

## 第3回 定数・別ファイル読み込み・フォーム送信①

### 3 - 1. 定数

#### 3 - 1 - 1. 定数

定数とは、数値や文字列などの値を、決まった文字列で定義しておけるものです。プログラム内では、値を直接記述する代わりに、定義した文字列を記述することで呼び出すことができます。

定数は宣言後に値を変更することはできません。

定数の宣言は下記のように **define 関数** を用いて行います。

#### ■定数(構文)

```
define('定数名', '値')
```

#### ■定数(例)

```
define('TAX', '0.1')
```

例の「TAX」のように、**定数名は、全て大文字**で宣言することが一般的です。

また、値に配列を指定することも可能です。配列の場合は、値を `[]` で囲みます。

#### ■定数(例:配列)

```
define('TYPE', ['String','int','double']);
```

define による定数の宣言の可否は下記の通りです。

#### ■define による定数の宣言

トップレベルでの宣言 (クラスの外)	○
クラス直下での宣言	×
メソッド内での宣言	○



define の値には、変数も指定することができます。

【例】

```
$abc = 'こんにちは';
```

```
define('TEST', $abc); // 定数 TEST の値には、「こんにちは」が格納される
```

### 3-1-2. メソッド

PHP はスクリプト言語なので、class やメソッドを記載しなくても、PHP ファイルを作成することが可能ですが、class やメソッドを宣言することもできます。

