

# 講座：PDOを使ったPHPでのページネーション

大量のデータ（たとえば記事）を表示する際、ページネーションは欠かせない技術です。Webページを重くしすぎず、ユーザー体験を向上させます。

この講座では、**PDOを使ってMySQLデータベースと接続し、PHPでクラシックなページネーションを実装する手順を説明します。**

## 1. データベースの準備

以下のテーブルを使用します。これはすでにビールデータベースに存在しています：

```
CREATE TABLE `article` (  
  `ID_ARTICLE` int(11) NOT NULL,  
  `NOM_ARTICLE` varchar(60) NOT NULL,  
  `PRIX_ACHAT` double NOT NULL,  
  `VOLUME` int(11) NOT NULL,  
  `TITRAGE` double DEFAULT NULL,  
  `ID_MARQUE` int(11) DEFAULT NULL,  
  `ID_Couleur` int(11) DEFAULT NULL,  
  `ID_TYPE` int(11) DEFAULT NULL  
);
```

## 2. データベースへの接続

まずはPDOでデータベースへ接続しましょう。PDOオブジェクトを変数に保存し、後のクエリで使います。

```
// これはサンプルの接続情報です。実際には適切な値にしてください。  
$pdo = new PDO('mysql:host=localhost;dbname=ma_base;charset=utf8mb4', 'user',  
  'password');
```

## 3. 現在のページを取得

URLから `page` パラメータを読み取り、現在のページを決定します。指定がなければ、1ページ目とします。

```
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;  
// マイナスページなどの異常値もチェックするとよいです。
```

表示する記事数を設定します：

```
$articlesPerPage = 10;
```

SQLクエリ用のオフセット（**OFFSET**）を計算します：

```
$offset = ($page - 1) * $articlesPerPage;  
// 1ページ目ではオフセットは0になります。
```

---

## 4. 記事の総数を取得

ページ数を計算するために、まず記事の**総数**を取得します：

```
$total = $pdo->query('SELECT COUNT(*) FROM article')->fetchColumn();  
// fetchColumn は1列のみの結果を取得します。  
$totalPages = ceil($total / $articlesPerPage);
```

---

## 5. 現在のページの記事を取得

オフセットとリミットがわかったので、現在のページに表示すべき記事を取得します：

```
$stmt = $pdo->prepare('SELECT * FROM article LIMIT :limit OFFSET :offset');  
$stmt->bindValue(':limit', $articlesPerPage, PDO::PARAM_INT);  
$stmt->bindValue(':offset', $offset, PDO::PARAM_INT);  
$stmt->execute();  
$articles = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

---

## 6. 記事の表示

取得した結果をHTMLで表示します：

```
foreach ($articles as $article) {  
    echo '<p>' . htmlspecialchars($article['NOM_ARTICLE']) . ' - ' .  
    $article['PRIX_ACHAT'] . ' €</p>';  
}
```

---

## 7. ページリンクの表示

400ページ分のリンクをすべて表示するのは非効率です。**現在のページの前後数ページのみ表示**するようにします。

関数を作成しましょう：

```
/**
 * ページネーションHTMLを生成
 *
 * @param integer $currentPage 現在のページ
 * @param integer $totalPages 総ページ数
 * @param integer $window 前後に表示するページ数
 * @return string HTML文字列
 */
function renderPagination(int $currentPage, int $totalPages, int $window = 2):
string {

    $html = '<nav><ul class="pagination">';

    $start = max(1, $currentPage - $window);
    $end = min($totalPages, $currentPage + $window);

    if ($currentPage > 1) {
        $html .= '<li><a href="?page=' . ($currentPage - 1) . '"><</a></li>';
    }

    for ($i = $start; $i <= $end; $i++) {
        $class = $i === $currentPage ? 'class="active"' : '';
        $html .= "<li $class><a href=?page=$i>$i</a></li>";
    }

    if ($currentPage < $totalPages) {
        $html .= '<li><a href="?page=' . ($currentPage + 1) . '">>></a></li>';
    }

    $html .= '</ul></nav>';
    return $html;
}
```

そして表示：

```
echo renderPagination($page, $totalPages);
```

## ✓ 期待される結果

たとえば `index.php?page=3` にアクセスすると：

- 3ページ目の **記事10件のリスト** が表示される
- 「前へ」「次へ」やページ番号などの **ページネーションリンク** が表示される

## 🧠 UXのヒント

- Bootstrapのクラス (`pagination`, `page-item`, `page-link`) でデザインできます。
  - **最初と最後のページ** へのリンクを追加することもできます。
  - 範囲外のページではリンクを無効化するのも良いです。
- 

## 🚩 まとめ

このページネーションの技術は、PHPにおけるリスト表示の基本です。APIや検索、管理画面などにも応用できます。