Page 1 sur 6

GLOSSAIRE / QUIZ

Ce glossaire est prévu pour valider les connaissances après chaque exercice clé de votre formation.

L’idée est de réussir à définir les différentes notions en une à 3 phrases maximum et d’adopter le

Vocabulaire technique approprié. Ce document vous servira comme base de révisions.

A chaque fois qu’un acronyme est cité dans le glossaire, il sera impératif de donner la signification

de chaque lettre initiale.

• Général

• Front-end

• UX / UI

• Programmation orientée objet

• Architecture

• Modélisation / base de données

• Symfony

• Sécurité

• RGPD

• SEO

• Gestion de projets / DevOps

• English

**Général**

1) Quel est l’environnement à installer pour exécuter un script PHP ? Citer 2 exemples de logiciels

Permettant ce contexte.

R : Pour exécuter un script PHP, vous avez besoin d'un environnement de développement qui comprend un serveur web et un interpréteur PHP.

* Laragon.
* Wamp.

Ils comprennent Apache, MySQL et PHP

2) Qu’est-ce qu’un algorithme ?

R : C’est une suite d’instructions permettant de réaliser des actions complexes.

3) Qu’est-ce qu’une variable ? Par quel symbole est préfixée une variable en PHP ?

R : Une Variable est un conteneur qui dispose d’une valeur qui peut être modifier au cours du temps.

En PHP une variable est représentée par le symbole dollar $

4) Qu’est-ce que la portée d’une variable ?

R : Encore appeler le Scope d’une variable, c’est son champ d’action.

5) Qu’est-ce qu’une constante ? Quelle est la différence avec une variable ?

R : Une constante est un conteneur disposant d’une valeur qui ne change pas, la différence avec une variable est que sa valeur reste constante et ne peut être changée.

6) Qu’est-ce qu’une superglobale, combien en existent-ils et donner un exemple d’utilisation

R : Une superglobale est une variable spéciale en PHP qui est accessible de n'importe où dans le script et qui contient des informations importantes.

* $\_GET : Contient les données envoyées dans l’url par la méthode GET d’un formulaire.
* $\_POST : Contient les données envoyées par la méthode POST d’un formulaire.
* $\_SESSION : Permet de stocker des informations persistantes entre différentes pages de l’application, pour maintenir par exemple un utilisateur connecter ou panier d’un e-commerce.
* $\_FILES : Contient des informations sur les fichiers téléchargés via un formulaire.

7) Quels sont les différents types (primitifs) que l’on peut associer à une variable en PHP ? Les citer

et en donner des exemples (ne pas oublier le type d’une variable sans valeur)

R :

* String : pour les chaines de caractères.
* Integer : pour les nombres entiers.
* Float : pour les nombres à virgules.
* Boolean : représente les valeur binaire, true or false ; 1 ou 0.
* Tableau (array) : pour la collection ordonnée d’élément, de même ou différents type;
* Null : Représente l’absence de valeur.

8) Existe-t-il plusieurs types de tableaux en PHP, si oui lesquels ?

R : En PHP il existe plusieurs type de tableaux .

* Les tableaux numériques.
* Les tableaux associatifs.
* Les tableaux multidimensionnels.

9) Quelles sont les différentes structures de contrôles qu’il existe en algorithmie ? Donner un

exemple pour chacune d’entre elles.

R :

* La Structure conditionnelle (IF-ELSE) : permet d’exécuter une instruction si une condition est vrai et une autre si elle est fausse. EX :

If (vrai){

Exécute si vrai

} else {

Exécute si faux

}

* Structure de boucle (FOR) : Permet de répéter un bloc de code un certain nombre de fois, en indiquant une condition de sortie. EX :

FOR ($i = 0 ; $i < 10 ; i++) {

Code à exécuter

}

* La boucle (WHILE) : Permet de répéter un bloc de code tant qu’une condition est vraie La condition est vérifiée avant l'exécution du bloc de code.. EX :

WHILE ($i < 7) {

Code à exécuter ;

$i++ ;

}

* La boucle (DO WHILE) : Permet de répéter un bloc de code tant qu’une condition est vraie. La condition est vérifiée après l'exécution du bloc de code. EX :

DO {

Code à exécuter ;

$i++ ;

}

WHILE ($i < 7)

* Structure de contrôle (SWITCH) : Permet de tester différentes valeurs d'une variable et d'exécuter des blocs de code correspondants en fonction de la valeur.

