Características básicas del lenguaje PHP

Nuestro primer código PHP

- El código php siempre va entre los simbolos <?php y ?>.
- Las instrucciones php terminan siempre con ;
- Para generar código html desde php podemos utilizar el método echo pasándole el texto del código que queremos generar.

Variables en PHP

- Los nombres de las variables siempre comienzan por $
- Después del $ los nombres de las variables deben ir seguidos por **una letra** o **el carácter** _ y pueden contener también números.
- No es necesario declarar una variable ni especificarle un tipo (entero, cadena,…) concreto, se decide en función del contexto en que se emplee.

```
// Al asignarle el valor 7, la variable es de tipo "entero"
$mi_variable = 7;
// Si le cambiamos el contenido
$mi_variable = "siete";
// La variable puede cambiar de tipo
// En este caso pasa a ser de tipo "cadena"
```

Tipos de datos en PHP

- **booleano** (boolean). Sus posibles valores son true y false. Además, cualquier número entero se considera como true, salvo el 0 que es false.
- **entero** (integer). Cualquier número sin decimales. Se pueden representar en formato decimal, octal (comenzando por un 0), o hexadecimal (comenzando por 0x).
- **real** (float). Cualquier número con decimales. Se pueden representar también en notación científica.
- **cadena** (string). Conjuntos de caracteres delimitados por comillas simples o dobles.
- **null**. Es un tipo de datos especial, que se usa para indicar que la variable no tiene valor. (http://php.net/manual/es/language.types.null.php)

Cadenas de texto

- Podemos usar tanto comillas simples como comillas dobles.
- Para concatenar cadenas utilizaremos el operador punto (.)
- Podemos introducir una **variable dentro de un texto** siempre y cuando usemos las **comillas dobles** para delimitar el texto. Esto hará que el contenido de la variable se expanda y se concatene con el texto existente en la cadena.

  ```
  echo "<p>Módulo: $modulo</p>"
  ```

- A veces, es necesario rodearla entre llaves

  ```
  echo "<p>Módulo: {$modulo}DAW</p>“
  ```

  - Si no pusiéramos las llaves el intérprete buscaría una variable que se llame $moduloDAW

Funciones de cadenas

Puedes consultarlas en:

