

# ToDoリストアプリケーションの開発

---

ToDoを管理するコマンドラインアプリケーションを開発します。

ToDoとは「やらなければいけないこと」や「忘れてはいけないこと」を意味します。

---

## ToDoリストアプリケーション概要

ToDoリストアプリケーションには3つの機能があります。

1. ToDo表示機能 (`todo_list.php`)
2. ToDo追加機能 (`todo_add.php`)
3. ToDo完了機能 (`todo_finish.php`)

各機能から参照する関数及び定数は`functions.php`に記述します。またアプリケーションで参照するデータは`todo_list.csv`ファイルに保存するものとします。

本演習では合計5つのファイルを使用します。ただし`functions.php`、`todo_list.csv`については完成しているものを配布します。

---

## todo\_list.csv

ToDoリストアプリケーションで参照するデータは`todo_list.csv`ファイルに保存するものとします。

```
1, 部屋を片付ける, "2019-10-01 11:00:00", 1
2, PHPの本を買いに行く, "2019-10-01 12:34:56", 0
3, PHPを勉強する, "2019-10-02 14:00:30", 0
```

`todo_list.csv`は以下の項目が, で区切られています。

- ID
  - タスク
  - 作成日時
  - ステータス (0: 未完了、1:完了)
-

## 1. ToDo表示機能 (todo\_list.php)

コマンドラインで以下のように実行します。

```
$ php todo_list.php
2:PHPの本を買いに行く (2019-10-01 12:34:56)
3:PHPを勉強する (2019-10-02 14:00:30)
```

todo\_list.csvの内容が表示されます。ただし、ステータスが0：未完了のレコードのみ表示します。

## 2. ToDo追加機能 (todo\_add.php)

コマンドラインでタスクを入力して実行します。

```
$ php todo_add.php HTMLを勉強する
```

todo\_list.csvに新規レコードが追加されます。

```
1,部屋を片付ける,"2019-10-01 11:00:00",1
2,PHPの本を買いに行く,"2019-10-01 12:34:56",0
3,PHPを勉強する,"2019-10-02 14:00:30",0
4,HTMLを勉強する,"2019-10-03 10:12:50",0
```

新規レコードのIDは自動で割り振ります。また作成日時は現在の日時、ステータスには0を登録します。

### 入力エラー：タスクが未入力の場合

```
$ php todo_add.php
タスクが未入力です。
```

### 入力エラー：タスクが140文字を超える場合

```
$ php todo_add.php
1234567890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901
タスクが140文字を超えています。
```

### 3. ToDo完了機能 (todo\_finish.php)

コマンドラインでIDを指定して実行します。

```
$ php todo_finish.php 2
```

todo\_list.csvにおいて、IDが2のレコードのステータスが1（完了）に更新されます。

```
1,部屋を片付ける,"2019-10-01 11:00:00",1  
2,PHPの本を買いに行く,"2019-10-01 12:34:56",1  
3,PHPを勉強する,"2019-10-02 14:00:30",0  
4,HTMLを勉強する,"2019-10-03 10:12:50",0
```

#### 入力エラー：不正なIDの場合

```
$ php todo_finish.php A  
入力されたIDは不正です。
```

未入力の場合も同様にエラーとします。

---

## functions.php

`functions.php`には以下の定数と関数を定義します。

### 定数一覧

定数	値
<code>STATUS_OPENED</code>	"0"
<code>STATUS_CLOSED</code>	"1"
<code>TODO_LIST_CSV</code>	"todo_list.csv"
<code>TASK_MAX_LENGTH</code>	140
<code>MESSAGE_TASK_EMPTY</code>	"タスクが未入力です。"
<code>MESSAGE_TASK_MAX_LENGTH</code>	"タスクが140文字を超えています。"
<code>MESSAGE_ID_INVALID</code>	"入力されたIDは不正です。"

定数は`define`関数を使って定義します。

### 関数一覧

関数	処理
<code>read_todo_list</code>	todo_list.csvファイルを読み込み、ToDoリスト（配列データ）返却する
<code>get_new_todo_id</code>	ToDoの新規IDを返却する
<code>add_todo_list</code>	新規ToDoを既存のToDoリストに追記する
<code>write_todo_list</code>	ToDoリストをtodo_list.csvファイルに書き込む

