



Introducción PHP



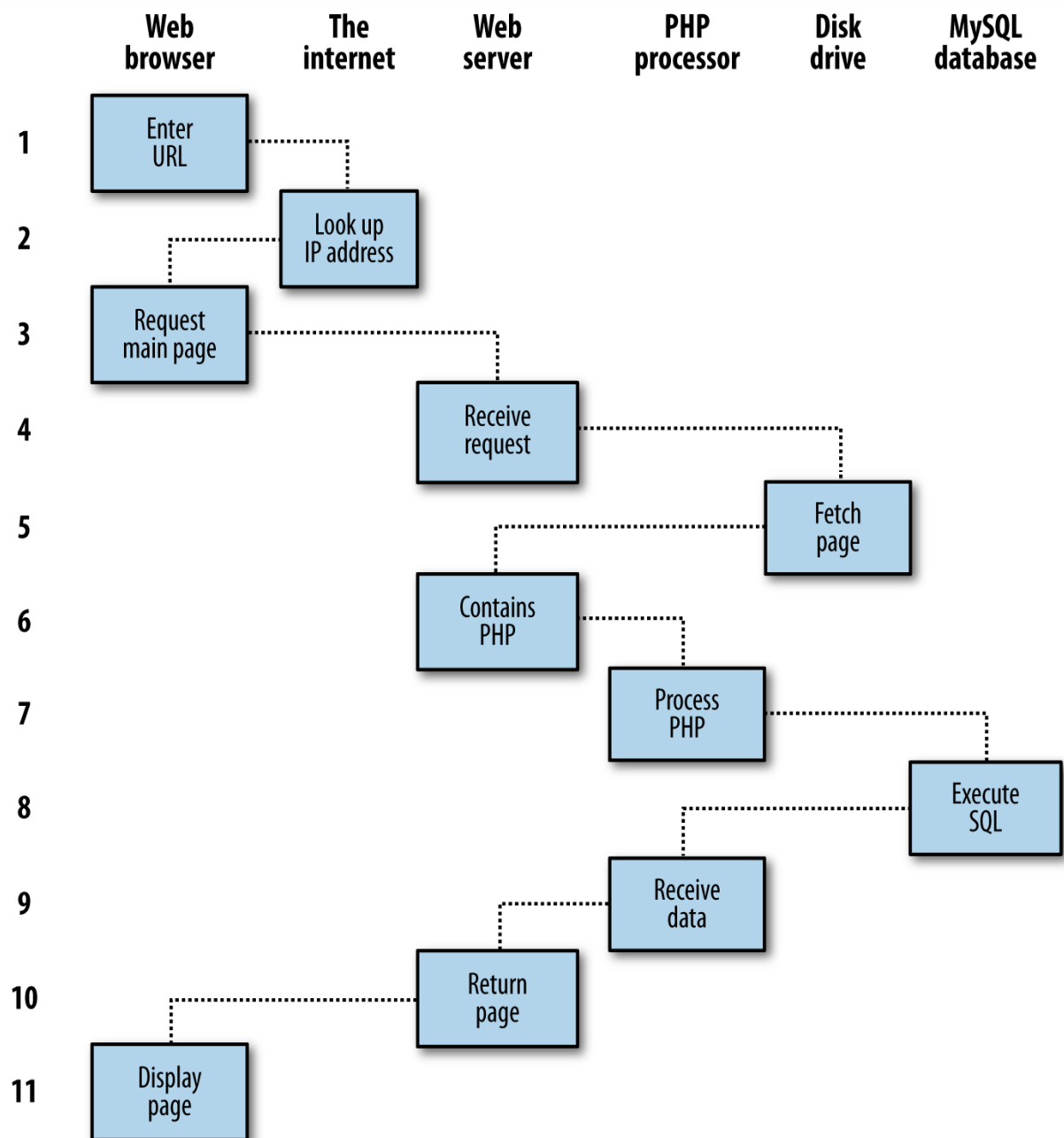
Lenguaje PHP

- Lenguaje de código abierto.
- Pensado para desarrollo web.
- Comúnmente utilizado como un módulo de Apache server.
- Puede usarse mediante línea de comandos como python, pearl, etc.
- Puede ser incrustado en HTML



Lenguaje PHP

- **Scripts del lado del servidor.**
- **Scripts desde la línea de comandos.**
- **Aplicaciones de escritorio.**





Lenguaje PHP

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo ("¡Hola, soy un script de PHP!");
    ?>

  </body>
</html>
```



Variables

```
$p = 3.14;
```

```
$s = "hola mundo";
```

```
$a = 5;
```

```
echo "$a_abc<br/>"; //there is no variable $a_abc
```

```
echo "{$a}_abc<br/>"; // OK
```

```
printf("hola %d mundo", $a);
```

```
$s1 = "hola";
```

```
$s2 = "mundo";
```

```
echo($s1.$s2." en PHP");
```

```
echo("$s1$s2 en PHP");
```

```
echo("{ $s1 } { $s2 } en PHP");
```



Variables

```
$alive = false; // $alive is false.  
$alive = true; // $alive is true.  
$alive = 1; // $alive is true.  
$alive = -1; // $alive is true, because -1 is nonzero.  
$alive = 5; // $alive is true.  
$alive = 0; // $alive is false.  
$alive = 'a'; // $alive is true.  
$alive = '1'; // $alive is true.  
$alive = '0'; // $alive is false.
```



Arrays