http://es.php.net/manual/es/ref.strings.php

Variables superglobales

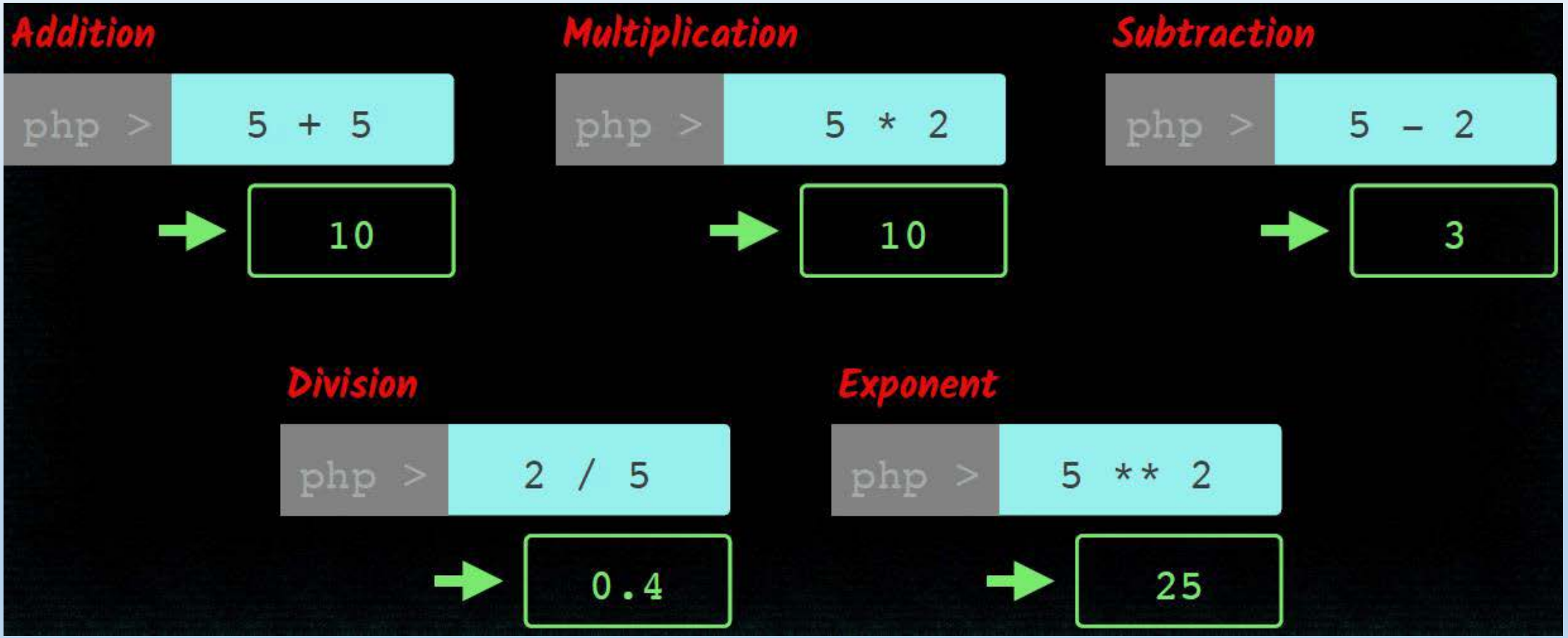
- Variables internas predefinidas de PHP que pueden usarse desde cualquier ámbito.
- Son arrays asociativos que contienen un conjunto de valores:
  - **$_SERVER**. Contiene información sobre el entorno del servidor web y de ejecución.
  - **$_GET**, **$_POST** y **$_COOKIE** contienen las variables que se han pasado al script actual utilizando, respectivamente, los métodos GET (parámetros en la URL), HTTP POST y Cookies HTTP.
  - **$_REQUEST** junta en uno solo el contenido de los tres arrays anteriores, $_GET, $_POST y $_COOKIE.
  - **$_ENV** contiene las variables que se puedan haber pasado a PHP desde el entorno en que se ejecuta.
  - **$_FILES** contiene los ficheros que se puedan haber subido al servidor utilizando el método POST.
  - **$_SESSION** contiene las variables de sesión disponibles para el guion actual.