`todo_list.csv`ファイルに対する読み込みや書き込みを関数として定義します。

```
<?php
define("STATUS_OPENED", "0");
define("STATUS_CLOSED", "1");
define("TODO_LIST_CSV", "todo_list.csv");

define("TASK_MAX_LENGTH", 140);
define("MESSAGE_TASK_EMPTY", "タスクが未入力です。");
define("MESSAGE_TASK_MAX_LENGTH", "タスクが140文字を超えています。");
define("MESSAGE_ID_INVALID", "入力されたIDは不正です。");

function read_todo_list($include_closed = true)
{
    $handle = fopen(TODO_LIST_CSV, "r");
    $todo_list = [];
    while ($todo = fgetcsv($handle)) {
        if (!$include_closed &&
            $todo[3] === STATUS_CLOSED) {
            continue;
        }
        $todo_list[] = $todo;
    }
    fclose($handle);
    return $todo_list;
}

function get_new_todo_id()
{
    return count(read_todo_list()) + 1;
}

function add_todo_list($todo)
{
    $handle = fopen(TODO_LIST_CSV, "a");
    fputcsv($handle, $todo);
    fclose($handle);
}

function write_todo_list($todo_list)
{
    $handle = fopen(TODO_LIST_CSV, "w");
    foreach ($todo_list as $todo) {
        fputcsv($handle, $todo);
    }
    fclose($handle);
}
```

`functions.php`、および`todo_list.csv`については完成版を配布しています。今回の演習においては作成する必要はありません。

# 1. ToDo表示機能の開発

---

以下のTerminalの期待値を出力する`todo_list.php`を開発します。

## Terminal

コマンドラインで以下のように実行します。

```
$ php todo_list.php
2:PHPの本を買いに行く (2019-10-01 12:34:56)
3:PHPを勉強する (2019-10-02 14:00:30)
```

`todo_list.csv`の内容が表示されます。ただし、**ステータス**が**0**：未完了のレコードのみ表示します。

---

## todo\_list.php

### 処理記述

- `require_once`関数によって`functions.php`を参照する
- `read_todo_list`関数の引数に`false`を指定して呼び出し、ToDoリストを取得する
- ToDoリストを画面に出力する

```
<?php
require_once("functions.php");

$todo_list = ???(false);

for ($i = 0; $i < count($todo_list); $i++) {
    $todo = $todo_list[$i];
    echo ???
}
```

`require_once`関数は、異なるPHPファイルを読み込むための関数です。ここでは`functions.php`ファイルを読み込むために使います。

---

## functions.php

```
<?php
define("STATUS_OPENED", "0");
define("STATUS_CLOSED", "1");
define("TODO_LIST_CSV", "todo_list.csv");

function read_todo_list($include_closed = true)
{
    $handle = fopen(TODO_LIST_CSV, "r");
    $todo_list = [];
    while ($todo = fgetcsv($handle)) {
        if (!$include_closed
            && $todo[3] === STATUS_CLOSED) {
            continue;
        }
        $todo_list[] = $todo;
    }
    fclose($handle);
    return $todo_list;
}

# その他の定数、及び関数は省略...
```

- `define`関数を使って定数を定義する
  - `fgetcsv`関数はCSV形式のレコードを読み込み、戻り値を配列で返す
  - 引数の`$include_closed`に`false`が指定された場合、ステータスが0：未完了のものだけを`$todo_list`に追加する
-

以下のTerminalの期待値を出力する**todo\_add.php**を開発します。

コマンドラインで**タスク**を入力して実行します。

todo\_list.csvに新規レコードが追加されます。

```
1,部屋を片付ける,"2019-10-01 11:00:00",1
2,PHPの本を買いに行く,"2019-10-01 12:34:56",0
3,PHPを勉強する,"2019-10-02 14:00:30",0
4,HTMLを勉強する,"2019-10-03 10:12:50",0
```

新規レコードのIDは自動で割り振ります。また作成日時は現在の日時、ステータスには0を登録します。

```
$ php todo_add.php
タスクが未入力です。
```

```
$ php todo_add.php
1234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901
タスクが140文字を超えています。
```