SWITCH (animal) {

Case «chien » :

Condition ;

Break ;

Case «chat » :

Condition ;

Break ;

Case «souris » :

Condition ;

Break ;

Default :

Condition ;

}

10) Quelle est la fonction PHP permettant de demander la longueur d’une chaîne de caractères ?

R : strlen(‘le mot`) ;

11) Qu’est-ce qu’une session ? Quelle fonction permet de démarrer une session en PHP ? Donner un

Exemple d’utilisation en PHP

R : Une session en PHP permet de maintenir des informations spécifiques à un utilisateur entre différentes pages de l’app ou requêtes http.

12) Qu’est-ce qu’un cookie ? Donner un exemple d’utilisation en PHP

R : Un cookie est un petit fichier texte stocké sur le navigateur d'un utilisateur par un site web. Il est utilisé pour stocker des informations spécifiques à cet utilisateur, telles que des préférences, des informations de connexion, des informations de suivi.

13) Quelle est la différence entre les instructions « require » et « include » en PHP

R : Les deux servent à exécuter le contenu d’un fichier dans un autre.

Avec  « require » : Si le fichier n’est pas trouvé une FATAL ERROR se déclenche et arrête le script

Avec « include » : Si le fichier n’est pas trouvé une erreur de type avertissement est générée mais le script continue de s’exécuter.

14) Comment effectuer une redirection en PHP ?

R : On peut utiliser la fonction header(‘location : page.php’) ; pour envoyer vers une page donnée.

15) Définir la partie « front-end » et « back-end » d’une application

R :

Front-End : C’est la partie de l’application avec laquelle les utilisateurs interagissent directement aussi appelé « coté client », cela inclut IU et UX

Back-End : C’est la partie invisible de l’application qui gère les fonctionnalités et la logique de traitement de données, aussi appeler « coté server ».

16) Définir le contrôle de version ? Qu’est-ce que Git ?

R : Encore appelé gestion de version, est un système qui permet de suivre et de gérer les modifications apportées à un projet.

Git : est l’un des système de contrôle de version les plus populaire.

17) Qu’est-ce qu’un CMS ? Citer au moins 2 exemples

R :

CSM : « Content Management System » ou « Système de gestion de contenu » est une plateforme logiciel qui permet de créer, de gérer et de publier de contenu en ligne sans nécessiter de compétences techniques avancées en programmation ou conception web.

Ex : WordPress, Joomla ; développer en PHP .

Page 2 sur 6

**Front-end**

1) Définir HTML

R : HTML (HyperText Markup Language) est un langage de balisage utilisé pour créer la structure et le contenu des pages web

2) Définir CSS

R : CSS (Cascading Style Sheets) est un langage de style utilisé pour définir la présentation et la mise en page des documents HTML et XML.

3) Définir Javascript

R : JavaScript est un langage de programmation polyvalent et interprété coté client par le navigateur, principalement utilisé pour créer des fonctionnalités interactives et dynamiques sur les sites web.

4) Définir JSON. Dans quel contexte ce format est-il utilisé ?

R :

JSON : JavaScript Objet Notation, est un format de données léger et utilisé pour stocker et échanger des données structurées.

5) Peut-on interpréter du Javascript côté serveur ? Si oui, comment ?

R : Il est possible d’interpréter JavaScript coté server grâce à JSNode

6) Qu’est-ce qu’un sélecteur CSS ?

R : Un sélecteur css est une expression utiliser pour cibler un élément HTML.

7) Quelle balise HTML permet de créer un lien hypertexte ?