http://es.php.net/manual/es/language.variables.superglobals.php

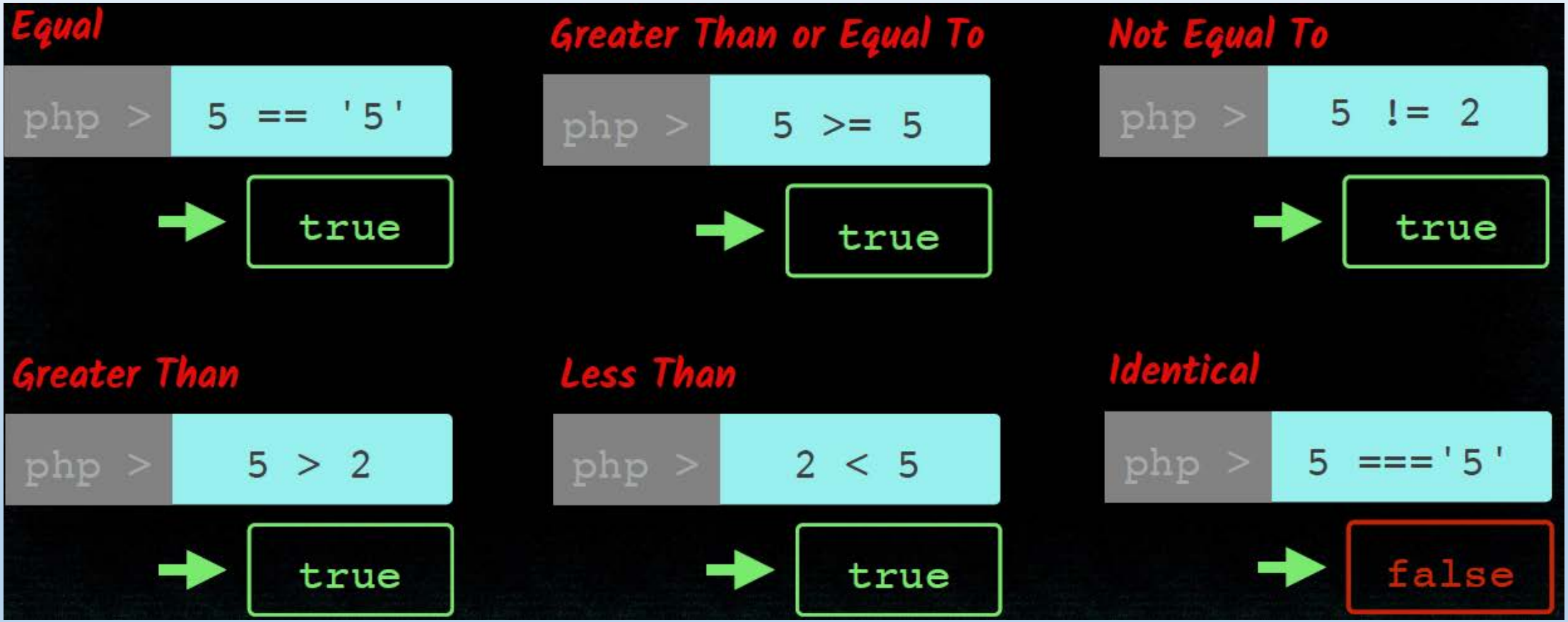
Funciones isset y unset

- **isset** determina si una variable o variables están definidas y no son **NULL**.
- **unset** destruye las variables especificadas.
- Si una variable ha sido eliminada con unset(), isset() devolverá FALSE.
- isset devolverá false si la variable ha sido definida como NULL.
- Si una variable contiene una cadena vacía isset devolverá true.
- Si son pasados varios parámetros, isset() devolverá true únicamente si todos los parámetros están definidos.
- isset se utiliza mucho para comprobar si se ha recibido un parámetro por GET.

