



Récapitulatif du cours et préparation à l'examen

<https://github.com/heig-vd-progserv-course/heig-vd-progserv1-course>

Visualiser le contenu complet sur GitHub [à cette adresse](#).

L. Delafontaine, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Retrouvez le contenu complet de cette présentation sur GitHub

*Cette présentation est un résumé du contenu complet disponible
sur GitHub.*

*Pour plus de détails, retrouvez le [contenu complet sur GitHub](#) ou en
cliquant sur l'en-tête de ce document.*

Récapitulatif du cours et préparation à l'examen

Vous l'avez fait, bravo ! 🎉

Retrospective

Jetons un coup d'œil sur ce que **vous** avez fait au cours de ce semestre.

Objectifs de l'unité d'enseignement

“ En résumé, vous devriez être capable de :

- Comprendre les bases de PHP et son rôle dans le monde web.*
- Écrire un code PHP propre et organisé.*
- Gérer les formulaires HTML et les données qu'ils contiennent de manière sûre.*
- Persister des données dans une base de données SQLite.*
- Implémenter des concepts de programmation orientée objet.*
- ~~Gérer les cookies et les sessions utilisateurs.~~*

”

Introduction à PHP

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- PHP est un langage de programmation côté serveur, principalement utilisé pour créer des applications web dynamiques.
- Il est intégré dans le HTML et peut interagir avec des bases de données.
- PHP est un langage interprété, ce qui signifie qu'il est exécuté sur le serveur avant d'envoyer le contenu au navigateur.
- La syntaxe de PHP est similaire à d'autres langages de programmation.

Fonctions

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- Les fonctions en PHP permettent de regrouper du code réutilisable.
- Elles peuvent accepter des paramètres et retourner des valeurs.
- Les fonctions peuvent être définies par l'utilisateur ou intégrées dans le langage.
- L'utilisation de fonctions permet d'améliorer la lisibilité et la maintenabilité du code.

Tableaux et boucles

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- Les tableaux en PHP sont des structures de données qui permettent de stocker plusieurs valeurs sous une seule variable.
- Ils peuvent être indexés numériquement ou associativement.
- Les boucles permettent de parcourir les tableaux et d'exécuter du code plusieurs fois.
- Les boucles `for`, `foreach`, `while` et `do...while` sont couramment utilisées en PHP.

Formulaires HTML et validation

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- Les formulaires HTML permettent aux utilisateurs de soumettre des données à un serveur.
- En PHP, les données des formulaires peuvent être récupérées via les superglobales `$_GET` et `$_POST` qui sont des tableaux associatifs.
- La validation des données est essentielle pour garantir la sécurité et l'intégrité des données.
- La validation peut être effectuée côté client (attributs HTML - optionnel) et côté serveur (PHP - obligatoire).

Base de données et PDO

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- Les bases de données permettent de stocker et de gérer des données de manière structurée.
- PDO (PHP Data Objects) est une extension de PHP qui permet d'interagir avec différentes bases de données de manière sécurisée et efficace.
- PDO offre une interface orientée objet pour exécuter des requêtes SQL et récupérer des résultats.

Sécurité et nettoyage des saisies utilisateurs

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- La sécurité est cruciale dans le développement web pour protéger les données et les utilisateurs.
- Les attaques courantes incluent l'injection SQL et les attaques XSS (Cross-Site Scripting).
- Les requêtes préparées avec PDO permettent de prévenir les attaques par injection SQL en séparant le code SQL des données.
- Les fonctions de nettoyage comme `htmlspecialchars()` aident à sécuriser les données affichées dans le navigateur en échappant les caractères spéciaux.

Programmation orientée objet

[Retrouvez ce cours sur le dépôt Git de l'unité d'enseignement.](#)

Éléments clés à retenir

- La programmation orientée objet (POO) est un paradigme de programmation qui organise le code en objets.
- Les objets sont des instances de classes, qui définissent des propriétés et des méthodes.
- La POO permet de structurer le code de manière modulaire et réutilisable.
- PHP supporte les concepts de POO.

