

September 19, 2025

```
mkdir output
mkdir uploads

sudo chown -R www-data:www-data /var/www/html/pdf/output
sudo chown -R www-data:www-data /var/www/html/pdf/uploads

sudo chmod -R 755 /var/www/html/pdf/output
sudo chmod -R 755 /var/www/html/pdf/uploads
```

main.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>PDF アップロード（選択+D&D+並べ替え）</title>
  <style>
    #dropArea {
      padding: 30px;
      border: 2px dashed #888;
      background-color: #f9f9f9;
      margin-bottom: 15px;
      text-align: center;
    }

    .file-item {
      padding: 10px;
      border: 1px solid #ccc;
      margin: 6px 0;
      background: #fff;
      cursor: move;
    }

    .sortable-ghost {
      opacity: 0.4;
    }
  </style>
</head>
<body>

  <h1>アップロードされたファイルはドラッグアンドドロップで順番を変更できます。</h1>

  <form id="pdfForm">
    <!-- ドロップエリア -->
    <div id="dropArea">ここにPDFファイルをドラッグ&ドロップ、または下から選択</div>

    <!-- ファイルリスト -->
    <div id="fileList"></div>

    <!-- 通常のファイル選択 -->
    <input type="file" id="fileSelector" accept="application/pdf" multiple><br><br>

    <!-- 操作選択 -->
    <label for="actionSelect" style="color: red; font-weight: bold; background-color: yellow;">選択し
      てください！ :</label>
    <select name="action" id="actionSelect" required>
      <option value="split">分割（1ファイル）</option>
      <option value="merge">結合（複数ファイル）</option>
    </select><br><br>

    <button type="submit">実行</button>
  </form>
```

```

<!-- SortableJS -->
<script src="https://cdn.jsdelivr.net/npm/sortablejs@1.15.0/Sortable.min.js"></script>

<script>
  const dropArea = document.getElementById('dropArea');
  const fileList = document.getElementById('fileList');
  const fileSelector = document.getElementById('fileSelector');
  const form = document.getElementById('pdfForm');
  let files = [];

  // ファイル追加関数
  function addFile(file) {
    if (file.type !== 'application/pdf') return;

    // 同じファイル名は2回追加しない
    if (files.some(f => f.name === file.name)) return;

    files.push(file);

    const div = document.createElement('div');
    div.className = 'file-item';
    div.textContent = file.name;
    div.setAttribute('data-filename', file.name);
    fileList.appendChild(div);
  }

  // ドロップイベント処理
  ['dragenter', 'dragover', 'dragleave', 'drop'].forEach(eventName => {
    dropArea.addEventListener(eventName, e => e.preventDefault(), false);
  });

  dropArea.addEventListener('dragover', () => dropArea.style.background = '#eef');
  dropArea.addEventListener('dragleave', () => dropArea.style.background = '#f9f9f9');
  dropArea.addEventListener('drop', (e) => {
    dropArea.style.background = '#f9f9f9';
    const droppedFiles = Array.from(e.dataTransfer.files);
    droppedFiles.forEach(addFile);
  });

  // 通常のファイル選択でも追加可能
  fileSelector.addEventListener('change', () => {
    Array.from(fileSelector.files).forEach(addFile);
    fileSelector.value = ''; // 同じファイルを再選択できるように
  });

  // 並び替え対応
  Sortable.create(fileList, {
    animation: 150,
    ghostClass: 'sortable-ghost'
  });

  // フォーム送信処理
  form.addEventListener('submit', function (e) {
    e.preventDefault();

    const formData = new FormData();
    const action = document.getElementById('actionSelect').value;
    formData.append('action', action);

