PHP
クラスの作り方

①クラス名（設計図名）をつける

②クラス内で値を入れておくための、
メンバ変数（プロパティ）を作る

③クラスで何を行いたいのか、の
メンバ関数（メソッド）を作る

③’クラス内で、
メンバ変数・メンバ関数に
アクセスするときは、$this

④クラス（設計図）を元に、
newでオブジェクトを作る
（インスタンス化）

⑤クラスから作ったオブジェクトを使って、
メンバ変数・メンバ関数に
アクセスするとき ->(アロー演算子)

```
$obj = new クラス名();
$obj -> メンバ変数名 = 1;
$obj -> メンバ関数名A(5);

class クラス名{
    public メンバ変数名;
    const 定数名 = 値;
    function __construct(変数、変数, …){
        初期化する処理;
    }
    function メンバ関数名(){
        print $this -> メンバ変数名;
    }
    function メンバ関数名A(引数){
        $this -> メンバ変数名 = 引数;
        $this -> メンバ関数名();
        return 値;
    }
}
```

⑥メンバ変数で引数を受け取る

⑦オブジェクトからメンバ関数を呼び出す
（引数受け渡しも）

⑧値を呼び出し元に返す時は、return

⑨クラス内に定数を設定するとき
※変数ではないので、$はつけない！

⑨’クラス内から定数を参照するとき
self::定数名　or　クラス名::定数名

※定数は、オブジェクト毎ではなく、クラス単位で
設定されるので、$thisはオブジェクト毎、
selfはクラス、として考える

⑩オブジェクトがnewで作られるときに、
自動で呼び出されるメソッド
※初期化が必要なときに便利！

© NullNote

PHP～クラス アクセス修飾子

メンバ変数・メンバ関数に
アクセスできる範囲を指定するもの

public	クラス内・クラス外の どこからでもアクセス可
private	同じクラス内からのみ アクセス可（オブジェクト からのアクセス不可）
protected	同じクラス、及び 子クラスからアクセス可

```
<!--サンプル-->
<?PHP
class クラス名{
    private メンバ変数名 = 'こんにちは';
    public static function メンバ関数名(){
        echo 'Hello';
    }
}
echo クラス名::sayHello();
?>
```

PHP～クラス staicのこと

staticが指定されたメソッド・プロパティは、
インスタンスを作らずに実行できる。

◎静的なものにする・全体で共有される

> staticが指定されたメソッド・プロパティは
> $thisが使えない

NG `echo $this -> メンバ変数名;`

> staticはインスタンス毎のデータが扱えない
> ＝ 実質グローバル関数と同じ

© NullNote