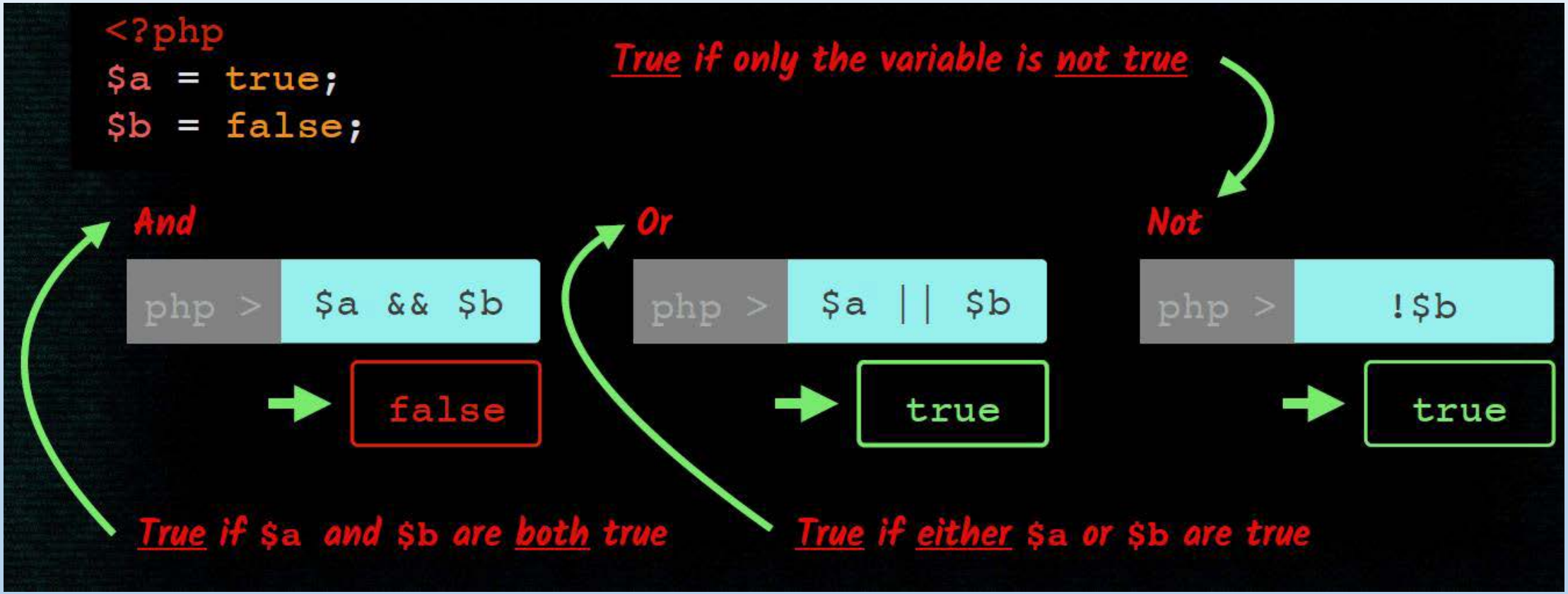
Operadores Aritméticos



Operadores de comparación



Operadores lógicos



Estructuras condicionales IF y SWITCH

IF:

```php
<?php
    $var = 3;
    if($var < 1) echo "Es un cero";
    elseif($var == 1){ echo "Es un uno"; }
    elseif($var == 2)
        echo "Es un dos";
    elseif($var == 3){
        echo "Es un tres";
    }else{
        echo "No es uno, ni dos, ni tres";
    }
?>
```

SWITCH:

```php
<?php
    $var = 3;
    switch($var){
        case 1:
            echo "Es un 1";
            break;
        case 2:
            echo "Es un 2";
            break;
        case 3:
            echo "Es un 3";
            break;
        default:
            echo "No es ni 1, ni 2, ni 3";
    }
?>
```

Estructuras de repetición FOR, WHILE y DO...WHILE

FOR:

```php
<?php
    for($i = 0; $i < 5; $i++){
        echo "$i <br>";
    }
?>
```

WHILE:

```php
<?php
    $i=0;
    while($i < 5){
        echo "$i <br>";
        $i = $i +1;
    }
?>
```

DO...WHILE:

```php
<?php
    $i=0;
    do{
        echo "$i <br>";
        $i = $i +1;
    } while($i < 5);
?>
```

Instrucción BREAK

```php
<?php
    echo "Primer for anidado: <br>";
    for($i=0; $i<3; $i++){
        for($j=0; $j<3; $j++){
            echo "i:$i j:$j <br>";
            if ($j == 1) {
                break;
            }
        }
    }
    echo "Segundo for anidado: <br>";
    for($i=0; $i<3; $i++){
        for($j=0; $j<3; $j++){
            echo "i:$i j:$j <br>";
            if ($j == 1) {
                break 2;
            }
        }
    }
?>
```

Salida:

- Primer for anidado:
  - i: 0 j:0
  - i: 0 j:1
  - i: 1 j:0
  - i: 1 j:1
  - i: 2 j:0
  - i: 2 j:1
- Segundo for anidado:
  - i: 0 j:0
  - i: 0 j:1

Instrucción CONTINUE

```php
<?php
    for($i=0; $i<5; $i++){
        if ($i == 3){
            continue;
        }
        echo "$i <br>";
    }
?>
```