    // 表示順にファイルを追加
    const items = fileList.querySelectorAll('.file-item');
    items.forEach(item => {
      const name = item.getAttribute('data-filename');
      const file = files.find(f => f.name === name);
      if (file) {
        formData.append('pdf[]', file, file.name);
      }
    });

    fetch('upload.php', {

```

```

        method: 'POST',
        body: formData
    })
    .then(response => response.text())
    .then(result => {
        document.body.innerHTML = result;
    })
    .catch(err => {
        alert('アップロードに失敗しました');
        console.error(err);
    });
});
</script>

</body>
</html>

```

upload.php

```

<?php
// アップロードされたPDFを処理
$uploadDir = __DIR__ . '/uploads/';
$outputDir = __DIR__ . '/output/';
@mkdir($uploadDir, 0777, true);
@mkdir($outputDir, 0777, true);

// ファイルアップロードの存在チェック
if (!isset($_FILES['pdf'])) {
    die("ファイルがアップロードされていません");
}

// 操作の種類取得
$action = $_POST['action'] ?? 'split';

if ($action === 'split') {
    // 分割は単一ファイルのみ対応
    $uploadedFile = $_FILES['pdf'];
    if (is_array($uploadedFile['name'])) {
        // 複数ファイルアップロードの場合、最初のファイルだけ使う
        $filename = basename($uploadedFile['name'][0]);
        $tmpPath = $uploadedFile['tmp_name'][0];
    } else {
        $filename = basename($uploadedFile['name']);
        $tmpPath = $uploadedFile['tmp_name'];
    }
    $destPath = $uploadDir . $filename;

    if (!move_uploaded_file($tmpPath, $destPath)) {
        die("ファイル保存失敗");
    }

    // 出力フォルダの古いPDFを削除（必要に応じて）
    foreach (glob($outputDir . '*.pdf') as $oldFile) {
        unlink($oldFile);
    }

    $outputPattern = $outputDir . pathinfo($filename, PATHINFO_FILENAME) . '_%d.pdf';
    $cmd = "pdfseparate " . escapeshellarg($destPath) . " " . escapeshellarg($outputPattern);
    exec($cmd, $output, $returnVar);

    if ($returnVar !== 0) {
        die("分割に失敗しました");
    }

    // リンク表示
    echo "<h2>PDFを分割しました:</h2><ul>";
    foreach (glob($outputDir . '*.pdf') as $file) {
        $url = rawurlencode(basename($file));
        echo "<li><a href='/pdf/output/$url' target='_blank'>" . htmlspecialchars(basename($file)) .
            "</a></li>";
    }
    echo "</ul>";
}

```

```

} elseif ($action === 'merge') {
    // 複数ファイルアップロード対応
    $files = $_FILES['pdf'];
    $numFiles = is_array($files['name']) ? count($files['name']) : 0;
    if ($numFiles === 0) {
        die("結合するファイルが選択されていません");
    }

    $inputFiles = [];

    for ($i = 0; $i < $numFiles; $i++) {
        if ($files['error'][$i] !== UPLOAD_ERR_OK) {
            die("ファイルアップロードエラー");
        }
        $tmpName = $files['tmp_name'][$i];
        $name = basename($files['name'][$i]);
        $destPath = $uploadDir . $name;
        if (!move_uploaded_file($tmpName, $destPath)) {
            die("ファイル保存失敗");
        }
        $inputFiles[] = $destPath;
    }

    // 出力ファイル名
    $outputFile = $outputDir . 'merged.pdf';

    // pdfunite で結合
    $cmd = 'pdfunite ' . implode(' ', array_map('escapeshellarg', $inputFiles)) . ' ' .
        escapeshellarg($outputFile);
    exec($cmd, $output, $returnVar);

    if ($returnVar !== 0) {
        die("結合に失敗しました");
    }

    // ダウンロードリンク表示
    $url = rawurlencode(basename($outputFile));
    echo "<h2>PDFを結合しました:</h2>";
    echo "<a href='/pdf/output/$url' target='_blank'>" . htmlspecialchars(basename($outputFile)) .
        "</a>";
} else {
    echo "不明な操作です。";
}

```