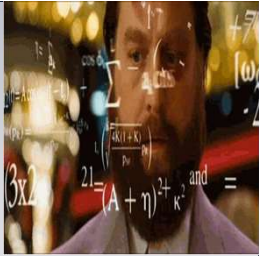




Concepts de base

Exercice



Soit un ensemble d'étudiants.

- 1) Définir une classe Etudiant qui se caractérise par un nom et un ensemble de notes.
- 2) Ajouter un constructeur qui initialise les différents attributs de la classe,
- 3) Ecrire une méthode qui affiche les notes de chaque étudiant,
- 4) Ecrire une méthode qui calcule la moyenne,
- 5) Ecrire une méthode qui affiche si l'étudiant est « admis » ou « non admis ».
- 6) Ecrire une page php permettant de calculer la moyenne et d'afficher le résultat de plusieurs étudiants tester l'exemple pour les deux étudiants suivants :
 - Aymen, 11, 13, 18, 7,10,13,2,5,1
 - Skander, 15, 9, 8, 16

Faite en sorte que lorsque la note est <10 le background soit rouge

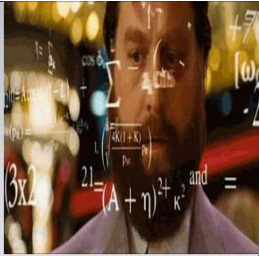
Faite en sorte que lorsque la note est >10 le background soit vert

Faite en sorte que lorsque la note est =10 le background soit orangé



Concepts de base

Exercice



Aymen

11

13

18

7

10

13

2

5

1

Votre moyenne est 8.88888888888889

Skander

15

9

8

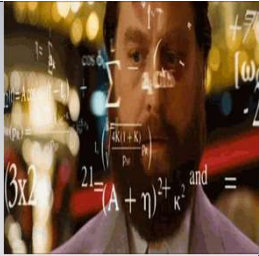
16

Votre moyenne est 12



Concepts de base

Exercice

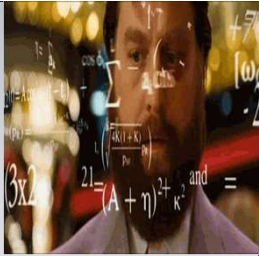


- Nous voulons modéliser une classe pour gérer nos sessions au lieu de passer par la variable `$_SESSION` directement.
- Réfléchissez aux attributs et méthodes que nous pouvons utiliser.
- Implémenter la classe
- Créer une page qui en utilisant les sessions affiche un des messages suivants :
 - Si c'est la première visite, on affiche à l'utilisateur : Bienvenue à notre plateforme.
 - Si c'est la même visite, afficher le message suivant : Merci pour votre fidélité, c'est votre « n » éme visite.
- Ajouter un bouton dans la page permettant de réinitialiser la session.



Héritage

Exercice (1)



Nous allons utiliser l'exemple des Pokémons pour illustrer l'héritage.

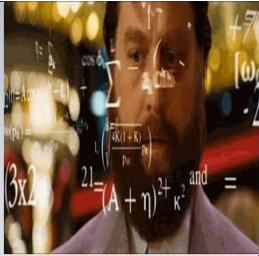
Créer une classe Pokemon qui contient :

- un attribut name qui représente le nom du Pokémon.
- un attribut url qui représente l'url de l'image du Pokémon.
- un attribut hp (pour Health Points) qui représente les points de vie du Pokémon.
- un attribut qui s'appelle attackPokemon qui représente un objet de la classe AttackPokemon qui est constitué de 4 attributs :
 1. attackMinimal qui représente le nombre de points minimal infligé par le pokemon lorsqu'il attaque
 2. attackMaximal qui représente le nombre de points maximal infligé par le pokemon lorsqu'il attaque
 3. specialAttack qui représente le coefficient de l'attaque spéciale du pockémon. Par exemple, si la valeur est de deux donc la force de son attaque sera multiplié par 2
 4. probabilitySpecialAttack qui représente la probabilité sur 100 d'avoir une attaque spéciale. Si la valeur est de 50, le pokemon aura une chance sur deux lors de l'attaque d'un autre pokemon d'effectuer une attaque spéciale



Héritage

Exercice (2)



- un constructeur pour instancier des Pokémons.
- les getters et les setters.
- une méthode isDead() qui retourne un boolean pour indiquer si un Pokémon est mort ($hp \leq 0$) ou non.
- une méthode attack(Pokemon p) qui permet au Pokémon appelant d'attaquer un Pokémon

L'attaque déduit en cas normal atk points de la vie hp du Pokémon attaqué p. En cas d'attaque spéciale, elle déduit $specialAttack * attackPoint$.

- une méthode whoAmI qui affiche les informations du Pokémon.

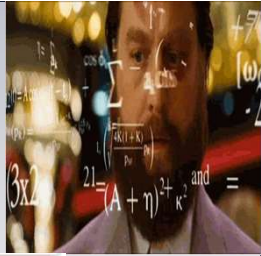
Créer un script qui affiche le scénario de combat entre deux Pokémon. Un combat consiste à une succession de coup entre les deux Pokémon (chacun un coup), jusqu'à la mort d'un des Pokémon.

