**1º Prueba Hora Libre Configuración**

**1. ¿Cuál es la diferencia entre las funciones asort() y ksort()?**

a) Ambas ordenan de forma ascendente arrays exclusivamente indexados. *Ksort* lo hace de acuerdo a la clave de cada elemento y *asort* lo hace de acuerdo al valor de cada elemento.

b) Ninguna es correcta.

**c) *asort* ordena ascendetemente arrays asociativos de acuerdo al valor de cada elemento y *ksort* ordena ascendentemente arrays asociativos de acuerdo a la clave de cada elemento.**

d) *asort* ordena ascendetemente arrays asociativos de acuerdo al valor de cada elemento y *ksort* ordena ascendentemente arrays indexados de acuerdo a la clave de cada elemento.

**2. ¿Qué es cierto en relación a las sentencias *include* y *require*?**

**a) c) y d) son correctas.**

b) *required* y *include* idénticos: producen errores leves y permiten que script continúe ejecutándose.

c) Si el archivo incluido con *required* contiene errores, se produce un error fatal y el script se detiene.

d) Si el archivo incluido con *include* produce un aviso o warming, y el script continúa ejecutándose.

**3. PHP es un lenguaje de programación del que podemos decir que:**

a) Se procesa en el lado cliente y es débilmente tipado. Por eso puede ser combinado con HTML.

b) En general, es la alternativa más extendida frente a otros lenguajes de servidor (ASP, .NET, JSP, ETC.)

**c) b) y d) son correctas.**

d) En función nde su procesado en el lado del servidor genera código HTML que es enviado al navegador (en el lado cliente).

**4. ¿Para qué se puede utilizar la función *prg\_match()*?**

a) Llevar a cabo validaciones de datos de entrada(p.ej en un formulario HTML), especialmente cuando hay restricciones en el formato de esos datos.

**b) a) y d) son correctas.**

c) Para subir al servidor preguntas definidas por el usuario

d) Para saber si una cadena contiene coincidencias con un patrón expresado mediante expresiones regulares (RegExp).

**5. Se requiere comprobar si una variable (por ejemplo, llamada $string), esta vacía. Además se quiere comprobar cuál es su longitud en caracteres. ¿Qué métodos o funciones se deberán utilizar?**

**a) empty($string) y strlen($string)**

b) size($) y chars($string)

c) is\_empty($string) y strlen($string)

d) empty($string) y strsize($string)

**6. La función PHP *var\_dump()...***

a) Es útil para hacer casting de tipos de datos constantes de variables.

**b) Devuelva el tipo de dato y valor de la variable.**

c) No existe.

d) Se usa con sintaxis $variable.var\_dump();

**7.Al procesar una página PHP, cuando aparecen mensajes de error del intérprete de PHP...**

**a) El script PHP se sigue procesando, siempre que no se trate de un Parse Error (error de sintaxis del código PHP) ni un Fatal Error.**

b) El script de PHP se detiene con cualquier tipo de error.

c) El script de PHP sigue procesando incodicionalmente.

d) Ninguna es correcta.

**8. ¿Cómo hacemos *Casting* de una variable en PHP?**

**a) Ninguna es correcta**

b) parseInteger($x)

c) <int>$x

d) (int)x

**9. Para recoger el valor de un elemento de un formulario HTML en PHP (por ejemplo de password), lo más adecuado sería usar:**

a) $\_GET[“pwd”]

b) $\_POST(“pwd”)

c) $\_GETPASSWORD[“pwd”]

