DigitalHouse > Coding School

Fullstack Web Development

PHP

Práctica Presencial 4

GET Y POST

Objetivo

Luego de retomar los conceptos aprendidos mediante el contenido multimedia, debatirlos y sacarnos las dudas practiquemos un poco cómo transmitimos datos en la web.

Consignas

1. Crea el archivo *imprimir.php* que imprima la variable $_GET a través de un var_dump y acceder al mismo con el siguiente Query String:

   imprimir.php?nombre=montoto&email=montoto@digitalhouse.com

2. Crea el archivo *formulario.html* en la misma carpeta que el anterior, con el siguiente código y accede al mismo:

```
<!DOCTYPE HTML>
	<html>
		<head>
			<title>TEST</title>
		</head>
		<body>

			<form action="imprimir.php" method="get">
				<label for="nombre">Nombre:</label>
				<input type="text" name="nombre">
				<br>
				<label for="email">E-mail:</label>
				<input type="text" name="email">
				<br>
				<input type="submit">
			</form>

		</body>
	</html>
```

   a. ¿Qué sucede ahora en imprimir.php si enviá el formulario con datos?
   b. Modifica *imprimir.php* para que imprima algún valor puntual (por ejemplo el nombre que se envíe en el formulario) de $_GET.
   c. Modifica *imprimir.php* para que imprima todos los valores utilizando un **foreach**.

3. Modifica el archivo *formulario.php* para que envíe los datos a través de POST. ¿Cómo deberías modificar *imprimir.php* para que siga funcionando y puedas ver los datos que envias?

4. Proba la función de php *getAllHeaders()* e imprimir su resultado. ¿Qué información ves?

5. Así como imprimiste $_POST y $_GET existen otras variables globales que podes usar. Probá cada una de estas variables en un pedido vacío y luego agregándole datos:

   (cada uno tiene el link a la documentación oficial de PHP si querés saber algo más!)

   a. **$_SERVER**
   b. **$_FILES**
   c. **$_REQUEST**
   d. **$_SESSION**
   e. **$_COOKIE**
   f. **$GLOBALS**

VALIDACIÓN Y PERSISTENCIA

Objetivo

Ahora qué manipulamos un poco mejor los métodos de transmisión de datos vamos a querer chequear que la información que viaje sea la que nos va a servir para trabajar, por eso vamos a tener que validar la misma, y en el caso de qué no esté completa de manera adecuada vamos a querer que el usuario no tenga qué volver a completar todos los datos *(esto es una muy buena práctica y mejora mucho la experiencia del usuario en nuestra web)*

Consignas

6. Añadiremos validación para el formulario de registración. Para esto te sugerimos algunos factores para tener en cuenta:
   a. Una buena estrategia es que el formulario de registración haga un pedido por POST hacia el mismo archivo. Esto permite que primero valides y luego lo envíes a la página de éxito.
   b. Es importante validar primero que nada si el usuario envió información.

   c. Si el usuario efectivamente envió información deberías validar campo por campo tomando el enfoque de “preguntar si hay un error en el campo”.
   d. Es importante acumular todos los errores para poder mostrarlos de forma ordenada en el HTML.
   e. Si tras todas las validaciones no hay ningún error, deberías redirigir al usuario a la página de éxito.
   f. Para redirecciones ver **header('Location: http://www.example.com/');**
7. En caso de que el formulario traiga errores tenés persistir los datos que el usuario ya había enviado en el formulario.

VERSIÓN NINJA: ¡En ésta versión sería ideal persistir sólo aquellos campos que no hayan tenido errores! ¿Se te ocurre cómo hacerlo?