

## Tasca 003 Primeres pàgines en PHP

### Preguntes:

1-. Explica què valdrà \$var després d'executar: [Valdra 3](#)

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

De quin tipus serà?. [Serà de tipus integer](#)

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

`isset()` - [Determina si una variable està definida i no és null](#)

- [Retorna True o False](#)

`empty()` - [Determina si una variable està buida](#)

- [Retorna True o False](#)

3-. Què fa la funció `unset()`?

[Destrueix una variable especificada.](#)

4-. Podem crear funcions amb PHP? [SI](#)

Com? [Utilitzant la paraula function](#)

Com retornem el valor? [Amb un return, si te de retornar alguna cosa](#)

Posa'n un exemple:

```
<?php
function preu_amb_iva($preu) {
    return $preu * 1.21;
}
$preu = 10;
$preu_iva = preu_amb_iva($preu);
print "El preu amb IVA és ".$preu_iva
?>
```

5-. Les variables, com sempre, poden ser d'àmbit local o global. A partir d'aquesta premisa 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`.

L'àmbit global agafa la variable que està fora de la funció la qual te el mateix nom, i l'utilitza dintre de la funció.

## Tasca 003 Primeres pàgines en PHP

**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>
</body>
</html>
```

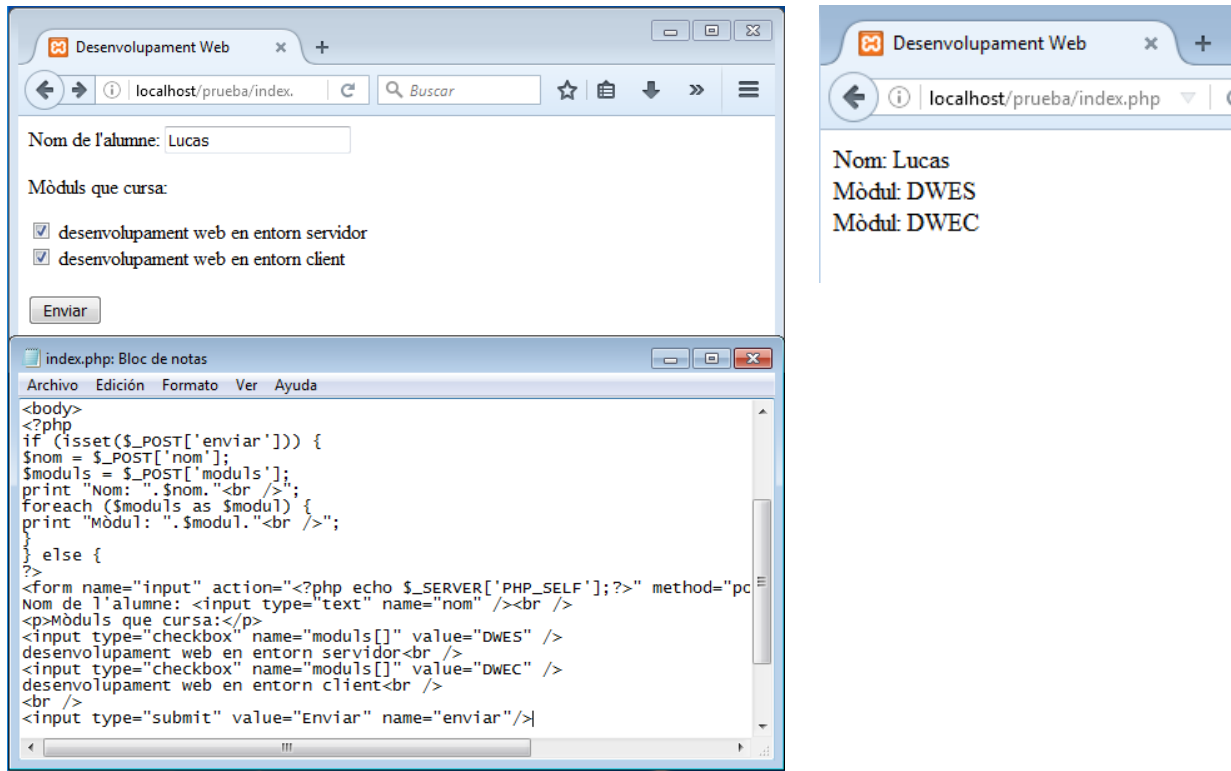
```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset=UTF-8">
  <title>Una página de prueba con PHP</title>
</head>
<body>
  <p>Un párrafo de texto.</p>
  <p>Otro párrafo de text</p>
  <?php
    for($i = 0; $i<10 ; $i++){
      echo "<p>¡Hola mundo!</p>";
    }
  ?>
</body>
</html>
```

---

```
<!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 text</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>
```

## Tasca 003 Primeres pàgines en PHP

**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.



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 fem AJAX i quin és més segur, justifica les teues respostes.

### Get

Les dades del formulari s'agrupen a l'URL i per culpa d'això, les dades les pot veure qualsevol, i el propi usuari podria modificar la URL, escrivint informació no prevista.

Hi ha una restricció en la longitud de dades, no pot superar la longitud màxima d'un url.

També només admet caràcters ASCII.

### Post

Les dades s'inclouen en el cos del formulari i s'envien utilitzant el protocol HTTP.

Les dades no són visibles, i per això és més adequat per formularis.

Admet tant text com dades binàries.

En AJAX tant és si es post o get, els dos funcionen, dependria de que volguessis fer.

El més segur seria el post, per que no es podrien veure les dades, no es guardarien les dades en el historial amb el url. A més en el get és molt més fàcil que pugui ser accedit per un robot que dispara les accions.