Salida:

0

1

2

4

Nuevo operador de fusión de null ??

```php
<?php

    if(isset($_GET['nombre']))
        $nombre = $_GET['nombre'];
    else
        $nombre = "Anónimo";
    echo $saludo . $nombre;

?>
```

==

```php
<?php

    $nombre = $_GET['nombre'] ?? "Anónimo";
    echo $saludo . $nombre;

?>
```

Arrays

- Un **array** es un tipo de datos que nos permite almacenar varios valores.
- Para acceder a un valor utilizaremos una **clave**.
- Las claves pueden ser **números** o **textos** (arrays asociativos).
- Si no indicamos ninguna clave, a cada elemento se le asociará una clave numérica correlativa.

Usando la función array - $nombres = **array**();

Usando atajo disponible desde versión 5.4 de PHP - $nombres = [];

Crear array con valores

- Si queremos asignar valores al array en el momento de crearlo:

```
$nombres = array('Sandra', 'Pedro', 'Andrea');

$nombres = ['Sandra', 'Pedro', 'Andrea'];
```

Añadiendo elementos al array

- Podemos añadir elementos al array utilizando corchetes vacíos:

```
$nombres[] = 'Raul';
$nombres[] = 'Marta';
```

- La clave de estos elementos será el siguiente índice numérico disponible.

Mostrar el contenido de un array

- Si intentamos mostrar el contenido de un array con **echo** obtendremos el texto “Array”.
- echo no muestra los datos que hay dentro del array.
- En lugar de esto podemos utilizar la función **print_r**:

  print_r($nombres);

- Si necesitamos darle formato a los contenidos del array, tendremos que recorrerlo con un bucle e ir mostrando elemento a elemento.

Acceder a los elementos del array

- Accederemos a los elementos del array indicando la clave del elemento entre corchetes:

```
echo $nombres[2];
```

- Podemos modificar su valor:

```
$nombres[0] = 'Sara';
```

Arrays asociativos

- Si el array contiene datos diversos y/o nos interesa acceder a ellos con claves más específicas que un simple índice numérico podemos utilizar arrays asociativos:

```php
$alumno = array(
    'nombre' => 'Sara',
    'apellido' => 'García',
    'edad' => 22
);
$alumno = [
    'nombre' => 'Sara',
    'apellido' => 'García',
    'edad' => 22
];
$alumno['nombre'] = 'Verónica';
echo $alumno['nombre'];
```

Arrays multidimensionales

- Imagina que queremos tener un array con nombres de deportes.
- Queremos dividirlos en invierno y verano.
- Podemos crear un array $deportes que contenga dos elementos que a su vez también serán arrays.

```
$deportes = [
    'invierno' => ['esquí de fondo', 'hockey sobre hielo'],
   'verano' => [ 'natación', 'voley playa']
];
```

- Para acceder al primer deporte de invierno:

```
echo $deportes['invierno'][0];
```

