

**Preguntes:**

1-. Explica què valdrà \$var després d'executar:

```
$var = "1"; $var +=2; $var /=2; $var *=2.
```

\$var valdrà 3 perquè el llenguatge php fa un cast, es a dir, que converteix la primera variable que es tipus String al tipus que convé per poder fer les operacions.

De quin tipus serà?. Double

2-. Cerca i explica què fan les funcions. isset() i empty (). Quin tipus de resultat retornen?.

Isset() → Et diu si una variable ha sigut declarada. Retorna true si està declarada i definida i false si no.

Empty() → Diu si una variable està buida. Retorna true si es buida i false si no.

3-. Què fa la funció unset ()?

Unset() → Destruïx la variable que tu especifiquis.

4-. Podem crear funcions amb PHP? Si. Com? Posant la paraula function nomfuncio(){codi a executar}. Com retornem el valor? Es pot retornar el valor amb la paraula return .

Posa'n un exemple

The screenshot shows the PHP Sandbox website interface. The browser tabs include 'Curso: 2n DAW - M0...', 'function php - Busc...', 'PHP 5 Functions', 'PHP Sandbox, test...', 'PHP: Devolver valor...', 'PHP: return - Manual', and '+'. The address bar shows 'sandbox.onlinephpfunctions.com'. The website has a navigation menu with links: Home, Sandbox, PHP Functions, Custom functions, Tutorials, PHP Benchmarks, and About. The main content area is titled 'PHP Sandbox' and 'Test your PHP code with this code tester'. It includes a text area for 'Your script:' with the following code:

```
1 <?php
2 function preu_amb_tva($preu) {
3
4     return $preu_tva = $preu * 1.21;
5
6 }
7
8 echo preu_amb_tva(10);
9 >
```

Below the code editor, there are dropdowns for 'Run on PHP version:' (set to 7.0.3) and 'Output:' (set to Textbox). There are buttons for 'Execute code' and 'Save or share your code'. The 'Result:' section shows the output '12.1'. On the right side, there is a sidebar with a 'Navigation' menu listing various PHP topics and a 'Similar functions' section listing 'atanh' and 'vsprintf'.

5-. Les variables, com sempre, poden ser d'àmbit local o global. A partir d'aquesta premissa i segons el codi següent:

```
function duplicar (){
    global
    $temp;
    $temp = $temp * 2;
}
$temp = 5;
duplicar ();
echo "El valor de la variable \$temp és: $temp";
```

Prova i explica què fa la instrucció global.

La instrucció 'global' el que fa es que es pugui accedir a la variable des de qualsevol part del codi. Amb la qual cosa el codi al executar es el següent:

The screenshot shows the PHP Sandbox website interface. The main content area displays the PHP code entered in the 'Your script' field, which is the same code as provided in the task. Below the code field, the 'Run on PHP version' is set to 7.0.3, and the 'Output' is set to 'Textbox'. The 'Execute code' button has been clicked, and the result is displayed in the 'Result' field: 'El valor de la variable \$temp és: 10'. On the right side, there is a navigation menu with links to various PHP topics, and a promotional banner for 'ARE YOU THE BEST CODER?'.

**Exercici 2** «Hola mundo +» En aquest primer exercici escriurem el típic programa «Hola mundo» però fent servir un bucle for per tal que es mostre 10 vegades en la pàgina web.

