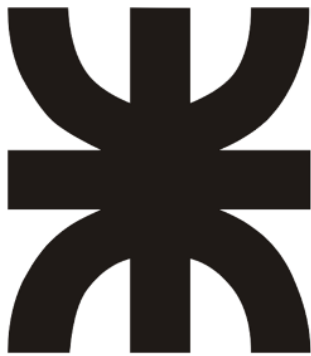


Entornos Gráficos

Práctica 4

DOCENTES

- DÍAZ DANIELA,
- JULIÁN BUTTI.



- Legajo: 44961. Cano Lautaro,
- Legajo: 44762. Gasparini Sofia,
- Legajo: 44800. Lorenzotti Guido

Año de cursado: 2021

Comisión 4E03

Grupo 01

Parte 1

1. Variables

- a:boolean
- b:string
- c:string
- d:integer
- f:integer
- g:integer

Operadores

- = unario
- += binario
- ++ unario
- * binario
- ?: ternario

Funciones y sus parámetros

- function doble(\$i) parámetro \$i

Estructuras de control

- if (is_int(\$d))
- if (is_string(\$a))

Salida por pantalla

- booleanstringstringinteger1xyzxyz184444

2. a) Si, todos estos códigos muestran una secuencia numérica del 1 al 10 inclusive.
- b) Si, al igual que en apartado a, todos los códigos muestran una secuencia numérica del 1 al 10 inclusive.
- c) Si, ambos códigos darán como resultado "i equals" y el valor que contiene la variable i (siempre y cuando sea 0, 1, o 2).

3. a) Se crea una tabla y por medio de php se realiza el diseño de la misma.
b) Se crea un formulario que posee un campo que al colocar un numero y al clickear el boton, arroja una salida de si la persona es o no mayor de edad (mayor o igual a 21 años o menor a 21 años)

4. La salida muestra al principio 2 alertas:

Warning: Undefined variable \$flor in C:\xampp\htdocs\practica4\ejercicio4.php on line 3

Warning: Undefined variable \$color in C:\xampp\htdocs\practica4\ejercicio4.php on line 3

Esto debido a que en la primera sentencia aún no se ha incluido el archivo datos y, por ende, las variables “flor” y “color” no están definidas en ese contexto. Lo único que llega a imprimirse de esto es el string “El ” (con un espacio al final por el `\n`) ya que no es una variable. Luego de incluir dicho archivo si puede imprimirse correctamente el valor que tienen las variables en el mismo. El resultado es: El clavel blanco.

De esta forma el resultado final de la salida es:

Warning: Undefined variable \$flor in C:\xampp\htdocs\practica4\ejercicio4.php on line 3

Warning: Undefined variable \$color in C:\xampp\htdocs\practica4\ejercicio4.php on line 3

El El clavel blanco

5. contador.php

\$archivo = "contador.dat";

Asigna a la variable archivo el fichero “contador.dat”, el cual es un archivo en el cual es posible guardar datos de tipo genérico, para que el programa que lo creó pueda usarlos posteriormente.

\$abrir = fopen(\$archivo, "r");

Se abre el archivo en modo “solo lectura”, en la variable abrir.

\$cont = fread(\$abrir, filesize(\$archivo));

Se lee el archivo en la variable cont, el cual es señalado en el primer parámetro. Con el segundo parámetro se señala la longitud de lectura deseada, en este caso, todo el archivo.

fclose(\$abrir);

Se cierra el archivo.

\$abrir = fopen(\$archivo, "w");

Se abre el archivo en modo “solo escritura”.

\$cont = \$cont + 1;

Se suma 1 al contenido de la variable cont.

\$guardar = fwrite(\$abrir, \$cont);

Se escribe, en el archivo señalado por el primer parámetro, el contenido del segundo.

fclose(\$abrir);

Se cierra el archivo.

echo "Cantidad de visitas:".\$cont."";

Se muestra la cantidad de visitas de la página mediante la variable cont, de este modo: "Cantidad de visitas: x" (utilizando la fuente y tamaño especificados).

visitas.php

<? include("contador.php")?>

Dentro de visitas.php, se incluye la página contador.php, para así mostrar el resultado.

Parte 2

1. Ambos códigos son equivalentes. En el primero se crea un array y se le agregan elementos en la misma sentencia. Los primeros 4 elementos se agregan por pares clave-valor. De esta manera el acceso al valor se realiza proporcionando la respectiva clave. El último elemento al prescindir de clave se le asigna automáticamente una de manera "indexada", con valor 0 en este caso debido a que es el primer elemento que se agrega de esta forma.

En el segundo conjunto de sentencias se crea exactamente el mismo array pero esta vez agregando los elementos en sentencias separadas, asignando valores al array en cada una de las claves dadas.

2. a) La salida es: **bar1**. En el código se crea una matriz, se le agregan 2 elementos y, posteriormente, se imprimen en pantalla ambos. El primer elemento se asocia a la clave "x" y su valor es el string "bar". El segundo se asocia a la clave o índice 12 y su valor es el true lógico. El string obviamente se visualiza como tal y el true lógico como 1.

b) La salida es **5942**. En este caso se crea una matriz multidimensional que contiene una única matriz simple. Se imprimen uno a uno los elementos de esta última.

c) En este caso no se visualiza nada como salida. En el código se crea una matriz, se le agregan 4 elementos de diversas formas, se elimina el elemento

guardado en el índice 5 y finalmente se destruye la variable que contiene la matriz.

3. a) Este código da como resultado, la hora exacta a la que ha ingresado a la página (incluyendo minutos y segundos), junto al día, mes y año.

Ej: **Has entrado en esta página a las 0 horas, con 42 minutos y 18 segundos, del 7/6/2021**

b) La función planteada, da como resultado la expresión de suma entre los valores de ambos parámetros, junto a su resultado, en este caso, como los valores de los parámetros son 5 y 6, el resultado es: **5+6=11**

4. La función “comprobar_nombre_usuario” recibe como parámetro un nombre de usuario, verifica si este es válido o no y devuelve true o false respectivamente. Para ser válido el nombre debe tener entre 3 y 20 caracteres. Además todos los caracteres tienen que estar dentro de los permitidos que son letras (mayúsculas o minúsculas), números y guiones (bajo y medio). La función verifica ambas condiciones y en caso de que no se cumplan devuelve false, notificando en pantalla que no es válido el nombre. De lo contrario, devuelve true y notifica que es válido.

Para poder utilizar la función basta con llamarla luego de definirla pasándole el parámetro necesario.

```
<?php
//Se define la función
function comprobar_nombre_usuario($nombre_usuario){
    //compruebo que el tamaño del string sea válido.
    if (strlen($nombre_usuario)<3 ||
strlen($nombre_usuario)>20){
        echo $nombre_usuario . " no es válido<br>";
        return false;
    }
    //compruebo que los caracteres sean los permitidos
    $permitidos="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_" ;
    for ($i=0; $i<strlen($nombre_usuario); $i++){
        if(strpos($permitidos,substr($nombre_usuario,$i,1))==false){
            echo $nombre_usuario . " no es válido<br>";
            return false;
        }
    }
    echo $nombre_usuario . " es válido<br>";
    return true;
}
```

```
}  
//Se llama a la función  
comprobar_nombre_usuario("user");  
?>
```

También podría guardarse en un archivo aparte e incluirse con `include(nombreakivo.php);`, llamándola luego de la misma forma.