Résultats des formulaires de feedback et évaluations GAPS

Discussions et retours sur l'unité d'enseignement.

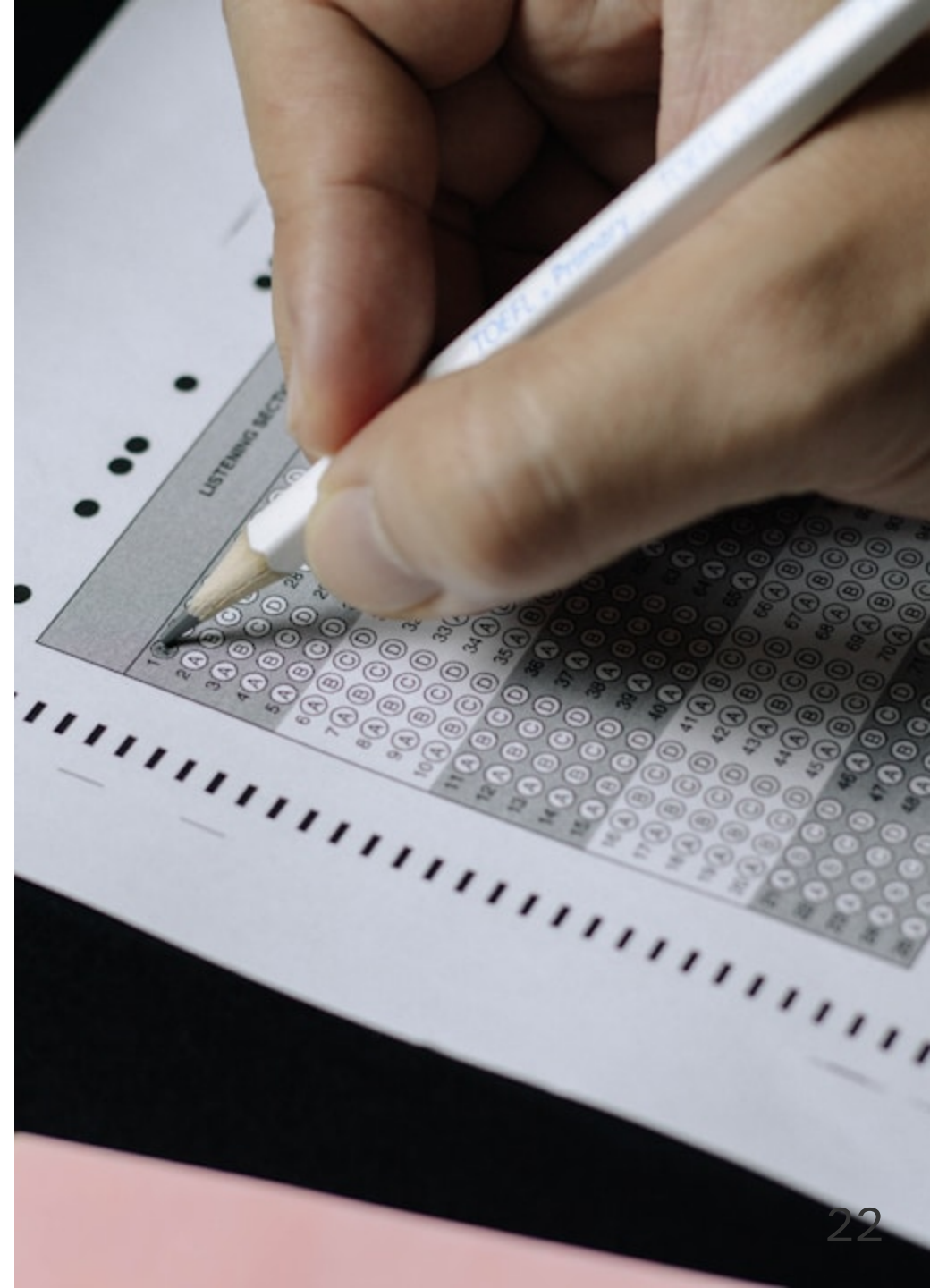
Vous pourrez les retrouver sur le [dépôt Git de l'unité d'enseignement](#).

