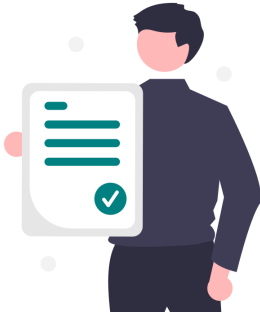


Web avancé – L2 MIASHS

Examen 2023 – Javascript, PHP et AJAX



Règles

- Durée : 1h30
- Moyen : Sur ordinateur
- **Rendu** : par Arche, sous forme de dossier compressé, juste ici :
<https://arche.univ-lorraine.fr/mod/assign/view.php?id=1665301>
- Documents autorisés. Documentation autorisée.
- Toute communication est interdite sous peine de zéro pour triche.
- Modèles génératifs (chatGPT, Bard, etc.) interdits sous peine de zéro pour triche.
- La beauté importe peu. Sauf spécifiquement demandé dans l'énoncé, le CSS n'est pas la priorité de ce cours.

Chaque partie de cet examen est notée sur 10 et compte pour 30 % de la note ! Les 10 % restant servant à évaluer la propreté du code.

Le nombre de piments 🌶️ indique la difficulté estimée.

C'est parti ! 😓

Partie 1 : Javascript (30 % de la note)



Pour commencer :

- Créez un fichier HTML nommé **exam.html** et un fichier Javascript nommé **exam.js**
- Liez le fichier HTML au Javascript
- Faites en sorte que votre fichier HTML affiche "EXAM" dans l'onglet du navigateur

Exercice 1 : (2 points)

Affichez votre nom, prénom et numéro d'étudiant dans la console à partir de variables dédiées créées à cet effet.


Exercice 2 : (2 points)

À partir des variables précédentes, créez un tableau **mesinfos** les contenant. À l'aide du tableau, créez une nouvelle variable qui les regroupe en une seule chaîne de caractères, chaque valeur séparée par un espace. Affichez la dans la console.

Exercice 3 : (2 points)

Créez une fonction qui prend en argument une question. Cette fonction retourne la réponse en fonction de la question qui peut être soit *Quelle différence entre let et var ?*, soit *Le Javascript est habituellement exécuté de quel côté (client ou serveur) ?*. Autrement, la fonction retourne "Maiiis chais pas !".

Utilisez-la pour afficher la réponse à la première question.


 Attention, vos réponses seront prises en compte !

Exercice 4 : DOM et JS : (4 points)

Dans votre fichier **exam.html**, créez un bouton qui permet d'ajouter dans la page un paragraphe contenant l'emoji de votre choix. Ce bouton n'ajoutera ce paragraphe que s'il n'est pas déjà présent dans le DOM.

Cliquer sur le paragraphe ajouté le supprime du DOM.

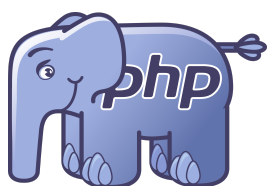
 À vous de bien penser à ce qu'il convient d'ajouter dans les différents fichiers 😊

 le résultat devrait ressembler par exemple à ceci



Ajouter emoji

Partie 2 : PHP (30 % de la note)



Exercice 5 : (2 points)

Créez une page PHP nommée **exam.php**. En son sein, créez un tableau associatif qui contient vos informations suivantes : votre nom, votre prénom, votre âge, votre parcours (miage ou sciences cognitives par exemple). Parcourez le pour afficher ensuite chaque valeur, une par ligne.

Exercice 6 : (3 points)

Affichez désormais les valeurs du tableau associatif, une par ligne, mais en sens inverse.

 Votre logique est à mettre à contribution ici 

Exercice 7 : (1 point)

Affichez dans la page, un paragraphe ayant la réponse à la question suivante :
“Le code PHP est-il activé côté serveur, côté client, ou les deux ?”

Exercice 8 : BDD et PHP (4 points)

Dans phpmyadmin, exécutez le SQL disponible au lien suivant afin de créer une nouvelle base de données :


https://gitlab.com/gguibon/idmc-storage/-/raw/main/web12/sql_requests.txt?inline=false

Dans **exam.php**, créez une fonction qui affiche tous les poèmes de la BDD et exécutez la. Chaque poème étant représenté par son titre en tant que titre de niveau 2, son auteur, puis son contenu en tant que paragraphe.

 Aidez-vous pour cela du SQL suivant : 

Partie 3 : AJAX (30 % de la note)



Pour bien débiter cette partie, créez un fichier nommé **api_exam.php**. À vous de l'utiliser à bon escient. 

Exercice 9 : Afficher un chien aléatoire : (5 points)

Dans **exam.html**, créez un bouton qui affiche une image de chien aléatoire à partir de la BDD. Cette image ne doit pas dépasser 200 pixels de hauteur et prend place dans un

emplacement prévu à cet effet (ainsi chaque clic remplace l'image existante). Choisissez la méthode HTTP qui vous semble appropriée.

💡 Aidez-vous pour cela du SQL suivant :

```
SELECT * FROM chiens ORDER BY RAND() LIMIT 1
```

💡 N'oubliez pas d'ouvrir votre fichier **exam.html** à l'aide de votre serveur (localhost) et non directement !

💡 Vous devriez obtenir un chien aléatoire comme ceci :

Ajouter emoji



Chien aléatoire

📁 Exercice 10 : Ajouter un chien (5 points) 🌶️🌶️🌶️

Créez un formulaire qui permet d'écrire le lien d'une image et de l'ajouter à la table d'images de chiens dans la base de données. Choisissez la méthode HTTP qui vous semble appropriée.

💡 Aidez-vous pour cela du SQL suivant :

```
"INSERT INTO `exam`.`chiens` VALUES( 'https://...lienimage' );"
```

💡 Utilisez par exemple le lien de cette image de chiot :

<https://gitlab.com/gguibon/idmc-storage/-/raw/main/web12/dog5.jpg>

Et voilà, l'examen est fini ! 😊