

## Ejercicios puntuales

1. Partiendo de un archivo con la línea `$titulo = 'Tutorial PHP'`, imprimir esta variable como título de un documento HTML. Esto se debe ver similar a:

### Tutorial PHP

**PHP** es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

2. Partiendo del archivo **register.php** que se encuentra en el campus:
  - a. Agregar un campo de "Confirmar contraseña"
  - b. Agregar funcionalidad para que si se recibe por GET un parámetro con el nombre "**versionCorta**" no se muestre el campo de confirmar contraseña.
  - c. Escribir en PHP un array con al menos 10 países. Dado esto incluir una etiqueta `<select>` para que el usuario pueda elegir su nacionalidad. Las opciones deben mostrarse dinámicamente según contenido del array.
3. Crear un archivo llamado **registro\_usuarios.php** que contenga un breve formulario de registro, utilizar **GET** como método y luego de que se registra un usuario nuevo:
  - a. Que envíe los datos a una página nueva llamada "**confirmacion.php**".
  - b. Dentro de esta pagina que se muestre por pantalla todos los datos ingresados, respetando cierta estética, Ej. *Muchas Gracias por registrarte NOMBRE COMPLETO, nos has dicho que tienes X años (mostrar edad). Hemos registrado tu email, (mostrar EMAIL). ¡Gracias!*
  - c. Tener en cuenta para elementos como checkboxes que el elemento puede no existir o tener múltiples valores (los cuales pueden ser resueltos con el uso de **IF** o **FOR**)
4. Modificar **registro\_usuarios.php** y **confirmacion.php** para que envíe y procese los datos por el método POST para evitar cualquier problema.
5. A modo de prueba haremos que el formulario envíe los datos al mismo archivo (**registro\_usuarios.php**). Una vez hecho esto haremos que el formulario ya aparezca cargado con aquellos campos que el usuario ya completó (exceptuando la contraseña) en caso de que el formulario ya haya sido enviado.
6. Modificar **confirmacion.php** para que en caso de que no haya datos enviados por post, redirija a **registro\_usuarios.php**.
7. Utilizando `strpos()`, encontrar la posición de la primera aparición de "php" dentro de la cadena: "Me encanta php, A Mi tambien me encanta php!"
8. A través de PHP, validar **register.php** desde el lado del servidor. La validación debe realizarse en el mismo archivo. En caso de éxito redirigir a **felicitaciones.html** que

mostrará un mensaje de felicitaciones. En caso de errores debe indicar cuales son los errores y debe rellenar el formulario con aquellos campos que ya se habían enviado.

## Ejercicios complementarios

1. Realizar un archivo **faq.php**, en donde exista una serie de preguntas y respuestas. Hacer que las mismas estén definidas como contenido de un array. Luego, imprimir el contenido HTML a partir de un **FOR** que recorra las preguntas y respuestas.

Ejemplo de array:

```
<?php
```

```
$faq = [
    ["pregunta" => "Cuanto es 1 + 1?", "respuesta" => "2"],
    ["pregunta" => "Cuando es 2 + 2?", "respuesta" => "4"]
];
```

```
?>
```

2. Partiendo de un archivo con la línea \$color = array('Blanco', 'Verde', 'Rojo').Escribir un script PHP que muestra los colores de la siguiente manera:  
Blanco, Verde, Rojo:
  - Blanco
  - Verde
  - Rojo
3. Modificar el ejercicio anterior para que cada elemento del array tenga el nombre del color y el valor hexadecimal (css) del mismo. Así, podríamos dibujar cada nombre con su color indicado. Se recomienda para esto utilizar el atributo **style y propiedad color**, para dentro del mismo imprimir el valor hexadecimal requerido.
4. Crear un array donde se especifica el nombre de un artículo con su respectivo precio. Escribir un script PHP para mostrar la secuencia, los valores dentro de una tabla, en la columna izquierda su nombre y en la derecha su precio.

Output:

Costo Articulo A	\$23421
Costo Articulo B	\$423
Costo Articulo C	\$8289

