**Los siguientes valores se consideran FALSE en php:**

* **El boolean FALSE**
* **El integer 0 (cero)**
* **El float 0.0 (cero)**
* **El valor string vacío y el string "0"**
* **Un array con cero elementos**
* **El tipo especial NULL (incluidas variables no establecidas)**

**Cualquier otro valor se considera como TRUE**

**Una variable es considerada NULL en php:**

* **Si se le ha asignado la constante NULL.**
* **Si no se le ha asignado un valor todavía.**
* **Si se ha destruido con unset().**

**No hay más que un valor de tipo null, y es la constante NULL insensible a mayúsculas y minúsculas. El valor NULL representa una variable sin valor. Se puede asignar explícitamente null a una variable $variableSinValor = null;**

**Los siguientes valores consideran vacíos (empty) en php:**

* **"" (una cadena vacía)**
* **0 (0 como un entero)**
* **0.0 (0 como un real)**
* **"0" (0 como una cadena)**
* **NULL**
* **FALSE**
* **array() (un array vacío)**
* **$var; (una variable declarada, pero sin un valor)**

empty()**no genera una advertencia si la variable no existe.**

**Determinar si una variable existe (isset) en php:**

* **Si una variable existe si está definida y tiene un valor distinto de NULL.**
* **isset($\_POST) o isset($\_GET) devolverá siempre true porque $\_POST y $\_GET están definidos como un array en el ámbito global de PHP, aunque esté vacío.**
* **Para verificar si un array está vacío, es mejor usar la función empty()**
* **Cuando se aplica a un elemento de un array para verificar si existe y está definido, isset() devuelve true si el elemento existe, incluso si su valor es null o un array vacío.**

**Hacer un if ($x) mientras que el valor de $x no esté definido, generará un error de nivel E\_NOTICE. Al igual que otras muchas operaciones.**

**Para evitar problemas debemos hacer uso de funciones empty() o isset o is\_null().**

**¿Cuándo usar isset(), empty() o is\_null()?**

**Hay que usar isset() cuando necesitemos averiguar si una variable está definida en el script que estamos ejecutando, es decir, si existe.**

**La función empty() hay que usarla cuando queramos averiguar si una variable está vacía o no, es decir, si tiene contenido o no.**

**La función is\_null() se usa cuando queremos saber si una variable es igual a NULL, normalmente, para saber si se ha destruido.**

**Averiguar tipo de dato de una variable**

**is\_string(): el valor es de tipo cadena.**

**is\_int(): el valor es un número entero. Si el número entero está como cadena de texto (entre comillas) no se considera como un número.**

**is\_float(): el valor es un número con decimales. Si el número decimal está como cadena de texto (entre comillas) no se considera como un número.**

**is\_numeric(): el valor es un número o una cadena numérica. Al contrario de lo que sucede con is\_int() e is\_float(), si el número se encuentra como cadena de texto (entre comillas) sí se considera que es numérico.**

**is\_array(): retorna true si la variable es un array. En caso contrario, false.**

**is\_object(): retorna true si la variable es un objeto.**

**method\_exists(): retorna true si un objeto contiene un método con el nombre especificado.**

**is\_resource(): la variable contiene una referencia a un recurso (un archivo abierto, conexión con una base de datos, etc.)**