## todo\_add.php

## 処理記述

- `require_once`関数によって`functions.php`を参照する
- 入力データを検証する。以下のケースに該当する場合はエラーメッセージを表示してプログラムを終了する
  - コマンドライン引数の要素数が2でない場合
  - コマンドライン引数の`タスク`が未入力の場合
  - コマンドライン引数の`タスク`が140文字を超える場合
- 新規`ToDo`を作成する
  - `get_new_todo_id`関数を呼び出して、新規`ID`を取得する
  - `date`関数を使って現在の日時に`作成日時`を取得する
  - `STATUS_OPENED`定数を参照して、ステータスを取得する
  - `add_todo_list`関数を呼び出して`ToDo`を追記する

```
<?php
require_once("functions.php");

if (count($argv) != ???) {
    die(MESSAGE_TASK_EMPTY . PHP_EOL);
}

$task = $argv[1];
if ($task === ???) {
    die(MESSAGE_TASK_EMPTY . PHP_EOL);
}

if (mb_strlen($task) > ???) {
    die(MESSAGE_TASK_MAX_LENGTH . PHP_EOL);
}

$id = ???();
$date = ???('Y-m-d H:i:s');
$status = ???;
$todo = [$id, $task, $date, $status];
???( $todo);
```

`mb_strlen`関数はマルチバイト文字用の文字列関数です。日本語データを扱うときに利用します。マルチバイト文字用の文字列関数を有効にするには`php.ini`ファイルの`mbstring`モジュールを有効にしておく必要があります。

## functions.php

```
<?php
define("TASK_MAX_LENGTH", 140);
define("MESSAGE_TASK_EMPTY", "タスクが未入力です。");
define("MESSAGE_TASK_MAX_LENGTH", "タスクが140文字を超えています。");

function get_new_todo_id()
{
    return count(read_todo_list()) + 1;
}

function add_todo_list($todo)
{
    $handle = fopen(TODO_LIST_CSV, "a");
    fputcsv($handle, $todo);
    fclose($handle);
}

# その他の定数、及び関数は省略...
```

- `get_new_todo_id`関数は、`read_todo_list`関数を呼び出し、戻り値の配列の件数 + 1を返す
- `add_todo_list`関数は引数で受け取ったタスク (`$todo`) をファイルに書き込む
- `fputcsv`関数は配列データをCSV形式に置き換えてファイルに書き込む

### 3. ToDo完了機能

---

以下のTerminalの期待値を出力する**todo\_finish.php**を開発します。

#### Terminal

コマンドラインで**ID**を指定して実行します。

```
$ php todo_finish.php 2
```

**todo\_list.csv**において、**ID**が**2**のレコードのステータスが**1**（完了）に更新されます。

```
1,部屋を片付ける,"2019-10-01 11:00:00",1
2,PHPの本を買いに行く,"2019-10-01 12:34:56",1
3,PHPを勉強する,"2019-10-02 14:00:30",0
4,HTMLを勉強する,"2019-10-03 10:12:50",0
```

#### 入力エラー：不正なIDの場合

```
$ php todo_finish.php A
入力されたIDは不正です。
```

未入力の場合も同様にエラーとします。

---

## todo\_finish.php

## 処理記述

- `require_once`関数によって`functions.php`を参照する
- 入力データを検証する。以下のケースに該当する場合はエラーメッセージを表示してプログラムを終了する
  - コマンドライン引数の要素数が2でない場合
  - コマンドライン引数のIDが未入力（あるいは不正な値）の場合
- `read_todo_list`関数を呼び出してToDoリストを取得する
- ToDoリストの中から、入力データのIDと一致するものを探索する
  - 入力データのIDと一致するものが存在する場合
    - 該当するToDoのステータスを`STATUS_CLOSED`に更新する
- `write_todo_list`関数を使ってToDoリストを上書き保存する

```
<?php
require_once("functions.php");

if (count($argv) != 2) {
    die(MESSAGE_ID_INVALID . PHP_EOL);
}

$id = (int)$argv[1];
if ($id <= 0) {
    die(MESSAGE_ID_INVALID . PHP_EOL);
}

$todo_list = ???();
foreach ($todo_list as &$todo) {
    if ((int)$todo[0] === $id) {
        $todo[3] = ???;
        break;
    }
}
???( $todo_list);
```

`foreach`構文を使って配列データを処理しています。添字配列も`foreach`構文で処理できます。

## functions.php

```
<?php
define("MESSAGE_ID_INVALID", "入力されたIDは不正です。");

function write_todo_list($todo_list)
{
    $handle = fopen(TODO_LIST_CSV, "w");
    foreach ($todo_list as $todo) {
        fputcsv($handle, $todo);
    }
    fclose($handle);
}

# その他の定数、及び関数は省略...
```

