

GET y POST

1. Crear 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. Crear el archivo **formulario.html** en la misma carpeta que el anterior, con el siguiente código y acceder al mismo:

```
<!DOCTYPE HTML>
<html>
<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 enviamos el formulario con datos?
 - b. Modificar **imprimir.php** para que imprima algún valor puntual (por ejemplo el nombre) de `$_GET`.
 - c. Modificar **imprimir.php** para que imprima todos los valores utilizando un `foreach`.
3. Modificar el archivo del **punto 2** para que envíe los datos a través de **POST**. ¿Cómo deberíamos modificar **imprimir.php** para que siga funcionando y podamos ver los datos que enviamos?
 4. Probar la función de php `getAllHeaders()` e imprimir su resultado. ¿Qué encontramos?
 5. Así como imprimimos `$_POST` y `$_GET` existen otras variables globales que podemos usar. Probar cada una de estas variables en un pedido vacío y luego agregándoles datos:
 - a. `$_SERVER`
 - b. `$_FILES`
 - c. `$_REQUEST`
 - d. `$_SESSION`
 - e. `$_COOKIE`

f. `$GLOBALS`

Persistencia y Validación

1. Añadiremos validación para el formulario de registración. Para esto les 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 validemos y luego lo enviemos a la página de éxito.
 - b. Es importante validar primero que nada si el usuario envió información.
 - c. Si el usuario efectivamente envió información vamos a querer 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, podemos redirigir al usuario a la página de éxito.
 - Para redirecciones ver [header\("Location..."\)](#)
2. En caso de que el formulario traiga errores queremos persistir los datos que el usuario ya había enviado en el formulario.
 - a. ¡En una segunda versión sería ideal persistir sólo aquellos campos que no hayan tenido errores!