Les commentaires

1. Qu'est-ce qu'un commentaire :

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l’interpréteur, car destinées en général à un lecteur humain et non censées influencer l’exécution du programme.

1. La bonne pratique :

Tous les commentaires doivent être rédigés en anglais et doivent décrire clairement le bloc de code commenté.

Référence pour les bonnes pratique [psr](https://github.com/php-fig-rectified/fig-rectified-standards/blob/master/PSR-2-R-coding-style-guide-additions.md" \l "commenting-code)

1. Pourquoi utiliser les commentaires :

"Code Tells You How, Comments Tell You Why".

Cette maxime nous explique que les commentaires nous aident à comprendre le code qu’est ce qu’il fait exactement.

On utilise les commentaires pour divers reasons :

* pour que d’autre développeur comprennent notre code
* pour que le code être facile à maintenir

1. Les types des commentaires :

* Les Commentaire en ligne (Inline comment) :

Généralement utilise pour explique une ligne de code

// This is a nice inline comment.

$this->doSomething();

* Les commentaires de block de code (DocBlocks):

Ils sont souvent utilisés pour les méthodes, les classes et les fichiers :

* Les méthodes :

Pour les méthodes le DocBlock doit contenir la description de la méthode ensuite la liste des paramètres enfin le type de retour

*/\*\*  
 \* Return a single brand according to the Id.*

*\* @param number $id it represents the brand Id   
 \* @return \CoreBundle\Entity\Brand   
 \*/***public function** getBrand($id){  
 **return** $this->repository->find($id);  
}

**@param** :

Représente le paramètre de notre méthode si on a plusieurs paramètres on doit le contenir chaqu’un dans une annotation @param

@return : le type de retour de la méthode

@throws : si la fonction retourne des exceptions dans certain cas

* Les classes :

Pour les classes le DocBlock doit contenir la description de cette classe ensuite l’auteur de cette classe

*/\*\*  
 \* BrandService managing the Brand entity.*

*\** ***@author*** *hn <****hn@gmail.com****>  
 \**

**class** BrandService  
{

// ……

}