R : la balise <a href= ‘’’’> </a>

8) Qu’est-ce qu’une requête AJAX ?

R : une requête AJAX est une méthode de communication entre une page web et un serveur web, permettant de mettre à jour dynamiquement le contenu d'une page sans recharger l'ensemble de la page.

9) Quel sélecteur CSS permet de sélectionner tous les éléments d’une classe spécifique ? D’un

identifiant spécifique ?

R : le sélecteur de classe est le point(.) ; et le sélecteur d’identifiant est dièse (#)

10) Définir le responsive design

R : Le responsive design est un design qui s’adapte à plusieurs tailles d’écran

11) Qu’est-ce que le templating ?

R : le templating est une technique qui facilite la création de pages web dynamiques en utilisant des modèles de données prédéfinis et en remplaçant les espaces réservés par des données spécifiques au moment de la génération de la page.

12) Qu’est-ce qu’une fonction anonyme en Javascript ?

R : En JavaScript, une fonction anonyme est une fonction qui n'a pas de nom, Elles sont généralement définies en ligne et utilisées directement là où elles sont déclarées.

13) Quelle méthode JavaScript est utilisée pour ajouter un élément à la fin d'un tableau ?

R : La méthode Push()

14) Qu’est-ce qu’un « media query » ?

R : Un "media query" (requête média) est une fonctionnalité de CSS qui permet de définir des règles de style en fonction des caractéristiques de l'appareil ou du média sur lequel une page web est affichée. Les media queries permettent par exemple de rendre une app web responsive, adaptative à différents appareils, tailles d'écran et orientations.

15) Qu’est-ce qu’un pseudo élément en CSS ?

R : En CSS les pseudo-éléments sont spécifiés à l'aide de deux points-colons « :: » suivis du nom du pseudo-élément. Elles permettent de styliser un élément HTML, ajouter du contenu sans ajouter de balise HTML

16) Qu’est-ce que Bootstrap ? Donner d’autres exemples équivalent,

R : C’est Framework basé en HTML CSS et JS, qui vise à accélérer le processus de développement web en fournissant des outils et des composants prêts à l’emploi et réutilisables.

17) Quand un formulaire HTML est créé, quelles sont les 2 méthodes qui peuvent lui être associées ?

Donner la différence entre ces 2 méthodes

R : Les deux méthode associé à un formulaire HTML sont

* La méthode GET : les données du formulaire sont incluses dans l’URL.
* La méthode POST : les données du formulaire sont incluses dans le corps de la requête http, ce qui les rendent invisibles dans l’URL.

**UX / UI**

1. Quelle est la différence entre UX Design et UI Design ?

R :

* UX Design : l’expérience utilisateur, permet créer une expérience significative et positive d’une application.
* UI Design : l’interface utilisateur, ce concentre sur l’apparence visuelle d’une application,

1. Qu’est-ce qu’un wireframe ?

R : Un wireframe est une représentation visuelle simple et schématique d'une page web. D’une application mobile ou d’une interface utilisateur. Permet d’esquisser, de planifier la structure et la disposition des éléments.

1. Qu’est-ce qu’un prototype ?

R : Un prototype est une représentation préliminaire d'une application. Il s'agit d'une version simplifiée ou une simulation de l'objet final.

1. Qu’est-ce que la hiérarchie visuelle en UI Design ?

R : c’est l’organisation et la structuration d’une interface utilisateur, pour guider l’utilisateur vers les éléments les plus importants.

EX : Contraste de couleur, taille, Typographie, Disposition.

1. Qu’est-ce que l’accessibilité en UX Design ?

R : c’est une UX qui est ouvert au plus grand nombre personne y compris celle avec des limitations physiques ou des besoin particuliers.

1. Qu’est-ce qu’une grille de mise en page ?

R : Wirframe : est un outil fondamental en design d’interface qui aide à organiser visuellement les éléments, d’une page web, d’une app mobile etc.

