

Durée : 15 minutes. Ce sujet contient 15 questions sur 4 pages, pour un total de 17 points.

---

Nom :

Prénom :

## 1 Git avancé

1. Quel est l'intérêt du dossier ".git" dans un projet ? (1 réponse possible) (1 point)
  - ☐ Il contient les fichiers de configuration du dépôt local.
  - ☐ Il stocke les modifications apportées aux fichiers du projet.
  - ☐ Il conserve l'historique des modifications et les métadonnées du dépôt git
  - ☐ Il est utilisé pour télécharger les dépendances externes du projet.
2. Quels sont les fichiers que l'on souhaite versionner dans un dépôt git ? (plusieurs réponses) (1 point)
  - ☐ Les fichiers de code source.
  - ☐ Les fichiers générés lors de la compilation du code source.
  - ☐ Les fichiers de configuration spécifiques à l'utilisateur
  - ☐ Les fichiers relatifs aux dépendances nécessaires pour lancer le code du projet .
  - ☐ Les fichiers sensibles contenant des informations d'identification.
  - ☐ Les fichiers de logs et fichiers générés a la suite d'un traitement du projet.
  - ☐ Le fichier .gitignore
3. Quelles propositions sont vraies? (plusieurs réponses) (1 point)
  - ☐ Git est un système de gestion de version de code centralisé
  - ☐ Je peux utiliser Git sans utiliser de ligne de commande
  - ☐ Je peux utiliser Git sans utiliser de forge logicielle (Github / Gitlab)
  - ☐ Je peux travailler sur plusieurs dépôts distants a partir d'un dépôt Git local

## 2 Portabilité / Configuration

1. Qu'est ce que la portabilité applicative? (1 réponse possible) (1 point)
  - ☐ La capacité d'une application à être utilisée sur différents systèmes d'exploitation sans nécessiter de modifications majeures.
  - ☐ La facilité avec laquelle une application peut être déployée sur différents serveurs sans perturber son fonctionnement.
  - ☐ La mesure dans laquelle une application peut être personnalisée pour répondre aux besoins spécifiques de différents utilisateurs.
  - ☐ La vitesse à laquelle une application peut être mise à jour pour intégrer de nouvelles fonctionnalités sans interruption du service.

2. Qu'est ce que la configuration applicative? (1 réponse possible) (1 point)
- ☐ La capacité d'une application à être utilisée sur différents systèmes d'exploitation sans nécessiter de modifications majeures.
  - ☐ La facilité avec laquelle une application peut être déployée sur différents serveurs sans perturber son fonctionnement.
  - ☐ La mesure dans laquelle une application peut être personnalisée pour répondre aux besoins spécifiques de différents utilisateurs.
  - ☐ La vitesse à laquelle une application peut être mise à jour pour intégrer de nouvelles fonctionnalités sans interruption du service.
3. Quels outils permettent la portabilité applicative en python ? (plusieurs réponses possibles) (1 point)
- ☐ Un fichier de configuration des variables du code projet
  - ☐ Une documentation d'installation de l'application
  - ☐ Un fichier de configuration des dépendances nécessaires au projet
  - ☐ Un mode de packaging incorporant tout le nécessaire de run du projet (type Docker)
  - ☐ Un gestionnaire d'environnement virtuel python anaconda.
  - ☐ Un set de variables d'environnement a charger dans l'application au démarrage.
4. Quel est l'intérêt d'utiliser des environnements virtuels Python ? (plusieurs réponses possibles) (1 point)
- ☐ Isoler les dépendances spécifiques à chaque projet.
  - ☐ Éviter les conflits entre différentes versions de packages.
  - ☐ Faciliter le déploiement et la portabilité des applications.
  - ☐ Permettre aux développeurs de travailler sur plusieurs projets Python avec des configurations différentes sur la même machine.
  - ☐ Faciliter la collaboration en partageant un environnement de développement cohérent.
5. Qu'est ce que le format .whl ou wheel? (1 point)
- ☐ Un format de fichier compressé utilisé pour stocker des données structurées dans un dépôt de données.
  - ☐ Un format de fichier binaire utilisé pour distribuer des packages Python et leurs dépendances.
  - ☐ Un format de fichier de configuration utilisé pour spécifier les dépendances d'un projet Python.
  - ☐ Un format de fichier de documentation utilisé pour générer des pages web statiques à partir de fichiers texte.
6. Qu'est-ce que le PEP (Python Enhancement Proposal) ? (1 point)
- ☐ Un outil de développement utilisé pour gérer les environnements virtuels Python.
  - ☐ Un format de fichier utilisé pour spécifier les dépendances d'un projet Python.
  - ☐ Un processus formel utilisé pour proposer et discuter des améliorations potentielles pour le langage Python.
  - ☐ Un format de fichier de documentation utilisé pour spécifier les bonnes pratiques de développement en Python.

### 3 Automatisation

7. Que permet l'analyse statique du code ? (plusieurs réponses possible) (2 points)
- ☐ Identifier les parties du code nécessitant moins de ressources systèmes.
  - ☐ Identifier les erreurs de syntaxe dans le code source.
  - ☐ Identifier les problèmes de portabilité du code source.
  - ☐ Détecter les erreurs potentielles, les violations de conventions de codage et les mauvaises pratiques de programmation sans exécuter le code.
  - ☐ Transformer le code source dans un langage de bas niveau pour une meilleure performance.
  - ☐ Identifier les vulnérabilités de sécurité potentielles dans le code
8. Que permet l'analyse dynamique du code ? (plusieurs réponses possible) (2 points)
- ☐ Évaluer la consommation et l'utilisation des ressources pendant l'exécution du code.
  - ☐ Identifier les erreurs de syntaxe dans le code source.
  - ☐ Tester le code en le faisant fonctionner et en observant son comportement réel.
  - ☐ Mesurer la couverture de code pour identifier les parties non testées du code.
  - ☐ Détecter les erreurs potentielles, les violations de conventions de codage et les mauvaises pratiques de programmation sans exécuter le code.
  - ☐ Transformer le code source dans un langage de bas niveau pour une meilleure performance.
9. A quoi servent les outils de formatting ? (plusieurs réponses possible) (1 point)
- ☐ Facilitation de la collaboration en limitant les différences de code sur le style.
  - ☐ Identifier les erreurs de syntaxe en amont de l'ajout de fonctionnalités.
  - ☐ Consistance et lisibilité : permettre au code d'être plus facilement assimilé et compris.
  - ☐ Réduire la taille du code.
  - ☐ Détecter les erreurs potentielles, les violations de conventions de codage et les mauvaises pratiques de programmation sans exécuter le code.
10. Qu'est ce que l'intégration continue? (plusieurs réponses possible) (1 point)
- ☐ L'ajout de code en continu sur un dépôt git.
  - ☐ L'exécution de contrôles sur une base de code existante a chaque changement.
  - ☐ Le lancement du projet dans plusieurs environnements avant de valider des changements dans la base de code.

## 4 HTTP

11. Qu'est ce que le HTML? (une reponse possible) (1 point)
- ☐ Un format de fichier qui contient des données et metadonnees d'un site
  - ☐ Un format de fichier qui permet la customisation de l'apparence d'un site
  - ☐ Un langage de programmation
12. Qu'est ce qu'une API webservice? (1 réponse) (1 point)
- ☐ Une application qui traite des demandes utilisateur de manière asynchrone
  - ☐ Une application qui expose des données dans des format documentés pour d'autres applications ?
  - ☐ Une application qui expose des ressources statiques uniquement: images, style.