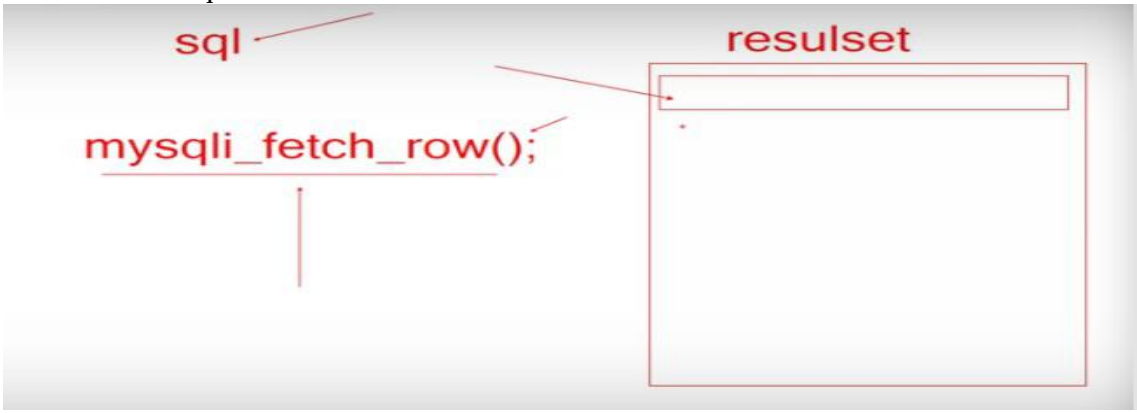
Vimos cómo el el recorset o resulset, no es más que una tabla virtual donde se almacena la instrucción sql.



Es decir al ejecutar esta instrucción sql , la respuesta se almacena en una tabla virtual.
Esta función mysqli_fetch_row(), lo que hace es acceder la primera vez al primer registro de esa resulset. Si la llamas una segunda vez automáticamente lo que hace es acceder al segundo registro y así sucesivamente…
por lo tanto vamos a copiar y pegar dos veces la instrucción con los echo…(ojo añadir un br al final del primero para que salte a la línea siguiente.

```
$resultados=mysqli_query($conexion, $consulta)

$fila=mysqli_fetch_row($resultados);

echo $fila[0] . " ";

echo $fila[1] . " ";

echo $fila[2] . " ";

echo $fila[3] . " ";

$fila=mysqli_fetch_row($resultados);

echo $fila[0] . " ";

echo $fila[1] . " ";

echo $fila[2] . " ";

echo $fila[3] . " ";
```

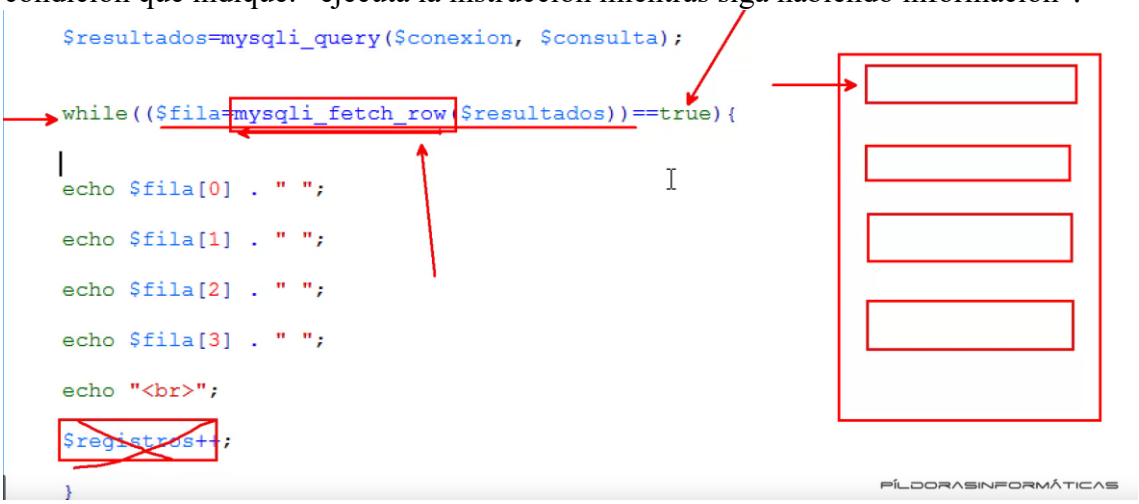
Este código te devuelve los dos primeros registros.
Si siguiéramos copiando y pegando una tercera vez, te leerá la tercera línea, y as