Le pokemon vainqueur est celui qui a le plus grand niveau de vie qui a gagné





Héritage

Exercise (3)



Les combattants

	
Points : 200	Points : 200
Min Attack Points : 10	Min Attack Points : 30
Max Attack Points : 100	Max Attack Points : 80
Special Attack : 2	Special Attack : 4
Probability Special Attack : 20	Probability Special Attack : 20



Round 1

48	44
----	----

	
Points : 156	Points : 152
Min Attack Points : 10	Min Attack Points : 30
Max Attack Points : 100	Max Attack Points : 80
Special Attack : 2	Special Attack : 4
Probability Special Attack : 20	Probability Special Attack : 20



Round 2

70	52
----	----

	
Points : 104	Points : 82
Min Attack Points : 10	Min Attack Points : 30
Max Attack Points : 100	Max Attack Points : 80
Special Attack : 2	Special Attack : 4
Probability Special Attack : 20	Probability Special Attack : 20



Round 3

41	36
----	----

	
Points : 68	Points : 41
Min Attack Points : 10	Min Attack Points : 30
Max Attack Points : 100	Max Attack Points : 80
Special Attack : 2	Special Attack : 4
Probability Special Attack : 20	Probability Special Attack : 20

Round 4

48	248
----	-----

	
Points : -180	Points : -7
Min Attack Points : 10	Min Attack Points : 30
Max Attack Points : 100	Max Attack Points : 80
Special Attack : 2	Special Attack : 4
Probability Special Attack : 20	Probability Special Attack : 20

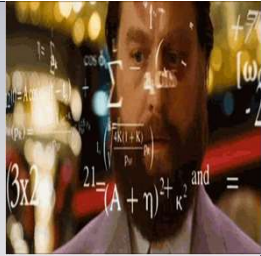
Le vainqueur est





Polymorphisme

Exercice (4)

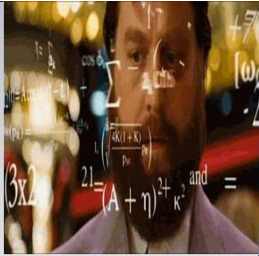


- En plus des Pokémon normaux (décrits à travers la classe Pokemon) on identifie trois types de Pokémon.
- Les Pokémon de type Feu, les Pokémon de type Eau et les Pokémon de type Plante:
- **les Pokémon de type Feu** sont super efficaces contre les Pokémon de type Plante et leur infligent deux fois plus de dégâts ($2 \cdot \text{atk}$). Par contre, ils sont très peu efficaces contre les Pokémon de type Eau ou de type Feu et ne leur infligent que la moitié des dégâts ($0.5 \cdot \text{atk}$). Ils infligent des dégâts normaux aux Pokémon de type Normal.
- **les Pokémon de type Eau** sont super efficaces contre les Pokémon de type Feu et leur infligent deux fois plus de dégâts ($2 \cdot \text{atk}$). Par contre, ils sont très peu efficaces contre les Pokémon de type Eau ou de type Plante et ne leur infligent que la moitié des dégâts ($0.5 \cdot \text{atk}$). Ils infligent des dégâts normaux aux Pokémon de type Normal.
- **Les Pokémon de type Plante** sont super efficaces contre les Pokémon de type Eau et leur infligent deux fois plus de dégâts ($2 \cdot \text{atk}$). Par contre, ils sont très peu efficaces contre les Pokémon de type Plante ou de type Feu et ne leur infligent que la moitié des dégâts ($0.5 \cdot \text{atk}$). Ils infligent des dégâts normaux aux Pokémon de type Normal.
- Créez trois classes PokemonFeu, PokemonEau et PokemonPlante qui héritent de la classe Pokemon.
- Créer des combats entre Pokémon de différents types.



PDO

Exercice



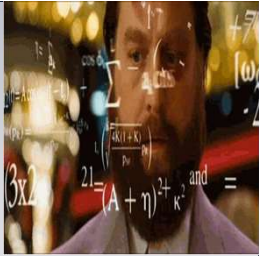
1. Créer une base de données
2. Créer la table « student »
3. Chaque étudiant devra avoir:
 1. Un id qui est sa clé primaire
 2. Un name
 3. Une date de naissance
4. Remplissez manuellement la table avec quelques étudiants
5. Créer une page qui affiche la liste des étudiants

id	name	birthday
1	Aymen	1982-02-07
2	Skander	2018-07-11



PDO

Exercice



1. Devant Chaque étudiant, ajouter un lien détails, en cliquant sur ce lien, rediriger l'utilisateur vers une page détailEtudiant.php affichant les détails de l'étudiant.

