

- [Aucun ami connecté](#)
- [2 999 Zéros connectés](#)
- [436 285 Zéros inscrits](#)
- 
- [Gya](#)
 - [Avatar](#)
 - [Profil](#)
 - [Identifiants](#)
 - [Études](#)
 - [Réglages](#)
 - [Contacts](#)
 - [Commandes](#)
 -
- [Lisez vos Messages Personnels](#)

Correction du Q.C.M.

[<= Retour au cours](#)

-

Question

Quelle classe nous permettra d'obtenir des informations concernant d'autres classes ?

- Bonne réponse

ReflectionClass

- Explications

C'est la question « de base », si vous vous êtes planté je ne peux vous dire que « bien joué ! ». 😊

-

Question

Quelle méthode de la classe évoquée en première question permet de savoir si notre classe est la fille de telle autre classe ?

- Bonne réponse

isSubclassOf()

- Explications

Il s'agit de la méthode `isSubclassOf()`, les deux autres n'existent bien évidemment pas.

-

Question

Quelle classe nous permet d'obtenir des informations à propos des attributs d'une autre classe ?

- Bonne réponse

ReflectionProperty

- Explications

La seconde réponse était la bonne. En effet, dans l'API de réflexivité, vous verrez toujours le mot « property » pour désigner un attribut, et non « attribute » comme on pourrait l'imaginer.

-

Question

Est-il possible, grâce à cette classe, de pouvoir modifier les attributs privés de nos objets ?

- Bonne réponse

Oui

- Explications

Eh oui, c'est possible. Je vous rappelle que c'est faisable en rendant l'attribut accessible puis en le modifiant.



-

Question

Quelle classe permet d'obtenir des informations à propos des méthodes de nos classes ?

- Bonne réponse

ReflectionMethod

- Explications

Il s'agit bien sûr de la première réponse. Sachez que la seconde classe existe réellement, je vous laisse la découvrir grâce à la documentation si ça vous intéresse. 😊

-

Question

Que m'affichera ce code ?

Code : PHP - Sélectionner

```
<?php
class MaClasse
{
    public function MaClasse()
    {
        echo 'Hello world !';
    }
}

$methode = new ReflectionMethod('MaClasse', 'MaClasse');

if ($methode->isConstructor())
    echo 'Constructeur';
else
    echo 'Non constructeur';
?>
```

○ Bonne réponse

« Constructeur »

• Explications

Je vous rappelle qu'une méthode ayant le même nom que la classe est considérée comme étant le constructeur, et ce à cause des traces que PHP 4 a laissé derrière lui. Cependant, même si ça fonctionne, **ne faites jamais ça : utilisez toujours la méthode magique `__construct`**. 😊



Note : 20 / 20

[<= Retour au cours](#)