この授業では PHP の導入部分ですので設計には触れませんが、メソッドの宣言方法については触れておきます。

#### ■メソッドの宣言(戻り値あり)

```
アクセス修飾子 function メソッド名(引数){  
    // ここに処理を記載  
    return 戻り値;  
}
```

#### ■メソッドの宣言(戻り値なし)

```
アクセス修飾子 function メソッド名(引数)::void{  
    // ここに処理を記載  
}
```

※アクセス修飾子は省略することも可能です。省略すると public となります。

## 3 – 2. 外部ファイルの読み込み

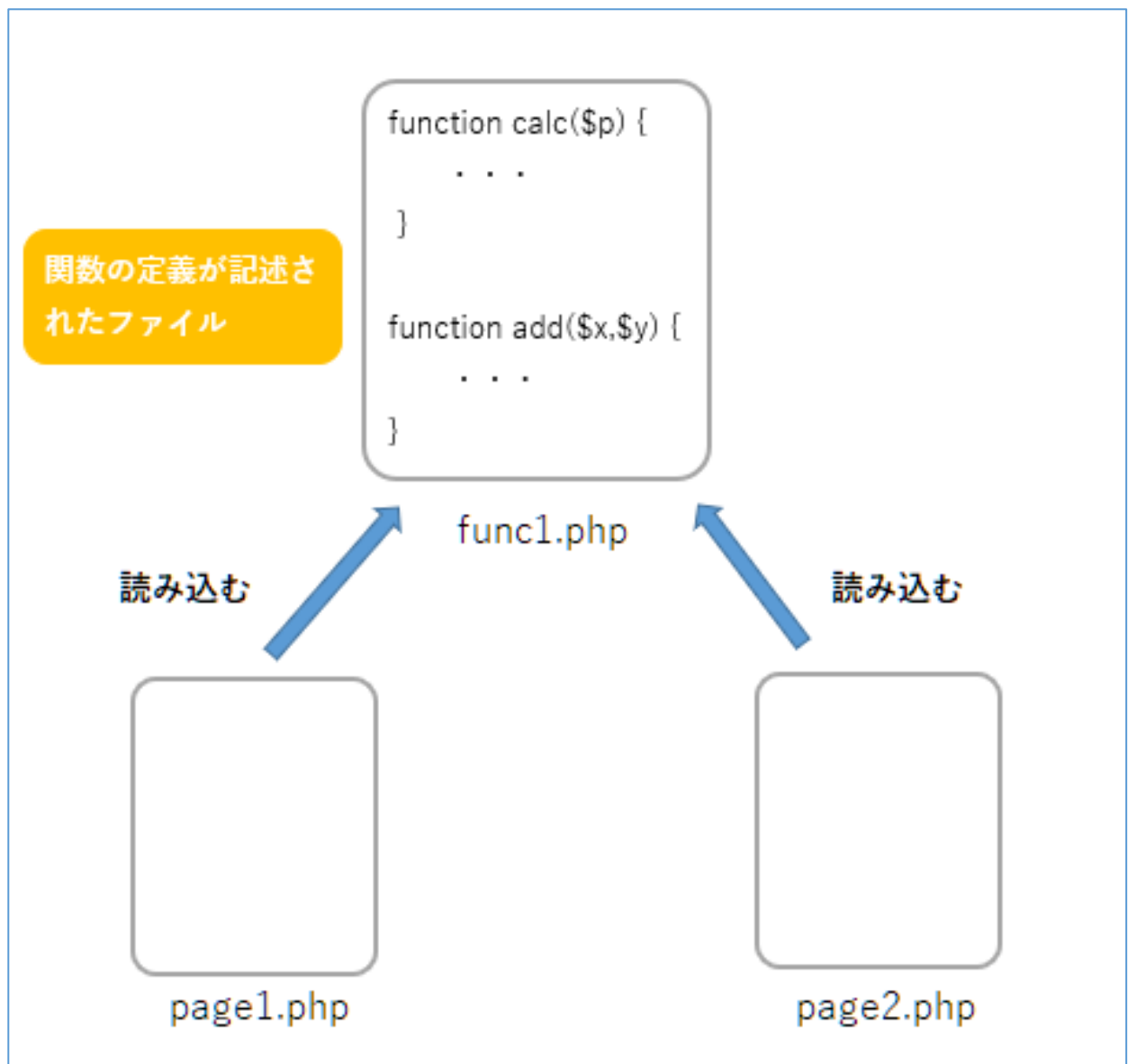
### 3 – 2 – 1. 外部ファイル読み込み

プログラミング設計では、用途ごとに class を作成し、一まとめの処理を関数に記述しましたね。

（設計は Java の授業で習得するので、ここでは概要のみ記載します）

PHP でも、複数の Web ページにて使用する関数は 1 つのファイルにまとめておき、それぞれの Web ページから読み込むようにすることができます。

ここでは、別ファイルの読み込み方法について学んでいきましょう。



ファイルの読み込みに使用する命令には下記が使用できます。

命令	意味
require	外部ファイルを読み込み、読み込めなければ処理を中止する
require_once	外部ファイルが既に読み込まれていない場合のみ、require 命令を実行する
include	外部ファイルを読み込み、読み込めなくても処理を続行する
include_once	外部ファイルが既に読み込まれていない場合のみ、include 命令を実行する

関数を記述したようなファイルは、

- 2重に同じファイルを読み込むとエラーになる場合もあること
- ファイルを読み込めないまま処理が続行しても関数が使えないこと

上記のような点から、require\_once での読み込みが安全側です。

## 3 - 3. オブジェクト定数

### 3 - 3 - 1. オブジェクト定数

クラス直下で宣言する、すなわちクラス固有の定数を「**オブジェクト定数**」といいます。

オブジェクト定数の宣言方法は、下記です。

#### ■オブジェクト定数(構文)

アクセス修飾子 **const** オブジェクト定数名 = 値

※アクセス修飾子は省略可能。(省略した場合はpublic)

### ■const によるオブジェクト定数の宣言

トップレベルでの宣言 (クラスの外)	○
クラス直下での宣言	○
メソッド内での宣言	×



const の値には、変数や関数の戻り値を指定することはできません。

【例】

```
$abc = 'こんにちは';
```

```
const TEST = $abc; // エラーとなる
```

### ■自クラス内から参照するとき

self::オブジェクト定数名

### ■自クラス外から参照するとき

クラス名::オブジェクト定数名

## 3 - 4. GET でのデータ送受信

### 3 - 4 - 1. GET データの送信

#### ■<form>開始タグ

method プロパティ	GET
action プロパティ	データの送信先を記載

前期の HTML・CSS の授業を思い出して下さい。

#### ■SampleCode (テキストボックスの内容を GET で送信)

※テキストボックス (input type="text" の箇所) の name 属性の値が「パラメータ名」となる。

下記の場合は、**name="keyword"**なので、「keyword」がパラメータ名として扱われる。

```
<form action="sample01_2.php" method="GET">
  <label for="keyword" class="form-label">
    キーワードを入力してください。

  </label>
  <input type="text" id="" class="form-control form-control-lg"
name="keyword">
  <div class="p-5 d-grid gap-2 d-md-flex justify-content-md-end">
    <button type="submit" class="btn btn-danger btn-lg">
      送信
    </button>
  </div>
</form>
```

## ■Sample（テキストボックスの内容を GET で送信・ブラウザ表示）

サーバーサイドスクリプト演習 1

サンプル

sampleフォーム（GET送信）

キーワードを入力してください。

ここはテキストボックスで入力できます

送信

Sample 画面の[送信]ボタン押下時、GET 形式でデータが送信されます。

ブラウザのアドレスバーには、「<http://localhost/>フォルダ名/ファイル名?パラメータ名1=値 1&パラメータ名2=値 2」というように URL の末尾に GET で送信したデータが追加されます。

## ■Sample 送信結果画面

※画像の赤枠部分（今回は送信したテキストボックスは 1 件なので、パラメータも 1 つのみ）



## 3 - 4 - 2. GET データの受け取り

### ■GET データを取得できるスーパーグローバル変数

<code>\$_GET</code>	HTTP GET 変数  URL パラメータで、現在のスクリプトに渡された変数の連想配列になります。  HTTP GET メソッドのデータだけではなく、クエリストリングも含まれます。
---------------------	---

### ■SampleCode（テキストボックスの内容を受け取り、変数\$result に格納）

※テキストボックス（input type="text"の箇所）の name 属性の値が「パラメータ名」となる。

Sample の送信時に、**name="keyword"**で送信したので、**`$_GET["keyword"]`**で値が取得できる。

```
<?php
    $result = $_GET["keyword"];
?>
```

表示するには HTML 部分の記載が必要。（Sample では割愛）



