

Bienvenidos

Clase 05. Introducción a PHP
Validaciones – Conexión a DB

Aplicaciones Móviles Y Cloud Computing

Temario

04

Principios de PHP Validaciones – Conexión DB

- ✓ Introducción a PHP
- ✓ Validaciones con PHP
- ✓ Conexión a DB con PHP

05

Validaciones PHP Archivos PHP Conexión DB

- ✓ [Validaciones con PHP](#)
- ✓ [Manejo básico archivos con PHP](#)
- ✓ [Conexión a DB con PHP](#)

05

Desarrollo FullStack

- ✓ Conceptos de desarrollo FullStack
- ✓ Stacks de programación



Objetivos de la clase

- Aprender a realizar validaciones con PHP
- Manejo básico de archivos con PHP
- Aprender a trabajar con PHP, en conexión con una BD



Validaciones con PHP



Validaciones con PHP

- ✓ `isset($variable)`: valida si la variable argumento existe
- ✓ `empty($variable)`: valida que la variable enviada como parámetro tenga valor
- ✓ `is_string($var)`: valida si \$var es una cadena de caracteres
- ✓ `preg_match(expr, $var)`: valida si se cumple la expresión regular expr, en \$var
- ✓ `filter_var($var, FILTER_)`: se utiliza para validar tipos de dato especiales (email, por ejemplo). Filtra una variable, según determinado filtro.

[Documentación Oficial PHP](#)

```
if(isset($_POST["nombre"]) &&
    isset($_POST["email"]) &&
    isset($_POST["password"])){

    $nombre=$_POST["nombre"];
    $email=$_POST["email"];
    $password=$_POST["password"];

    if(!empty($_POST["nombre"]) &&
        !empty($_POST["email"]) &&
        !empty($_POST["password"])){

        if(!is_string($nombre) || preg_match("/[0-9]/", $nombre)){
            $error="Error: nombre invalido";
            // header("Location:index.php?error=$error");
        }

        if(!is_string($email) || !filter_var($email, FILTER_VALIDATE_EMAIL)){
            $error="Error: email invalido";
            // header("Location:index.php?error=$error");
        }

        if(strlen($password)<6){
            $error="Error: password debe tener al menos 6 caracteres";
            // header("Location:index.php?error=$error");
        }

        // echo "<h2>Datos correctos</h2>";
        var_dump($error);
        if($error){
            header("Location:index.php?error=$error&nombre=$nombre&email=$email&password=$password");
        }
    }
}
```



Manejo básico de archivos con PHP



FileSystem en PHP

- ✓ `file_exists("nombreArchivo.json")`: valida existe el archivo enviado como argumento
- ✓ `file_get_contents("nombreArchivo.json")`: recupera el contenido de un archivo, en formato texto
- ✓ `file_put_contents("nombreArchivo.json", $datos)`: escribe en el archivo al que apunta el primer argumento, los datos enviados como segundo parámetro a la función
- ✓ `json_decode` y `json_encode`: sirven para pasar de texto a Json, y de Json a texto, respectivamente.

```
if(file_exists("usuarios.json")){  
    $json=file_get_contents("usuarios.json");  
    $usuarios=json_decode($json);  
}else{  
    $usuarios=[];  
}
```

```
$usuariosTexto=json_encode($usuarios, JSON_PRETTY_PRINT);  
file_put_contents("usuarios.json", $usuariosTexto);
```

[Documentacion Oficial PHP](#)



Universidad
Provincial del Sudoeste
Promoviendo el Desarrollo Armónico de la Región

Acceso a Base de Datos con PHP



Conexión a Base de Datos

Existe una gran cantidad de funciones que sirven para este fin. En su mayoría comienzan con "mysqli_". Algunos ejemplos:

- ✓ `mysqli_connect()`: establece la conexión con la DB. Recibe como argumentos la dirección del server, usuario y clave, y DB
- ✓ `mysqli_connect_errno()`: devuelve true si ocurre algún error en una instrucción ejecutada contra la DB
- ✓ `mysqli_error()`: devuelve el detalle del error, si es que existe alguno.
- ✓ `mysqli_query($conexión, $query)`: ejecuta una query en la DB indicada por \$conexión.
- ✓ `mysqli_fetch_assoc($query)`: recorre un recordset, y asigna cada fila a un array asociativo.

```
// conexión a MySQL
$conexion=mysqli_connect('localhost', 'root','','cursoupso01');

if(mysqli_connect_errno()){
    echo "La conexión ha DB ha fallado. ".mysqli_connect_error();
}else{
    echo "Conexión a DB realizada correctamente...!!!";
}

// configurar codificación consultas
mysqli_query($conexion, "SET NAMES 'utf8'");

// consulta:
$query=mysqli_query($conexion, "SELECT * FROM usuarios");
var_dump($query);

echo "Primer print:<br>";
while($usuario=mysqli_fetch_assoc($query)){
    // var_dump($usuario);
    echo $usuario['nombre']."<br>";
    echo $usuario['email']."<br>";
}
echo "<hr>";

// Volver a la primera posición del resultado
mysqli_data_seek($query, 0);
```

[Documentación Oficial PHP](#)



Universidad
Provincial del Sudoeste
Promoviendo el Desarrollo Armónico de la Región

Ejercicio:

Siguiendo el ejemplo desarrollado en clase, diseñe un login básico, que permita acceder con email y password. Considerar el mismo esquema de tabla, cuya estructura se muestra en la página siguiente.

Ejercicio:

phpMyAdmin

Servidor: 127.0.0.1 » Base de datos: cursoupso01 » Tabla: usuarios

Examinar Estructura SQL Buscar Insertar Exportar Importar

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	nombre	varchar(256)	utf8_general_ci		No	Ninguna		
<input type="checkbox"/> 3	email	varchar(256)	utf8_general_ci		No	Ninguna		
<input type="checkbox"/> 4	password	varchar(256)	utf8_general_ci		No	Ninguna		

Esquema tabla usuarios

¿Preguntas?



Universidad
Provincial del Sudoeste
Promoviendo el Desarrollo Armónico de la Región

Muchas gracias.



Universidad
Provincial del Sudoeste
Promoviendo el Desarrollo Armónico de la Región