Composer de ligne horizontales et verticales qui délimitent la disposition des éléments visuels.

1. Qu’est-ce que la notion d’affordance en UX Design ?

R : c’est la capacité d’un élément de conception à suggérer ou à indiquer naturellement son utilisation ou sa fonction.

1. Qu’est-ce qu’un « mobile first design » ?

R : C’est une approche de conception et d’expérience utilisateur qui privilégie la conception et le développement d’un site web ou d’une application pour les appareils mobiles en premier, avant de concevoir les versions pour les ordinateurs et tablettes.

**Programmation orientée objet (POO)**

1. Donner une définition de la programmation orientée objet

R : la POO est une approche de programmation qui repose sur le concept d’objet, qui sont des unités autonomes regroupant à la fois des donnée (propriétés ou attributs) et des fonctions (méthodes). En POO, le code est structuré autour d’entités autonomes, les objets, qui interagissent entre eux pour réaliser des tâches et des fonctionnalités.

1. Qu’est-ce qu’une classe ? Comment la déclare-t-on ?

R : En POO une classe est un modèle pour créer des objets.

Class Maclasse {

Type Attribut1

Type Attribut2

Type Nom\_de\_la\_méthode1 (parametres) {

Corps de la méthode

}

Type Nom\_de\_la\_méthode1 (parametres) {

Corps de la méthode

}

}

1. Qu’est-ce qu’un objet ?

R : En POO un objet est une instance d’une classe.

1. Définir la notion de propriété / attribut / méthode

R : En POO

* **Une propriété :** est une variable interne à une classe qui stocke une donnée propre à cette classe
* **Une méthode :** est une fonction interne à la classe, qui définit l’action que peuvent avoir les objets de cette classe.

1. Qu’est-ce que la visibilité d’une propriété ou d’une méthode ? Citer les différents styles de visibilité

R : La visibilité d’une propriété ou d’une méthode est la portée de cette dernière par rapport au reste élément du code. Il en existe 3 principales visibilités

- **public** : Ces éléments sont accessibles depuis n’importe où dans le code.

- **Protected** : Ces élément sont accessibles depuis l’intérieur de la classe qui la définit, ainsi que depuis les classes dérivées (sous-classes) de cette classe.

- **Private** : Ces éléments sont accessibles depuis l’intérieur de la classe qui la définit. Ils ne sont pas accessibles depuis l’extérieur de la classe ni des sous-classes.

6) Quelle est la méthode spécifique utilisée pour créer un nouvel objet à partir d’une classe ?

R : La méthode constructeur (en PHP la méthode magique \_construc().)

1. Qu’est-ce que l’encapsulation ?

R : L’encapsulation c’est la gestion de la visibilité d’une variable au sein d’une classe, c’est l’un des quatre piliers de la POO, les trois autres étant **l’abstraction, l’héritage et le polymorphisme**.

1. Que signifie « étendre une classe » ? Quelle est le concept clé mis en œuvre ? Donner un exemple.

R : Etendre une classe est le fait de créer une sous-classe (classe fille) qui hérite des propriétés de la classe mère tout en ayant ses propres spécificités. C’est le principe de l’héritage.

1. Définir l’opérateur de résolution de portée

R : L’opérateur de résolution de porté est utilisé pour accéder aux membre (méthodes et variables) d’une classe depuis l’intérieur ou l’extérieur de cette classe représenté (selon les langages) par un « point » ou « une flèche ».

- Depuis l’intérieur de la classe on utilise le mot **this** suivi de l’opérateur exemple (en php) $this->maPropriete.

- Depuis l’extérieur de la classe on utilise le nom de l’objet suivie de l’opérateur exemple monObjet->maMzthode().

10) Définir une méthode / propriété statique.

R : Une propriété ou méthode statique est une propriété qui est associé à la classe elle-même plutôt qu’à une instance de cette classe.

11) Définir le polymorphisme en POO

R : Le polymorphisme se traduit par la capacité d’un objet à prendre différentes formes en fonction du contexte.