El resultat de visualitzar el codi html en el navegador un cop executada la pàgina ha de ser el següent.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Una página de prueba con PHP</title>
</head>
<body>
<p>Un párrafo de texto.</p>
<p>Otro párrafo de texto.</p><p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
<p>;Hola mundo!</p>
</body>
</html>
```

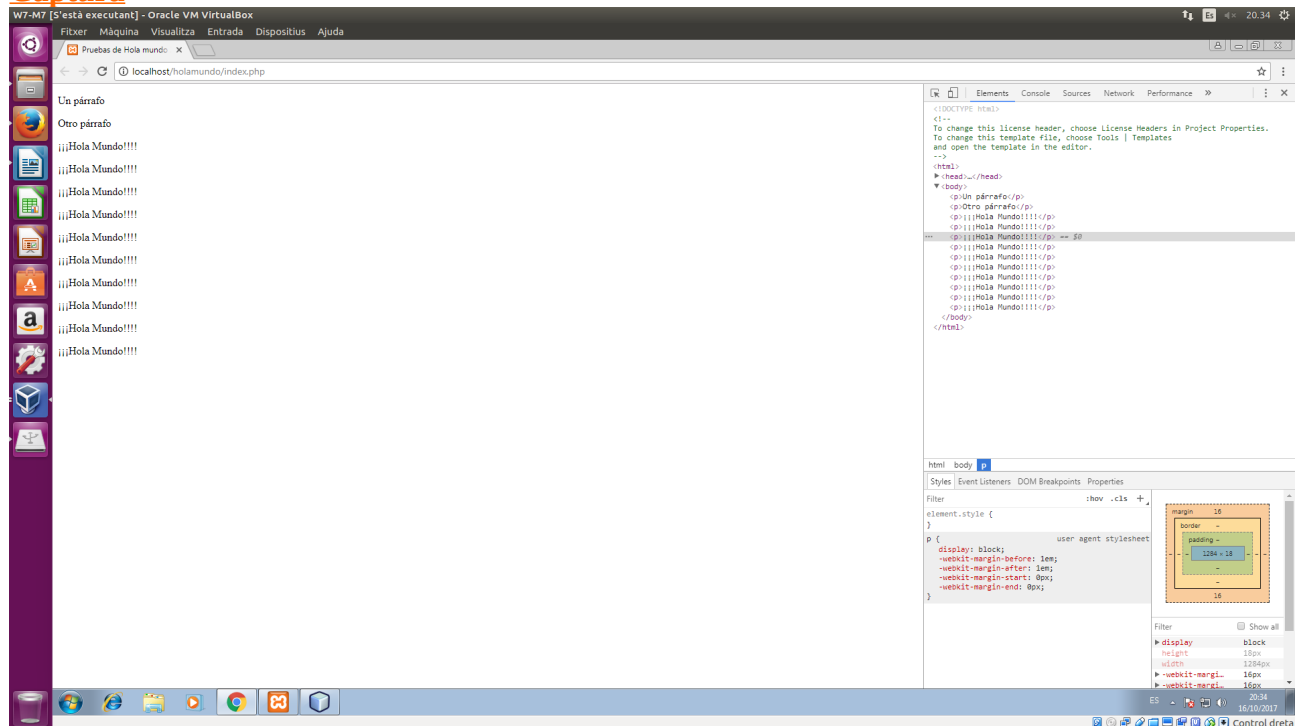
### Codi

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Pruebas de Hola mundo 2017</title>
  </head>
  <body>
    <p>Un párrafo</p>
    <p>Otro párrafo</p>
    <?php

      for($i=0;$i<10;$i++){
        echo "<p>¡¡¡Hola Mundo!!!!</p>/**/>";
      }

    ?>
  </body>
</html>
```

## Captura



**Exercici 3** Prova i executa els exemples que pots trobar al document Formularis.odp del present curs. Introdueix noves dades en els formularis i mostra la informació introduïda en el document recepcio.php fent servir els dos mètodes.

## POST



## GET



Analitza quins són els avantatges i inconvenients del mètodes post i get, quin dels dos té més capacitat per enviar dades, quin s'ha de fer servir si emprem AJAX i quin és més segur, justifica les teues respostes.

Post	Get
<ul style="list-style-type: none"><li>• Les dades se inclouen en el cos del formulari i s'envien per HTTP.</li><li>• Mostra les dades en la pantalla.</li><li>• Tracta les dades mes privadament.</li><li>• Es adequat per a formularis.</li><li>• Aconseguir que nostre navegador es connecti amb el servidor i vaig enviar les dades dels camps del formulari de manera absolutament invisible.</li><li>• A diferència del mètode get, les dades enviades pel mètode post no tenen un límit respecte a la quantitat de variables enviades en una mateixa petició, o de la longitud dels seus valors.</li></ul>	<ul style="list-style-type: none"><li>• Les dades s'agrupen al URL.</li><li>• Guarda les dades en la URL i les mostra per pantalla.</li><li>• Les dades les pot veure qualsevol.</li><li>• Porta les seves dades de forma visible al client.</li><li>• El propi usuari podrà modificar els paràmetres escrits a la URL.</li><li>• Codificar el codi HTML necessari per enviar una dada mitjançant una petició get (un enllaç que s'adjunta variables) és molt senzill, no necessitem crear cap formulari.</li><li>• Les adreces visitades quedessin guardades en l'historial del navegador.</li></ul>

**En Ajax** utilitzaria POST per les seves característiques, tracta les dades privadament, es connecta el navegador directament al servidor, no hi ha limitació a l'hora de utilitzar variables.