```
$state[0] = "Lanus";  
$state[1] = "Quilmes";  
$state[2] = "Avellaneda";  
$state[49] = "CABA";
```

```
print_r($state);
```

Array ([0] => Lanus [1] => Quilmes [2] => Avellaneda [49] => CABA)



Arrays

```
$state = array() ;
```

```
$state[] = "Lanus";
```

```
$state[] = "Quilmes";
```

```
$state[] = "Avellaneda";
```

```
$state[] = "CABA";
```

```
print_r($state);
```

```
Array ( [0] => Lanus [1] => Quilmes [2] => Avellaneda [3] => CABA )
```

Iteración:

```
foreach($state as $val)
```

```
{
```

```
    echo($val);
```

```
}
```



Diccionarios

```
$state["clave1"] = "valor1";  
$state["clave2"] = "valor2";  
$state["clave3"] = "valor3";  
$state["clave4"] = "valor4";
```

```
print_r($state);
```

```
Array ( [clave1] => valor1 [clave2] => valor2 [clave3] => valor3  
        [clave4] => valor4 )
```



Diccionarios

```
$state = array("clave1"=>"valor1", "clave2"=>"valor2");
```

```
Array ( [clave1] => valor1 [clave2] => valor2 )
```

Iteración:

```
foreach($state as $k =>$val)  
{  
    echo ($k." ".$val);  
}
```



Funciones

```
function sumar($arg1,$arg2) {  
    return $arg1 + $arg2;  
}
```

```
$r = sumar(5,6);
```

```
echo ("r:$r");
```



Clases

```
class Employee
{
    private $name;
    private $title;

    public function getName() {
        return $this->name;
    }
    public function setName($name) {
        $this->name = $name;
    }
    public function sayHello() {
        echo ("Hi, my name is {$this->getName()}.");
    }
};

$e = new Employee();
$e->setName("Ernesto");
$e->sayHello();
```



Clases

```
class Book
{
    private $title;
    private $isbn;
    private $copies;

    function __construct($isbn)
    {
        $this->setIsbn($isbn);
    }
    public function setIsbn($isbn)
    {
        $this->isbn = $isbn;
    }
}
```

```
$book = new Book("0615303889");
```



Herencia

```
class Employee {  
    protected $name;  
    function __construct($name) {  
        $this->name=$name;  
    }  
}  
  
class Manager extends Employee {  
    function __construct($name) {  
        parent::__construct($name);  
    }  
    public function greeting(){  
        echo("Hi, I am ".$this->name);  
    }  
}  
  
$m = new Manager("Ernesto");  
$m->greeting();
```



```
class Visitor
{
    private static $visitors = 0;

    function __construct()
    {
        self::$visitors++;
    }
    static function getVisitors()
    {
        return self::$visitors;
    }
}
```

static

```
$visits = new Visitor();
echo Visitor::getVisitors()."<br />"; //1
```

```
$visits2 = new Visitor();
echo Visitor::getVisitors()."<br />"; //2
```




Inclusión de archivos

- `include()`
- `include_once()`
- `require()` : Error si no existe el archivo
- `require_once()`



Strings

- Función **explode()**

```
$str = "hola,mundo";
```

```
$parts = explode(",",$str);
```

```
print_r($parts);
```

```
// Array ( [0] => hola [1] => mundo )
```



Strings

- Función trim()

```
$str = " hola mundo ";
```

```
$str = trim($str);
```

```
echo($str); // "hola mundo"
```



Strings

- Función `json_encode()`

```
$data = array("name"=>"Juan", "age"=>56, "logged"=>true);  
  
$jsonStr = json_encode($data);  
  
echo($jsonStr); // {"name":"Juan","age":56,"logged":true}
```



Strings

```
class User implements JsonSerializerizable {  
    ...  
  
    public function jsonSerialize() {  
        return array("name"=>$this->name, "age"=>$this->age);  
    }  
}  
  
$u = new User("Juan", 56);  
  
$jsonStr = json_encode($u);  
  
echo($jsonStr); // {"name":"Juan","age":56}
```



Strings

- Función `json_decode()`

```
$jsonStr = '{"name":"Juan","age":56,"logged":true}';
```

```
$data = json_decode($jsonStr);
```

```
print_r($data);
```

```
//Object ( [name] => Juan [age] => 56 [logged] => 1 )
```



Excepciones

```
try {  
    // run your code here  
  
}  
catch (Exception $e) {  
    //code to handle the exception  
  
}  
finally {  
    //optional code that always runs  
  
}
```



Excepciones

```
function dividirSeguro($a,$b)
{
    if ($b==0)
        throw new Exception("Error Divisor cero");
    else
        return $a/$b;
}

try {
    dividirSeguro(5,0);
}
catch (Exception $e) {
    echo ($e->getMessage());
}
```




Bibliografía

- <https://www.php.net/manual/es/intro-whatism.php>
- Robin Nixon. Learning PHP, MySQL & Javascript. O'Reilly Media. 2018.
- Frank M. Kromann, Beginning PHP and MySQL. Apress. 2018.