache, en un local host



The screenshot shows a web browser window with the address bar displaying 'localhost/cursoPHP/04.PDOySQL/editado/'. The page has a black navigation bar with links: 'Registro', 'Ingreso', 'Usuarios', and 'Salir'. Below the navigation bar, the title 'REGISTRO DE USUARIO' is centered. The registration form consists of three text input fields labeled 'Usuario', 'Contraseña', and 'Email', followed by a 'Enviar' button.

Registro	Ingreso	Usuarios	Salir
REGISTRO DE USUARIO			
<input type="text" value="Usuario"/>			
<input type="text" value="Contraseña"/>			
<input type="text" value="Email"/>			
<input type="button" value="Enviar"/>			

FIN