Préparation à l'examen final

Tout ce que vous devez savoir pour réussir l'examen final.

Examen final

- L'unité d'enseignement se termine par un examen final.
- L'examen aura lieu le **24.06.2025**, de **9h00 à 12h00** dans les locaux communiqués par e-mail.
- **Venez à 8h30 pour vous installer et vous préparer.**
- L'examen est composé de deux parties (théorique + pratique)



Partie théorique

- ~45 minutes.
- 40% de la note.
- Questions ouvertes et QCM avec argumentation possible.
- Portée sur les connaissances théoriques et les exercices.
- Sur la plateforme d'évaluation en ligne de la HEIG-VD (via réseau "*HEIG-VD*" ou VPN)
- Vous gérez votre temps.
- Une fois la partie terminée, nous vous fournissons la consigne de la partie pratique.

Aucune ressource autorisée.

Partie pratique

- ~2 heures 15 minutes.
- 60% de la note.
- Mini-projet à réaliser sur l'ordinateur.
- Portée sur le mini-projet et les exercices.
- Sur l'ordinateur avec votre IDE habituel.
- Rendu à faire **avant la fin de l'examen** par mail.
- **Aucun retard pour le rendu accepté** (= note de 1).

Ressources autorisées : dépôt Git du cours, notes et code personnels, php.net et developer.mozilla.org.

Calcul de la note finale

La note finale de l'examen est calculée en combinant les notes des deux parties de l'examen :

$$\text{Note finale} = \text{Note pour la partie théorique} * 0.4 + \text{Note pour la partie pratique} * 0.6$$

Sanctions possibles en cas de tricherie (1/3)

Source : [Règlement d'études Bachelor de la HEIG-VD.](#)

“ **Art. 16. - Fraude, plagiat et usage de faux**

*1 - Toute fraude, y compris le plagiat ou la tentative de fraude dans les travaux d'évaluation, les examens et le travail de bachelor, **entraîne la note de 1.0 au module**, impliquant la non-acquisition des crédits ECTS correspondants, voire l'invalidation du titre, et peut faire l'objet d'une des sanctions prévues à l'article 14.*

Sanctions possibles en cas de tricherie (2/3)

“ **Art. 14. - Sanctions**

1 - L'étudiant qui ne respecte pas les règles ainsi que les directives ou les consignes de la HEIG-VD, [...] est passible des sanctions disciplinaires suivantes :

1. l'avertissement ;

2. l'exclusion temporaire ;

*3. **l'exclusion de la filière, voire du domaine** si les*

régléments du domaine le prévoient.

Sanctions possibles en cas de tricherie

(3/3)

Les personnes qui surveilleront l'examen sont informées des règles d'évaluation et des sanctions possibles en cas de triche.

Elles sont en charge de surveiller le bon déroulement de l'examen et de signaler les éventuels cas de triche. **Aucune tolérance ne sera accordée en cas de triche avérée ou de doute sur la bonne conduite de l'examen.**

Vous avez travaillé dur pour arriver jusqu'ici et nous avons confiance en vos capacités à réussir cet examen. **Ne prenez pas le risque de nous décevoir et/ou de vous faire renvoyer**

Comment se préparer ?

- Avoir un environnement de développement fonctionnel (cf. cours Introduction à PHP, plus particulièrement le mini-projet).
- **Désactiver toutes aides externes (Copilot, ChatGPT, etc.).**
- Relire les supports de cours avec leurs objectifs respectifs (utilisez ces objectifs pour valider vos acquis).
- Refaire les mini-projets et les exercices.
- Réaliser l'examen blanc à disposition.

Questions

Est-ce que vous avez des questions ?

À vous de jouer !

- Réaliser l'examen blanc.
- Poser des questions si nécessaire.

Pour le mini-projet ou l'examen blanc, n'hésitez pas à vous entraidez si vous avez des difficultés !



