第３回　定数・別ファイル読み込み・フォーム送信①

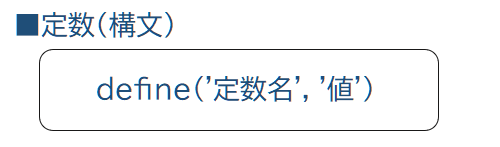
３－１．定数

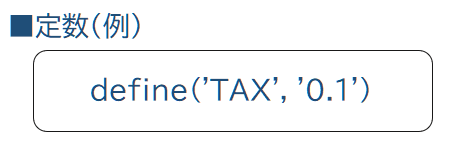
３－１－１．定数

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。

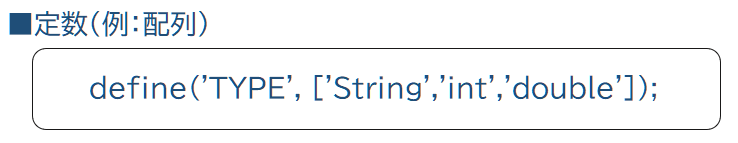
定数の宣言は下記のようにdefine関数を用いて行います。





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。



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

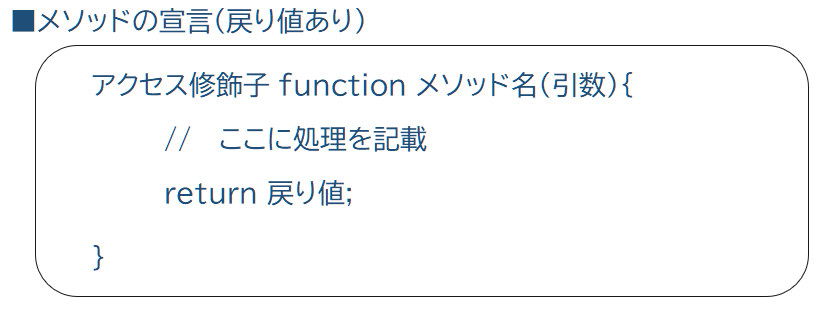
|  |  |
| --- | --- |
| トップレベルでの宣言  （クラスの外） | **〇** |
| クラス直下での宣言 | **×** |
| メソッド内での宣言 | **〇** |

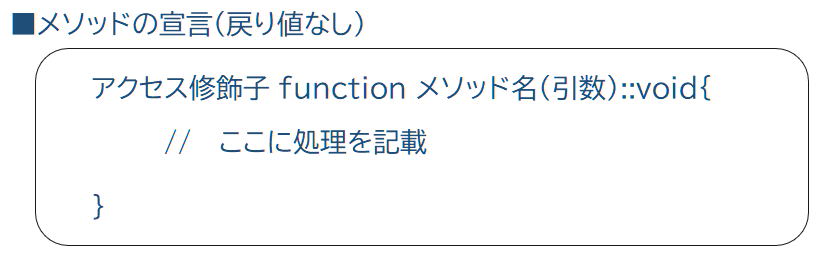
|  |  |
| --- | --- |
| 警告 単色塗りつぶし | defineの値には、変数も指定することができます。  【例】　 $abc = ‘こんにちは’;  define(‘TEST’, $abc); // 定数TESTの値には、「こんにちは」が格納される |

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。





　※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

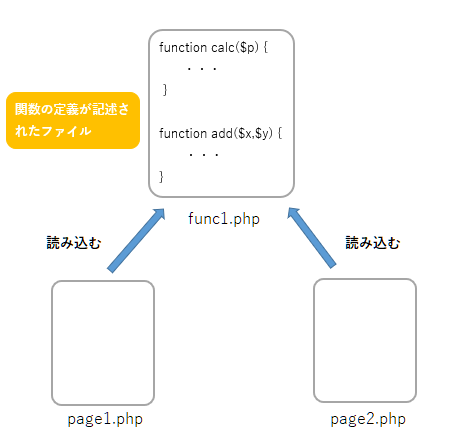
３－２－１．外部ファイル読み込み

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

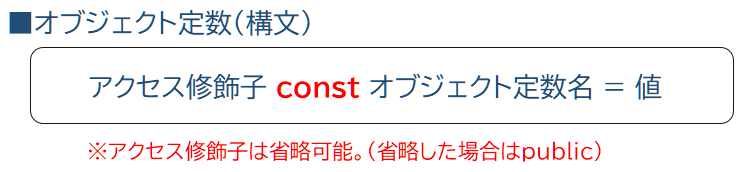
上記のような点から、require\_onceでの読み込みが安全側です。

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。



■constによるオブジェクト定数の宣言

|  |  |
| --- | --- |
| トップベルでの宣言  （クラスの外） | **〇** |
| クラス直下での宣言 | **〇** |
| メソッド内での宣言 | **×** |

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | 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 |



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）



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

|  |  |
| --- | --- |
| $\_GET | 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）