

Ámbito de variables

El ámbito de una variable es el contexto dentro del que la variable está definida, es decir el grado de disponibilidad de esta a lo largo de nuestro programa.

Existen 4 tipos:

- Variables locales.
- Variables estáticas.
- Variables globales.
- Variables superglobales.

VARIABLES LOCALES

Las de ámbito local son las que definimos dentro de una función o una clase y podrán ser utilizadas solo en ese ámbito, esto quiere decir que fuera de ellas no existirán ya que se destruyen al finalizar de ejecutarse el código. Por consecuente, si tenemos una variable de igual nombre fuera de la función, estas dos serán dos variables completamente distintas.



```
1 $varLocal = "Hola<br>";
2 function miFuncion()
3 {
4     $varLocal = "Hola desde función <br>";
5 }
6 miFuncion();
7 echo $varLocal;
```

Fig. 1

Si observamos la Fig.1 vemos que las variables de la línea 1 y 4 son llamadas bajo el mismo nombre, sin embargo son de ámbitos distintos y por lo tanto, variables distintas.

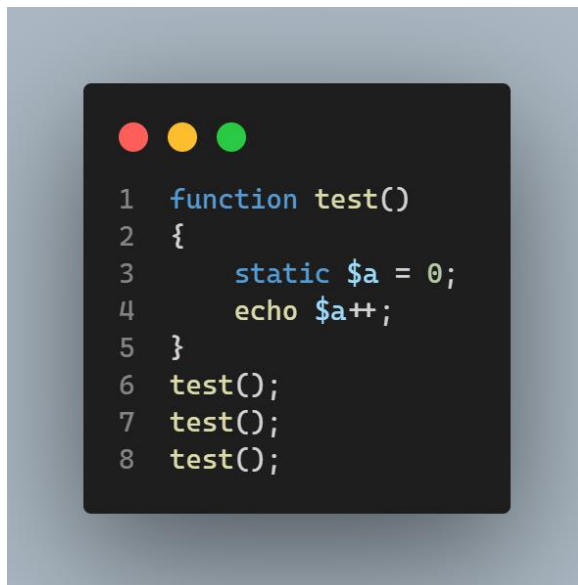
Las variables locales son destruidas una vez que la función termina de ser ejecutada, cuando volvemos a llamarla, la variable vuelve a definirse con el primer valor

que se le asigne.

VARIABLES ESTÁTICAS

Este tipo de variable se definen dentro de una función anteponiendo a la variable la palabra reservada **static**.

Las variables estáticas son similares a las locales en cuanto al funcionamiento de los ámbitos. La diferencia con ellas, es que solo se definen con la primera llamada a la función y luego mantienen sus valores aun después de haber terminado la ejecución de la función que la contiene. Esto quiere decir que el último valor que tome la variable, será el mismo cuando volvamos a llamar a la función.



```

1  function test()
2  {
3      static $a = 0;
4      echo $a++;
5  }
6  test();
7  test();
8  test();

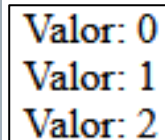
```

Fig. 2

En la fig. 2 tenemos un claro ejemplo de una función con una variable estática.

Como podemos observar, la línea 3 declara una variable del tipo static inicializada en 0, luego escribirá el valor de la variable y la incrementará en 1.

El resultado, sería tal cual lo observamos en la figura 3. La variable \$a mantiene su valor aún finalizada la ejecución de la función, es por ello que cuando la volvemos a invocar, el valor continúa incrementándose sin



```

Valor: 0
Valor: 1
Valor: 2

```

Fig. 3

inicializarse en 0.

VARIABLES GLOBALES

Las variables del ámbito global pueden ser utilizadas en cualquier lugar de nuestro código.

La condición para que una variable sea de este tipo es que no esté dentro de una función o una clase.

Es decir que, todas las variables que hemos realizado en las prácticas hasta el día de hoy, son del ámbito global.

Las variables globales, son inaccesibles para las funciones o clases, a menos que nosotros se lo indiquemos, caso contrario la función creará su variable para trabajarla en forma local mientras dure su ejecución.



```

1  $varGlobal = "Hola <br>";
2  echo $varGlobal;
3  function miFuncion2()
4  {
5      global $varGlobal;
6      $varGlobal = "Hola desde función <br>";
7  }
8  miFuncion2();
9  echo $varGlobal;

```

Fig. 4

En la fig. 4 podemos observar que en la primer línea estamos declarando una variable de tipo global para luego ser utilizada dentro de la función.

Tal como mencionamos en párrafos anteriores, debemos indicarle a la función que vamos a hacer uno de una variable del ámbito

global y para ello, debemos anteponerle la palabra reservada `global` (línea 5). De esta manera, al asignarle el valor en la línea 6, lo hará utilizando la variable declarada en la línea 1.



```
1 $varGlobal = "Hola <br>";
2 echo $varGlobal;
3 function miFuncion2()
4 {
5     $GLOBALS['varGlobal'] = "Hola desde función <br>";
6 }
7 miFuncion2();
8 echo $varGlobal;
```

Fig. 5

Otra sintaxis que se puede encontrar sobre el uso de estas variables (aunque no tan común) es como se visualiza en la fig. 5, línea 5.

El funcionamiento es idéntico al ejemplo anterior solo que modifica la forma de

declaración.

VARIABLES SUPERGLOBALES

Este tipo de variables tienen mayor disponibilidad que todas las otras. A diferencia de las globales estas variables se las puede invocar desde cualquier parte de nuestra programación, incluso por fuera de nuestro archivo en el que estemos programando.

Existen varios tipos de variables superglobales:

- Información de formularios:
 - `$_POST`
 - `$_GET`
 - `$_FILES`
 - `$_REQUEST`
- Información del servidor:
 - `$_SERVER`
- Sesión y cookies:
 - `$_SESSION`
 - `$_COOKIE`
- Variables de entorno:
 - `$_ENV`