◉リスト　select/results.php

```
<?php
    declare(strict_types=1);

    require_once dirname(__FILE__) . '/functions.php';

    // メインルーチン
    try {
        if (!isset($_GET['title']) || trim($_GET['title']) === '') {
            return;
        }
        $pdo = connect();
        $statement = $pdo->prepare("SELECT * FROM books WHERE title LIKE :title
ESCAPE '#' ORDER BY published DESC");  ←❸
        $statement->bindValue(':title', '%' . escapeLike($_GET['title']) . '%', PDO::
PARAM_STR);
        $statement->execute();
    } catch (PDOException $e) {
        echo '本の検索に失敗しました。';
        return;
    }
?>
<body>
    <h3>タイトルに「<?=escape($_GET['title'])?>」を含む書籍の検索結果</h3>
    <table border="1">
        <tr>
            <th>タイトル</th>
            <th>価格</th>
            <th>発売日</th>
        </tr>
        <?php while ($row = $statement->fetch(PDO::FETCH_ASSOC)):  ←❶
            <tr>
                <td><?=escape($row['title'])?></td>
                <td><?=escape(number_format($row['price']))?>円</td>
                <td><?=escape($row['published'])?></td>
            </tr>
        <?php endwhile; ?>
    </table>
</body>
```

◉リスト　select/functions.php

```
<?php
    declare(strict_types=1);

    /**
     * PDOインスタンスを取得する関数
     */
    function connect(): PDO
    {
```

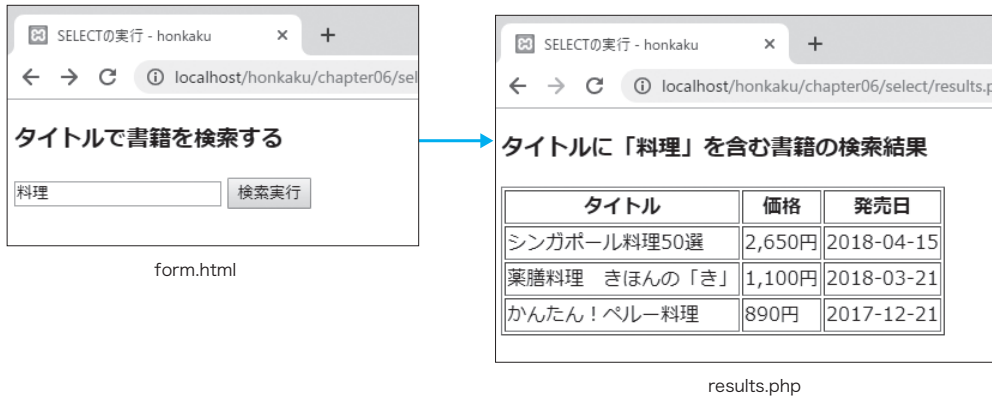
```
    $pdo = new PDO('mysql:host=localhost; dbname=honkaku; charset=utf8mb4', 'root', '');
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $pdo->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
    return $pdo;
}

/**
 * HTMLエスケープする関数
 */
function escape(?string $value)
{
    return htmlspecialchars(strval($value), ENT_QUOTES | ENT_HTML5, 'UTF-8');
}

/**
 * LIKE演算子のワイルドカードをエスケープする関数
 */
function escapeLike(?string $value)
{
    return preg_replace('/([_%#])/u', '#${1}', $value);
}
```

❷

◉実行結果



form.html

results.php

フェッチをおこなっているのが❶です。フェッチモードにFETCH_ASSOCを指定しているので、$row['title']のように、カラム名が連想配列のキーになっています。1回目にfetchメソッドをコールした時は1レコード目、2回目にコールした時は2レコード目が変数$rowに保存され、最後のレコードが終わると戻り値がfalseになることで、whileループを抜けます。

fetchAllメソッドを使ってこの処理を書くと、以下のようになります（fetchAllほうが、より多くメモリを消費することに注意しましょう）。

```
<?php $rows = $statement->fetchAll(PDO::FETCH_ASSOC); ?>
<?php foreach ($rows as $row): ?>
    <tr>
```

```
            <td><?=escape($row['title'])?></td>
            <td><?=number_format(escape($row['price']))?>円</td>
            <td><?=escape($row['published'])?></td>
        </tr>
    <?php endforeach; ?>
```

escapeLike関数(❷)は、❸のSQL中のESCAPE句で指定した「#」記号を使って、ワイルドカード記号をエスケープします。ワイルドカード記号である「%」「_」と、エスケープ記号自身である「#」がエスケープ対象となります。

6-3-6 PDO::FETCH_OBJによるフェッチ

フェッチモードにPDO::FETCH_OBJを指定した時のフェッチ結果は、PHPにあらかじめ用意されているstdClassというクラスのインスタンスとなります。stdClassは空のクラスであり、publicプロパティのみを持ちえます。メソッドは持っていません。

Note フェッチモードとコーディング規約への配慮

フェッチモードとしてよく使われるのは、FETCH_ASSOC（連想配列でフェッチ）か、FETCH_OBJ（オブジェクトでフェッチ）です。基本的には好みで選んでいいと思いますが、コーディング規約との兼ね合いが1つの選択基準になる可能性があります。

データベースのカラム名は習慣的に、「member_name」のようなスネークケース記法を使います。その一方で、PHPのコーディング規約では、

- **プロパティ名は「$memberName」のようなローワーキャメル記法とすること**
- **連想配列のキーはどんな記法でもいい**

というルールになっていることが多いです。

もしこのルールであった場合、FETCH_ASSOCでレコードを取得したときは$row['member_name']のようになるのでルール違反とはならず、FETCH_OBJなら$row->member_nameのようになるのでルール違反となってしまいます。FETCH_ASSOCのほうが、コーディング規約に反するリスクは少ないといえます。

6-3-7 プリコンパイルすることで、SQL実行時のセキュリティリスクを減らす

これまでのプログラム例でたびたび登場したPDO::prepareメソッドは、内部的にはデータベースに対して、実行前のSQL文をプリコンパイル（SQLの構文解析をすませ、機械語に変換し、値をバインドするだけで実行できるよう準備しておくこと）せよという指示を出しています。そして、このプリコンパイル済のSQLのことを、プリペアードステートメントと呼びます。

SQLのプリコンパイルは、PHPの機能ではなく、データベースが持っている機能です。