Il existe deux principaux types de polymorphisme :

- **Polymorphisme de sous-types** (ou polymorphisme d’héritage)

- **Polymorphisme d’interface**.

12) Définir une méthode / classe abstraite ?

R : Une classe abstraite est une classe qui ne peut pas être instanciée directement, c'est-à-dire que vous ne pouvez pas créer d'objets à partir de cette classe. Elle sert de modèle ou de base d’autres classes, et définit un ensemble de méthodes ou de propriétés abstraites qui doivent être implémenter par les classes dérivées.

13) Définir le chaînage de méthodes

R : Est un concept de conception de programmation qui permet d'appeler plusieurs méthodes d'un objet de manière séquentielle, en chaîne, sur une seule ligne de code.

14) Qu’est-ce que la méthode \_\_toString() ? Existe-t-il d’autres méthodes « magiques »

R : La méthode magique **\_\_toString()** est utilisée pour convertir un objet en une chaine de caractères ; il en existe d’autre ;

**\_\_construct():** Le constructeur, utilisé pour initialiser un objet lorsque vous créez une nouvelle instance de classe.

**\_\_destruct():** Le destructeur, utilisé pour effectuer des opérations de nettoyage ou de libération de ressources lorsque l'objet est détruit.

**\_\_get() et \_\_set():** Utilisées pour accéder (lecture) et modifier (écriture) les propriétés d'un objet de manière dynamique.

**\_\_isset() et \_\_unset():** Utilisées pour vérifier si une propriété existe et pour la supprimer de manière dynamique.

**\_\_call()**: Utilisée pour appeler une méthode qui n'existe pas dans la classe.

**\_\_callStatic()**: Utilisée pour appeler une méthode statique qui n'existe pas dans la classe.

15) Qu’est-ce qu’un « autoload » ?

R : L'autoload est un concept couramment utilisé en programmation, en particulier dans le contexte des langages de POO, pour charger automatiquement des classes ou des fichiers lorsque cela est nécessaire.

16) Comment appelle-t-on en français les « getters » et les « setters » ?

R : getter = accesseur ; setter = mutateur.

17) Qu’est-ce que la sérialisation en PHP ?

R : La sérialisation en PHP est le processus de conversion d'une structure de données, telle qu'un objet, un tableau ou un ensemble de variables, en une représentation de données qui peut être stockée dans un fichier, transmise sur un réseau ou utilisée à d'autres fins de stockage ou de transmission de données.

**serialize()**: Cette fonction prend une structure de données PHP (comme un objet, un tableau ou un ensemble de variables) en entrée et renvoie une chaîne de caractères représentant cette structure de données sous une forme sérialisée. La chaîne sérialisée peut ensuite être stockée dans un fichier ou transmise sur un réseau.

**unserialize():** Cette fonction prend une chaîne de caractères sérialisée en entrée et la convertit en une structure de données PHP valide. Cela signifie que vous pouvez reconstituer l'objet, le tableau ou les variables à partir de la chaîne sérialisée.

Ex :

// Structure de données à sérialiser

$data = ['nom' => 'John', 'age' => 30, 'email' => 'john@example.com'];

// Sérialisation

$serializedData = serialize($data);

// Stockage dans un fichier

file\_put\_contents('data.txt', $serializedData);

// Récupération et désérialisation

$loadedData = file\_get\_contents('data.txt');

$unserializedData = unserialize($loadedData);

Page 3 sur 6

**Architecture**

1) Qu’est-ce que l’architecture client / serveur ? Grâce à quel type de requête peut-on interroger le

serveur. Définir l’acronyme de ce type de requête. Si on ajoute un « S » à cet acronyme, expliquer

la différence

2) Donner la définition d’un design pattern. Citer au moins 3 exemples de design pattern

3) Qu’est-ce que l’architecture MVC ?

4) Quel est le rôle de chaque couche du design pattern MVC : Model, View, Controller ?

5) Quels sont les avantages de l’architecture MVC ?