…pero si tienes 3.000 líneas no lo vas a cortar y pegar 3000 veces, así que la solución es hacer un bucle

Primero inicias una variable registros a 1
Y luego haces el bucle

```
$resultados=mysqli_query($conexion, $consulta);

$registros=1;

while($registros<=3){

$fila=mysqli_fetch_row($resultados);

echo $fila[0] . " ";

echo $fila[1] . " ";

echo $fila[2] . " ";

echo $fila[3] . " ";

echo "<br>";

$registros++;

}
```

While tiene como condición la instrucción $registros<=3 , por que sabemos que son 3 registros.
Qué conseguimos con esto. Partimos de la variable registros es 1, y el while indica que mientras registros sea menor que tres se ejecute la función mysqli_fetch_row(), con el parámetro $resultados donde se almacena la conexión a la base de datos.
Según va aumentando el bucle se va imprimiendo cada vez un echo, hasta imprimir todos.
El problema está en que no sepamos el total de registros en la tabla. Así que la forma definitiva de hacerlo sería: cambiar la condición del while, y cambiarlo por una condición que indique: ”ejecuta la instrucción mientras siga habiendo información”.



Con esto conseguimos que el código cuando entra una vez en el bucle while, ejecuta la instrucción , y si es capaz de ejecutarla es por que en la tabla hay información(registros)

Funciona así: cuando entra en esta condición del while, y es (true), verdad que encuentra registro, lo imprime, ….así hasta que llega al final, y cuando llega al final. y entonces devuelve false, sale del bucle.
La condición del While se podría simplificar, eliminando el true, puesto que al ser una condición booleana, se toma por defecto que esto es true.
En definitiva podría quedar así:

```
while($fila=mysqli_fetch_row($resultados)){

echo $fila[0] . " ";

echo $fila[1] . " ";

echo $fila[2] . " ";

echo $fila[3] . " ";

echo "<br>";
```

Por último deberíamos cerrar la conexión.
Me explico: hemos abierto una conexión, esa conexión ha permanecido abierta todo el rato. Esto consume recursos. Si la dejas abierta ahora no vas a tener ningún problema pero para optimizar recursos, es una vez que ya no necesites acceder a la base de datos, cerra r la conexión.
Y la conexión se cierra con el **mysqli close**, indicando que conexión quieres cerrar.
.

```
    echo $fila[0] . " ";

    echo $fila[1] . " ";

    echo $fila[2] . " ";

    echo $fila[3] . " ";

    echo "<br>";


    }

    mysqli_close($conexion);
?>
```

Un mismo documento php, se puede conectar a un montón de bases de datos por lo que será importante que se vayan cerrando las conexiones, para optimizar recursos

- Cómo manejar los errores que puedan ocurrir en el proceso de conexión con la base de datos.
    - Por haber escrito mal la dirección de la base de datos
    - Por haber escrito mal el nombre de la base de datos

Errores

- para que nos avisara que había un fallo en la conexión utilizamos el if con mysqli_connect-errno, después de el mysqli_connect.

```
$conexion=mysqli_connect($db_host,$db_usuario,$db_contra,$db_nombre);
if(mysqli_connect_errno()){
    echo "Fallo al conectar con la BBDD";
    exit();
}
$consulta="SELECT * FROM DATOSPERSONALES";
```

Con un mensaje de error que indique que hay un fallo en la conexión. Esta función se ejecuta cuando hay algún error en la conexión con la base de datos.
muy importante!. Que después de ejecutar el mensaje, salga del código php, (con el **exit(),** por que si no seguirá leyendo código, y pasará a intentar ejecutar la consulta,…y dará fallo tras fallo.
provoco poner mal la dirección del servidor (pongo localhooost o 12777,0,0,1 en lugar de localhost o 127.0.0.1

- para que no nos de error de tipografía como en el ejemplo....

51972854W MAR□A GOMEZ 27

utilizamos el set_Charset (después de la conexión), para indicar que queremos utilizar el juego de caracteres utf 8

```
mysqli_set_charset($conexion, "utf8");
```

y entonces ya nos saldrán los acentos normales.

- Ahora vamos a imaginarnos que el error está en la base de datos (por ejemplo

porque ponemos mal el nombre de la base de datos…pruebassss);
En este caso podremos quitar el comando opcional de mysqli_connect **,** ***$db_nombre*** y lo sustituimos, líneas mas abajo, por ***mysqli_select_db($conexion, $db_nombre) añadiendo or die (no se encuentra la BBDD)***

De este modo nos va a devolver un mensaje que es más exacto puesto que dirá ***no se encuentra la BBDD,*** en caso de que haya un error en el nombre.