**d) $\_POST[“pwd”]**

**10. ¿Para que sirve el método *isset()* en PHP?**

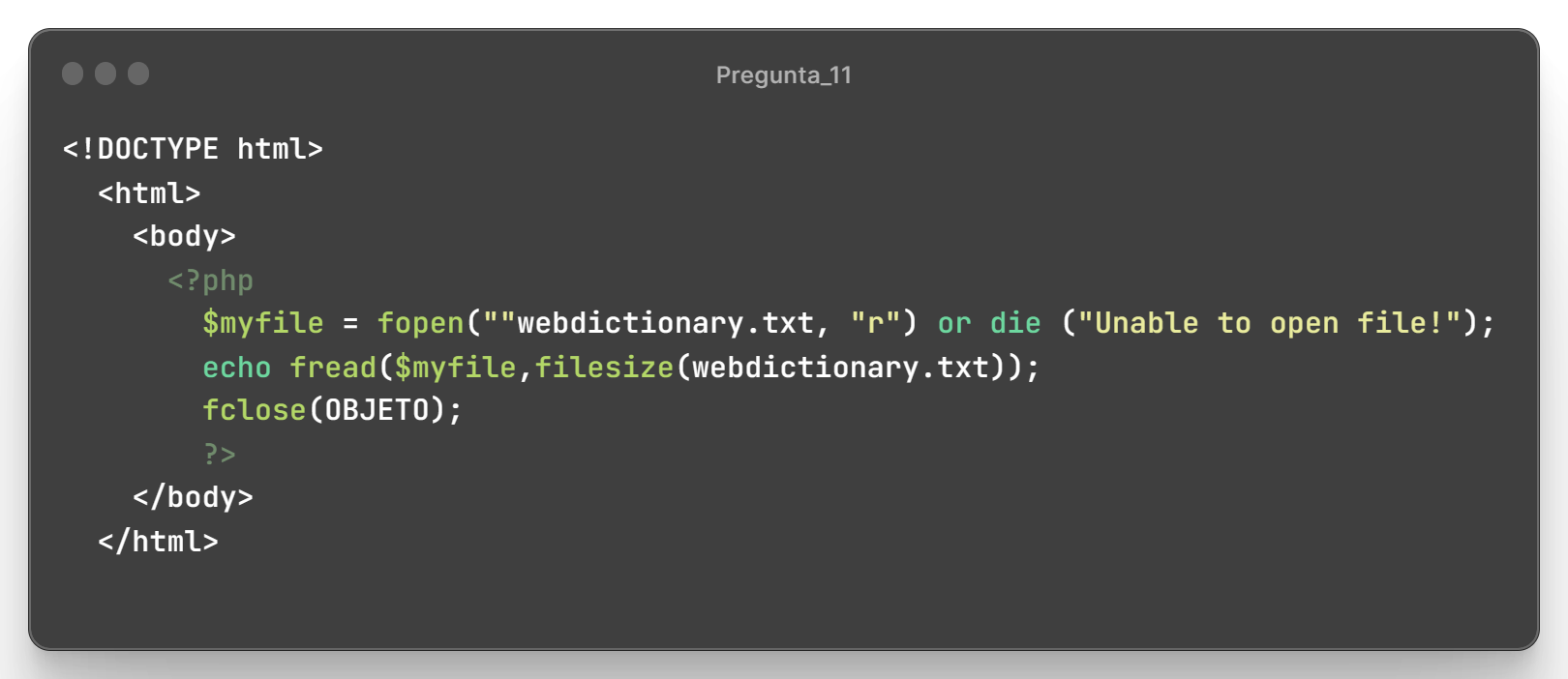
a) Determina si una variable tiene un valor asignado, independientemente de que sea null.

b) Permite saber si un array esta vacío.

**c) Determina si una variable está definida (e.d, tiene un valor asignado) y no es null.**

d) Ninguna de las anteriores es correcta.

**11. Para el siguiente ejemplo de código fuente sea correcto, OBJETO debe sustituirse por:**



a) “webdictionary.txt”

b) No lleva nada (sin parámetro/variable de entrada).

c) Ninguna es correcta.

**d) $myfile**

**12. En PHP, ¿Cuál de las siguientes afirmaciones describe correctamente la diferencia entre ‘==’ ‘===’ al comparar variables fuera de una lista (array)?**

a) ‘==’ compara los pares clave/valor en el mismo order y tipo de variable, mientras que ‘===’ compara los pares clave/valor.

b) ‘==’ compara el valor y el tipo de la variable, mientras que ‘===’ compara solo el valor.

c) Ninguna es correcta.

**d) ‘==’ compara solo el valor, mientras que ‘===’ compara el valor y el tipo de la variable.**

**13. ¿Cuál es la diferencia entre “echo” y “print” en PHP?**

a) La unica diferencia entre “*echo*” y “*print*” es el nombre. Puedes elegir cualquiera de ellos, ya que no tienen efectos diferentes.

b) Ninguna es correcta.

**c) Ambos muestran datos, pero “*echo*” puede tomar múltiples parámetros, mientras que “*print*” solo puede tomar uno y siempre devuelve 1.**

d) “*print*” es simplemente una versión más antigua de “*echo*” en PHP. No hay ninguna ventaja en usar “*echo*” sobre “*print*” o viceversa; simplemente cuestión de estilo.

**14. La función predefinida de PHP array\_pop($variable)...**

a) Borra el último elemento del array $variable y no devuelve nada.

**b) Devuelve el último elemento del array $variable antes de borrarlo de dicho array.**

c) Simplemente devuelve último elemento del array $variable, sin modificar nada.

d) Devuelve el primer elemento del array $variable antes de borrarlo de dicho array.

**15. Si, estando dentros de una función local(...) de PHP, se desea aplicar la variable *unset(VARIABLE)* a una variable global ¿Cuál de las siguientes irá en el lugar *VARIABLE* para que *unset* consiga su propósito?**

**a) $GLOBALS, de modo que la variable en ámbito de la llamada local(...) es destruida.**

b) $\_ENV, de modo que la variable en el ámbito de la llamada a local(..) es destruida.

c) $\_SEVER, de modo que sólo la variable local es destruida. La variable de entorno de la llamada mantendrá el mismo valor anterior a la llamada de *unset*.

d) $\_GLOBALS, de modo que sólo la variable local es destruida. La variable en el ámbito de la llamada a *local(...)* mantendrá el mismo valor anterior a la llamada a *local(...).*