id	↑↓	name	↑↓	birthday	
1		Aymen		1982-02-07	i
2		Skander		2018-07-11	i



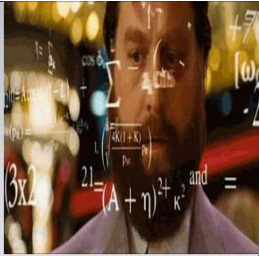
Aymen >

GI
1982-02-07



PDO

Exercice



1. Modifier le code de la page détail afin de récupérer les détails d'un étudiant en utilisant les prepared statements

id	name	birthday	
1	Aymen	1982-02-07	
2	Skander	2018-07-11	



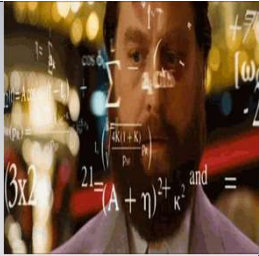
Aymen >

GI
1982-02-07



PDO

Exercice

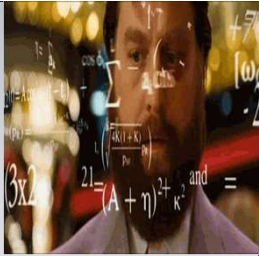


1. Nous voulons implémenter une petite application de gestion des étudiants.
2. La plateforme ne peut être accessible que pour un utilisateur authentifié.
3. Un utilisateur possède les propriétés suivantes :
 1. id
 2. username
 3. email
 4. role
4. Un administrateur peut consulter et gérer (CRUD) les étudiants et les sections
5. Il peut faire la même chose pour les sections
6. Un étudiant a
 1. Id
 2. Name
 3. birthday
 4. Image
 5. section



PDO

Exercice



7. Une section possède un

- id
- Designation
- description

Pour chaque table, penser à créer une classe qui encapsule ses fonctionnalités.

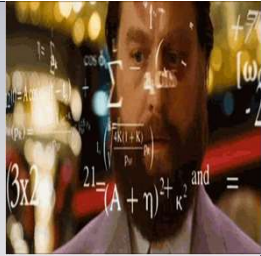
Students Management System Home Liste des étudiants Liste des sections Logout

Hello, PHP LOVERS! Welcome to your administration Platform



PDO

Exercice



La liste est paginée en utilisation une datatable

Vous pouvez exporter la liste en excel, csv et PDF

Vous pouvez filtrer la liste des étudiants par nom (en PHP) ou via le filtre de la datatable

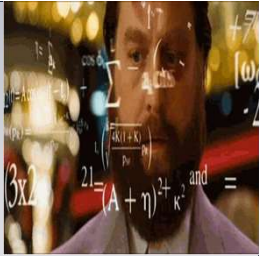
The screenshot shows a web browser at localhost:8000/studentList.php. The page has a navigation bar with links: Home, Liste des étudiants, Liste des sections, and Logout. Below the navigation bar, there's a section titled 'Liste des étudiants'. It includes a search bar with the placeholder 'Veuillez renseigner votre' and a red 'Filtrer' button. Below the search bar are buttons for 'Copy', 'Excel', 'CSV', and 'PDF'. To the right of these buttons is a 'Search:' input field. The main content is a table with columns: id, image, name, birthday, section, and Actions. The table contains two rows of student data. At the bottom, there's a pagination control showing 'Showing 1 to 2 of 2 entries' and buttons for 'Previous', '1' (selected), and 'Next'.

id	image	name	birthday	section	Actions
1		Aymen	1982-02-07	G1	
2		Skander	2018-07-11	G1	



PDO

Exercice



Pour les sections, vous pouvez récupérer la liste des étudiants inscrits dans cette section

Students Management System

[Home](#) [Liste des étudiants](#) [Liste des sections](#) [Logout](#)

Liste des sections

Copy

Excel

CSV

PDF

Search:

id	↑	designation	↑	description	↑	Actions
1		GI		Génie Logiciel		<div>123</div>
2		RT		Réseau et Télécommunication		<div>123</div>

Showing 1 to 2 of 2 entries

Previous

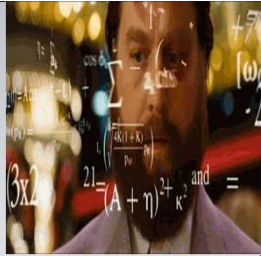
1

Next



PDO

Exercice



Pour un utilisateur avec un rôle user, il peut accéder uniquement en mode lecture.

Liste des étudiants

CopyExcelCSVPDF

Search:

id	image	name	birthday	section	Actions
1		Aymen	1982-02-07	GI	
2		Skander	2018-07-11	GI	

Showing 1 to 2 of 2 entries

Previous

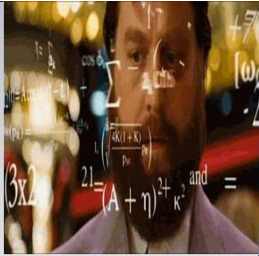
1

Next



PDO

Exercice



1. Créer une classe générique Repository qui offre les fonctionnalités de base d'une base de données suivantes :
 1. findAll qui retourne la liste de tous les enregistrements
 2. findById qui cherche l'enregistrement d'un id
 3. create qui ajoute un enregistrement
 4. delete qui supprime l'enregistrement d'un id

Utiliser ce qu'on a vu en POO

Tester votre code avec la table section et user.