6) Existe-t-il des variantes à l’architecture MVC ?

7) Qu’est-ce qu’une API ? Définir l’architecture REST

Modélisation / Base de données

1) Qu’est-ce que la modélisation de données ? Définir la méthode Merise

2) Quelles sont les 3 étapes principales de la méthode Merise ?

a. Analyse, conception et réalisation

b. Planification, exécution et contrôle

c. Création, modification et suppression

3) Qu’est-ce qu’un modèle conceptuel de données (MCD) en Merise ?

4) Qu’est-ce qu’un modèle logique de données (MLD) en Merise ?

5) Donner la définition des mots suivants :

a. Entité

b. Relation

c. Cardinalité

d. Clé primaire / clé étrangère

6) Que devient une relation de type « Many To Many » dans le modèle logique de données ?

7) Qu’est-ce qu’une base de données ?

8) Définir les notions suivantes :

a. SQL

b. MySQL

c. SGBD (donner 2 exemples de SGBD)

9) Dans une base de données, les données sont stockées dans des \_\_\_. Celles-ci sont constituées de

lignes appelées \_\_\_ et de colonnes appelées \_\_\_

10) Quelle est la différence entre une base de données relationnelle et non relationnelle ?

11) Qu’est-ce qu’une jointure dans une base de données ? En existe-t-il plusieurs ? Si oui lesquelles ?

12) A quoi sert une vue dans une base de données ?

13) Qu’est-ce que l’intégrité référentielle dans une base de données ?

14) Quelles sont les fonctions d’agrégation en SQL ?

15) Qu’est ce qu’un CRUD dans le contexte d’une base de données ?

16) Quelles sont les clauses qui permettent de :

a. Insérer un nouvel enregistrement dans une table

b. Modifier un enregistrement dans une table

c. Supprimer un enregistrement dans une table

d. Supprimer la base de données

e. Filtrer les résultats d’une requête SQL

f. Trier les résultats d’une requête SELECT

g. Regrouper les résultats d'une requête SELECT en fonction d'une colonne spécifique

h. Concaténer 2 chaînes de caractères

17) Comment se connecter à une base de données en PHP ? Quelle est la classe native utilisée ?

Page 4 sur 6

**Symfony**

1) Qu’est-ce que Symfony ?

2) Sur quel langage de programmation et design pattern repose Symfony ?

3) Quelle est la dernière version en date de Symfony ?

4) Qu’est-ce qu’un bundle ?

5) Quel est le moteur de template utilisé par défaut dans Symfony ?

6) Qu’est-ce qu’un ORM ? Quel est son utilité et comment s’appelle-t-il au sein de Symfony ?

7) Qu’est-ce que l’injection de dépendances ? Quel est l’outil utilisé dans ce contexte et quel fichier

contient l’intégralité des dépendances du projet ?

8) Que permet le bundle Maker au sein de Symfony ?

9) Quel est le langage de requêtage exploité au sein d’un projet Symfony ?

10) Quel est le composant qui garantit l’authentification et l’autorisation des utilisateurs ?

Sécurité

1) Qu’est-ce que l’injection SQL ? Comment s’en prémunir ?

2) Qu’est-ce que la faille XSS ? Comment s’en prémunir ?

3) Qu’est-ce que la faille CSRF ? Comment s’en prémunir ?

4) Définir l’attaque par force brute et l’attaque par dictionnaire

5) Existe-t-il d’autres failles de sécurité ? Citer celles-ci et expliquer simplement leur comportement

6) A quoi servent l’authentification et l’autorisation dans un contexte d’application web ?

7) Définir la notion de hachage d’un mot de passe et citer des algorithmes de hachage

8) Qu’est-ce qu’une politique de mots de passe forts ?

9) Qu’est-ce que l’hameçonnage ?

10) Définir la « validation des entrées »

**RGPD**

1) Qu’est-ce que le RGPD ?

2) Quel est son objectif principal ?