Conclusion

Vous avez réussi ! Félicitations !

Vous pouvez être fier.es de vous ! Tout le monde fait du bon travail ! J'ai eu énormément de plaisir à suivre vos progrès au cours de ce semestre.

Ce cours est maintenant terminé, mais j'espère que vous l'avez apprécié et que vous avez beaucoup appris.

Ce cours fait partie d'un programme plus large, et j'espère que vous serez en mesure d'appliquer ce que vous avez appris dans les prochains cours.

Et maintenant ?

Vous avez acquis des connaissances importantes au cours de cette formation. Utilisez ces connaissances à bon escient. Un grand pouvoir s'accompagne d'une grande responsabilité. Vous pouvez maintenant :

- Approfondir le monde du développement web
- Développer des applications web plus complexes (je vous souhaite que ProgServ2 permette cela)

Apprenez toujours plus, et n'hésitez à partager vos connaissances avec les autres.

Remarques finales

Mes remarques finales personnelles sont les suivantes :

- **Posez-vous toujours les bonnes questions** : *pourquoi ? est-ce que je fais la bonne chose pour la bonne cause ?*
- **Faites toujours ce qui est bon pour vous** : la santé, les relations, les ami.es sont plus important.es que le travail !
- **Ayez toujours confiance en vous et en vos tripes** : faites ce que vous pensez être juste !
- **Aidez les autres, soyez gentil** : la coopération vaut mieux que la compétition !
- **Restez critique** : votre opinion compte et peut faire la différence

Merci, bonne continuation et à la prochaine
!

Apéro time! 🎉

Examen final

Déroulement

Partie théorique

- ~45 minutes.
- Lien pour la plateforme en ligne envoyé par mail.
- Une fois la partie théorique terminée, vous recevez la partie pratique.
- **Aucune ressource autorisée.**

Partie pratique

- ~2 heures 15 minutes.
- Rendu à faire avant la fin de l'examen par mail (sinon 1).
- **Ressources autorisées :**
 - [dépôt Git du cours](#), notes et code personnels, [php.net](#) et [developer.mozilla.org](#).

Sanctions possibles en cas de tricherie (1/3)

Source : [Règlement d'études Bachelor de la HEIG-VD.](#)

“ **Art. 16. - Fraude, plagiat et usage de faux**

*1 - Toute fraude, y compris le plagiat ou la tentative de fraude dans les travaux d'évaluation, les examens et le travail de bachelor, **entraîne la note de 1.0 au module**, impliquant la non-acquisition des crédits ECTS correspondants, voire l'invalidation du titre, et peut faire l'objet d'une des sanctions prévues à l'article 14.*

Sanctions possibles en cas de tricherie (2/3)

“ **Art. 14. - Sanctions**

1 - L'étudiant qui ne respecte pas les règles ainsi que les directives ou les consignes de la HEIG-VD, [...] est passible des sanctions disciplinaires suivantes :

1. l'avertissement ;

2. l'exclusion temporaire ;

*3. **l'exclusion de la filière, voire du domaine** si les*

régléments du domaine le prévoient.

Sanctions possibles en cas de tricherie

(3/3)

Les personnes qui surveilleront l'examen sont informées des règles d'évaluation et des sanctions possibles en cas de triche.

Elles sont en charge de surveiller le bon déroulement de l'examen et de signaler les éventuels cas de triche. **Aucune tolérance ne sera accordée en cas de triche avérée ou de doute sur la bonne conduite de l'examen.**

Vous avez travaillé dur pour arriver jusqu'ici et nous avons confiance en vos capacités à réussir cet examen. Ne prenez pas le risque de nous décevoir et/ou de vous faire renvoyer

Partie théorique

- ~45 minutes.
- Vous recevez la partie pratique une fois cette partie terminée.

Partie pratique

- ~2 heures 15 minutes.
- Rendu à faire avant la fin de l'examen par mail.

15:11:00

Sources

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [Nguyen Dang Hoang Nhu](#) sur [Unsplash](#)
- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)