Ejercicio 1:
En el siguiente código identificar:

- las variables y su tipo
- los operadores
- las funciones y sus parámetros
- las estructuras de control
- cuál es la salida por pantalla

```php
<?php
function doble($i) {
   return $i*2;
}
$a = TRUE;
$b = "xyz";
$c = 'xyz';
$d = 12;
echo gettype($a);
echo gettype($b);
echo gettype($c);
echo gettype($d);
if (is_int($d)) {
   $d += 4;
}
if (is_string($a)) {
   echo "Cadena: $a";
}
$d = $a ? ++$d : $d*3;
$f = doble($d++);
$g = $f += 10;
echo $a, $b, $c, $d, $f , $g;
?>
```

Variables y su tipo:

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

Operadores:

- $i*2: (binario)
- $a = TRUE: (binario)
- $b = “xyz”: (binario)
- $c = ‘xyz’: (binario)
- $d = 12: (binario)
- $d += 4: (binario)
- $d = $a ? ++$d : $d*3 (ternario)
- $f = doble($d++) (binario)
- $g = $f += 10 (binario | binario)

Funciones y sus parámetros:

- doble(**$i**)
- gettype(**$a**)
- gettype(**$b**)
- gettype(**$c**)
- gettype(**$d**)
- if(**is_int($d)**)
- is_int(**$d**)
- if(**is_string($a)**)
- is_string(**$a**)
- doble(**$d++**)
- echo **$a, $b, $c, $d, $f , $g**

Estructuras de control:

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

Salida por pantalla:
booleanstringstringinteger1xyzxyz184444

Ejercicio 2:
Indicar si los siguientes códigos son equivalentes.

a)

```php
<?php
 $i = 1;
 while ($i <= 10) {
    print $i++;
 }
?>
```

```php
<?php
$i = 1;
 while ($i <= 10):
    print $i;
    $i++;
 endwhile;
?>
```

```php
<?php
 $i = 0;
 do {
    print ++$i;
 } while ($i<10);
?>
```

Si, son equivalentes.

b)

```php
<?php
for ($i = 1; $i <= 10; $i++) {
    print $i;
 }
?>
```

```php
<?php
for ($i = 1; $i <= 10; print $i, $i++) ;
?>
```

```php
<?php
for ($i = 1; ;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
 }
?>
```

```php
<?php
 $i = 1;
 for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
 }
?>
```

Si, son equivalentes.

c)

```
<?php
…
…
if ($i == 0) {
    print "i equals 0";
} elseif ($i == 1) {
    print "i equals 1";
} elseif ($i == 2) {
    print "i equals 2";
}
?>
```

```
<?php
…
…
switch ($i) {
    case 0:
        print "i equals 0";
        break;
    case 1:
        print "i equals 1";
        break;
    case 2:
        print "i equals 2";
        break;
}
?>
```

Si, son equivalentes, ya que el switch es una combinación de if / elseif en cuanto a su funcionamiento.

Ejercicio 3:
Explicar para qué se utiliza el siguiente código:

a)

```
<html>
<head><title>Documento 1</title></head>
<body>
<?php
   echo "<table width = 90% border = '1' >";
   $row = 5;
   $col = 2;
   for ($r = 1; $r <= $row; $r++) {
         echo "<tr>";
         for ($c = 1; $c <= $col;$c++) {
               echo "<td> </td>\n";
         }         echo "</tr>\n";
   }
   echo "</table>\n";
?>
</body></html>
```

Crea una tabla, y le agrega 5 filas y 2 columnas a través de código php.

b)

```
<html>
<head><title>Documento 2</title></head>
<body>
<?php
if (!isset($_POST['submit'])) {
?>
   <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
   Edad: <input name="age" size="2">
   <input type="submit" name="submit" value="Ir">
   </form>
<?php
   }
else {
   $age = $_POST['age'];
   if ($age >= 21) {
      echo 'Mayor de edad';
      }
   else {
      echo 'Menor de edad';
   }
}
?>
</body></html>
```

Si todavía no se ha utilizado el botón de submit, entonces se ejecutará el primer bloque de código (if), el cual crea un formulario con html y la acción será abrir la misma página, pero pasando como datos los datos del formulario.
Luego que se apretó el botón, la página se reinicia y se pasa a través del método POST la edad. Vuelve a ejecutarse la estructura de control y esta vez entrara por el segundo bloque (else), el cual tendrá como parámetro la edad insertada en la formulario que pasara por el POST, luego utilizando esa edad ejecutará una estructura de control php y en base a eso mostrará un mensaje u otro.

Ejercicio 4:
Si el archivo datos.php contiene el código que sigue:

```
<?php
$color = 'blanco';
$flor = 'clavel';
?>
```

Indicar las salidas que produce el siguiente código. Justificar.

```
<?php
echo "El $flor $color \n";
include 'datos.php';
echo " El $flor $color";
?>
```

El clavel blanco
El clavel blanco

Ejercicio 1:
Indicar si los siguientes códigos son equivalentes.

```
<?php
$a = array( 'color'  => 'rojo',
            'sabor'  => 'dulce',
            'forma'  => 'redonda',
            'nombre' => 'manzana',
            4
          );
?>
```

```
<?php
$a['color']  = 'rojo';
$a['sabor']  = 'dulce';
$a['forma']  = 'redonda';
$a['nombre'] = 'manzana';
$a[]         = 4;
?>
```

Si $a aún no existe, se creará, siendo también esta forma una alternativa de crear un array. Sin embargo, se desaconseja esta práctica porque que si $a ya contiene algún valor (p.ej. un string de una variable de petición), este estará en su lugar y [] puede significar realmente el operador de acceso a strings. Siempre es mejor inicializar variables mediante una asignación directa.

Ejercicio 2:
En cada caso, indicar las salidas correspondientes:

a)

```
<?php
$matriz = array("x" => "bar", 12 => true);
echo $matriz["x"];
echo $matriz[12];
?>
```

b)

```
<?php
$matriz = array("unamatriz" => array(6 => 5, 13 => 9, "a" => 42));

echo $matriz["unamatriz"][6];
echo $matriz["unamatriz"][13];
echo $matriz["unamatriz"]["a"];
?>
```

c)

```
<?php
$matriz = array(5 => 1, 12 => 2);
$matriz[] = 56;
$matriz["x"] = 42;  unset($matriz[5]);  unset($matriz);
?>
```

Salida a) bar1
Salida b) 5942
Salida c) No hay echo, no hay salida. Pero en resumen, el array $matriz se destruye.

a)

```
<?php
$fun = getdate();
echo "Has entrado en esta pagina a las $fun[hours] horas, con $fun[minutes] minutos y $fun[seconds]
segundos, del $fun[mday]/$fun[mon]/$fun[year]";
?>
```

b)

```
<?php
function sumar($sumando1,$sumando2){
  $suma=$sumando1+$sumando2;
  echo $sumando1."+".$sumando2."=".$suma;
}
sumar(5,6);
?>
```

Salida a) Has entrado en esta pagina a las 1 horas, con 37 minutos y 28 segundos, del 12/10/2023

Salida b) 5+6=11

Ejercicio 4:
Analizar la siguiente función, y escribir un script para probar su funcionamiento:

```
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;
}
```

```php
<?php
    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;
    }
    comprobar_nombre_usuario("matias")
?>
```