Arrays Multidimensionales asociativos

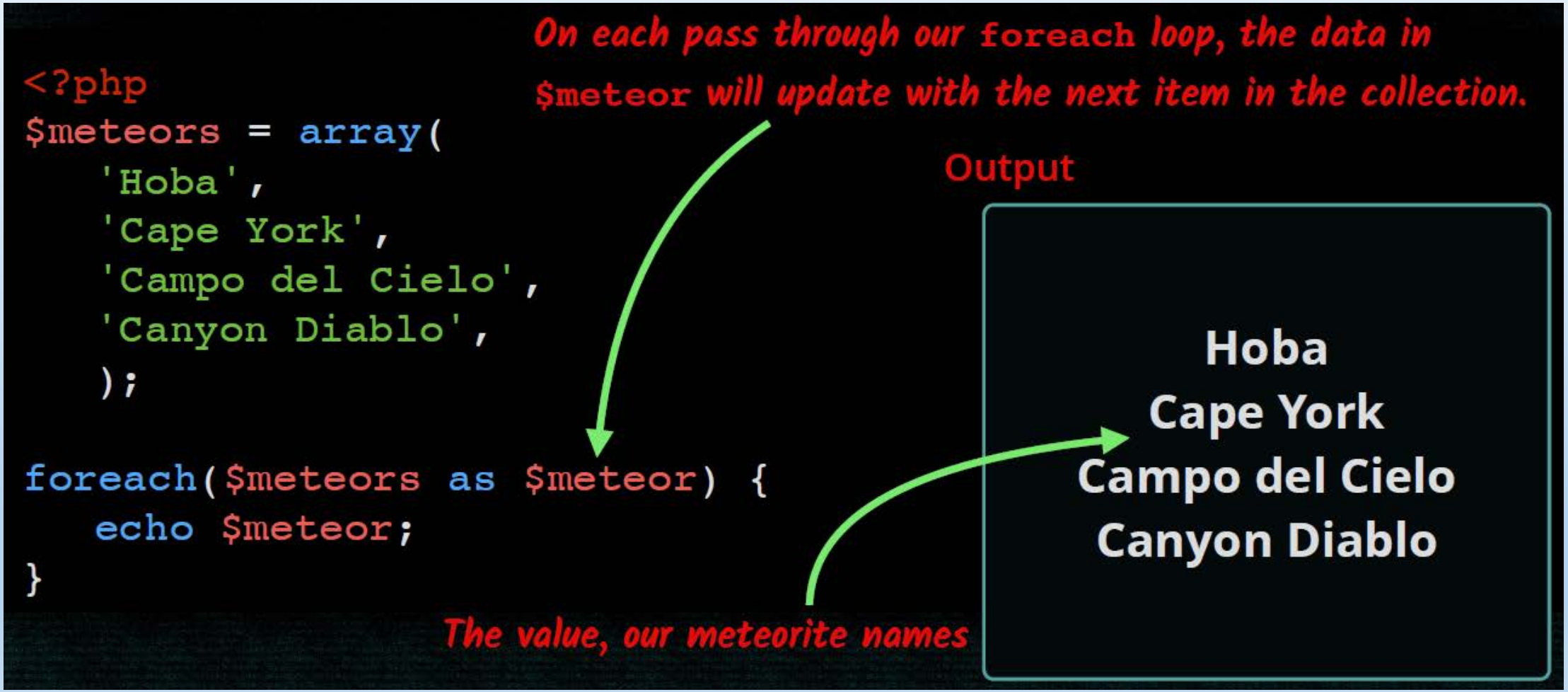
- La segunda dimensión del array también puede ser un array asociativo:

```php
$alumnos = [
    [
        'nombre' =>'Sara',
        'edad' =>25
    ],
    [
        'nombre' =>'Pedro',
        'edad' =>23
    ],
];
```

- Mostraremos el nombre del primer alumno:

```php
echo $alumnos[0]['nombre'];
```

Recorrer arrays



Recorrer arrays asociativos

```php
<?php
$meteors = array(
    'Hoba' => 600000000,
    'Cape York' => 58200000,
    'Campo del Cielo' => 50000000,
    'Canyon Diablo' => 30000000,
    );

foreach($meteors as $meteor) {
    echo $meteor;
}
```

The value is our meteorite weight!

Output

600000000
58200000
50000000
30000000

Recorrer claves y valores

```php
<?php
$meteors = array(
    'Hoba' => 600000000,
    'Cape York' => 58200000,
    'Campo del Cielo' => 50000000,
    'Canyon Diablo' => 30000000,
    );

foreach($meteors as $name => $weight){
    echo "$name weighs $weight grams.";
}
```

Output

```
Hoba weighs 600000000 grams.
...
Canyon Diablo weighs 30000000 grams.
```

Funciones de arrays

Puedes consultarlas en:

https://www.php.net/manual/es/ref.array.php