3) Quelle est la date d’entrée en vigueur du RGPD ?

4) Quelles sont les sanctions possibles en cas de non-respect du RGPD ?

5) En France, quel est l’autorité administrative qui s’occupe de faire appliquer le RGPD ?

6) Quel est le consentement valide selon le RPGD ?

7) Qu’est-ce qu’une politique de confidentialité ?

8) Quelle est la durée de conservation maximale des données personnelles selon le RGPD ?

9) Quels sont les droits des utilisateurs selon le RGPD ?

10) Qu’est-ce que le principe de minimisation des données selon le RGPD ?

Page 5 sur 6

**SEO**

1) Qu’est-ce que le SEO ?

2) Quel est l’objectif principal du SEO ?

3) Existe-t-il plusieurs types de référencement ? Lesquels ?

4) Qu’est-ce que la densité de mots-clés en SEO ?

5) Qu’est-ce qu’une balise « alt » ?

6) Qu’est-ce que la balise « meta description » ?

7) Qu’est-ce que le « nofollow » en SEO ?

8) Quelle est l'importance du contenu de qualité pour le référencement d'un site web ?

9) Pourquoi est-il important d'utiliser des balises de titre (h1, h2, h3, etc.) de manière structurée ?

10) Quelle est la recommandation pour les URL d'un site web bien référencé ?

11) Qu'est-ce que le maillage interne et pourquoi est-il important pour le référencement ?

12) Qu'est-ce que l'optimisation des images pour le référencement ?

13) Qu'est-ce qu'un plan de site (sitemap) et pourquoi est-il important pour le référencement ?

**Gestion de projets / DevOps**

1) Qu’est-ce que la gestion de projet ?

2) Qu’est-ce qu’une méthode Agile de gestion de projet ?

3) Expliquer la méthode MoSCoW en quelques lignes et citer ses avantages

4) A quoi sert la méthodologie MVP ? Citer les caractéristiques clés

5) Qu’est-ce que la planification itérative ?

6) Citer 3 méthodes Agiles dans le cadre d’un projet informatique

7) Qu’est-ce qu’une réunion de revue de projet ?

8) Qu’est-ce qu’un livrable dans un projet ?

9) Quels sont les 3 piliers SCRUM ? Définir chacun d’entre eux

10) Qu’est-ce que le DevOps et quel est son objectif principal ?

11) Qu’est-ce que l’intégration continue ?

12) Qu’est-ce que Docker ? Et en quoi est-il utile dans le cadre du DevOps ?

13) Qu’est-ce qu’un test unitaire ?

14) Quelle est l'unité de code testée lors d'un test unitaire ?

15) Quelles sont les caractéristiques d'un bon test unitaire ?

16) Qu'est-ce qu'une assertion dans un test unitaire ?

Page 6 sur 6

**English**

1) What does JavaScript enable you to do on a website ?

a. Add interactive behavior and dynamic content

b. Define the layout and design of web pages

c. Handle server-side operations

2) Which programming language is primarily used for server-side web development ?

a. PHP

b. JavaScript

c. HTML

3) What is the purpose of a web browser ?

a. To render and display web pages

b. To execute serve-side code

c. To manage databases

4) What is the difference between GET and POST methods in HTTP ?

a. GET retrieves data from a server, while POST submits data to a server

b. GET submits data to a server, while POST retrieves data from a server

c. GET and POST methods are interchangeable

5) What is the purpose of version control systems (e.g., Git) in web development ?

a. To track changes and manage collaborative development

b. To optimize website loading speed

c. To handle server-side scripting

6) What is the purpose of a framework in web development ?

a. To provide a structured environment for building web applications

b. To handle network protocols and data transfer

c. To create visual designs and layouts for websites

7) What does NoSQL stand for ?

a. Not Only SQL

b. Non-Structured Query Language

c. New Object-Oriented Language

8) Which of the following is a characteristic of NoSQL databases ?

a. Strict schema enforcement

b. Support for complex transactions

c. Scalability and flexible data models