

# Réponses aux Questions de Devoirs

## Question 1: Langages de programmation

Citez au moins 5 langages que vous connaissez :

### JavaScript

Création: 1995

Créateur: Brendan Eich

Domaine: Développement web, applications

### Python

Création: 1991

Créateur: Guido van Rossum

Domaine: IA, data science, web

### Java

Création: 1995

Créateur: Sun Microsystems

Domaine: Applications d'entreprise

### C++

Création: 1985

Créateur: Bjarne Stroustrup

Domaine: Systèmes, jeux

### Go

Création: 2009

Créateur: Google

Domaine: Services cloud, microservices

## Question 2: Comparaison de langages

### JavaScript vs Python:

Syntaxe

**JavaScript:** Syntaxe C-like, accolades obligatoires

**Python:** Syntaxe basée sur l'indentation, plus lisible

Facilité d'apprentissage

**JavaScript:** Courbe d'apprentissage moyenne

**Python:** Très facile pour débiter, syntaxe intuitive

Utilisation en entreprise

**JavaScript:** Omniprésent (front-end, back-end, mobile)

**Python:** Dominant en IA, data science, automatisation

Avantages

**JavaScript:** Écosystème vaste, performance, flexibilité

**Python:** Lisibilité, bibliothèques scientifiques, rapidité de développement

## Questions 3-5: Concepts avancés

### 3. Qu'est-ce qu'un framework ?

Un framework est une structure logicielle qui fournit une base et des outils pour développer des applications plus efficacement.

Exemples web:

- **React:** Bibliothèque JavaScript pour interfaces utilisateur
- **Django:** Framework Python full-stack pour développement web rapide

### 4. Langage compilé vs interprété

**Compilé (C++):** Le code source est traduit en code machine avant l'exécution. Plus rapide à l'exécution, optimisé.

**Interprété (Python):** Le code est lu et exécuté ligne par ligne. Plus flexible, développement plus rapide.

### 5. Bibliothèques en programmation

Une bibliothèque est un ensemble de fonctions pré-écrites réutilisables.

**Exemples concrets:** NumPy (calculs scientifiques Python), Lodash (utilitaires JavaScript)

## Question ouverte 6: Spécialisation en programmation

### Recommandation: Maîtriser un seul langage d'abord

Il est préférable de maîtriser parfaitement un langage avant d'en apprendre d'autres. Cela permet de:

- Comprendre en profondeur les concepts de programmation
- Développer une logique solide et des bonnes pratiques
- Être plus efficace et productif dans ce langage
- Faciliter l'apprentissage de nouveaux langages par la suite

Une fois un langage maîtrisé, les concepts acquis se transfèrent facilement vers d'autres langages.

## Questions 1-2: Modèle OSI

### 1. Qu'est-ce que le modèle OSI ?

Le modèle OSI (Open Systems Interconnection) est un modèle conceptuel qui standardise les fonctions de communication d'un système de télécommunication.

- **À quoi sert-il ?** Faciliter l'interopérabilité entre différents systèmes
- **Organisation:** ISO (International Organization for Standardization)
- **Année:** Développé dans les années 1980

### 2. Les 7 couches du modèle OSI

#### Couche 7

Interface utilisateur, services réseau (HTTP, FTP, SMTP)

**Exemple:** Navigateur web, client email

#### 6. Présentation

#### Couche 6

Chiffrement, compression, traduction de données

**Exemple:** SSL/TLS, JPEG, ASCII

#### 5. Session

## Couche 5

Gestion des sessions, connexions

**Exemple:** SQL sessions, NetBIOS

### 4. Transport

## Couche 4

Transport fiable des données

**Exemple:** TCP, UDP

### 3. Réseau

## Couche 3

Routage, adressage logique

**Exemple:** IP, routeurs

### 2. Liaison

## Couche 2

Contrôle d'accès au support, détection d'erreurs

**Exemple:** Ethernet, Wi-Fi, switches

### 1. Physique

## Couche 1

Transmission des bits, support physique

**Exemple:** Câbles, hubs, signaux électriques

## Questions 3-4: OSI vs TCP/IP

### 3. Différence OSI vs TCP/IP

Nombre de couches

**OSI:** 7 couches (modèle théorique)

**TCP/IP:** 4 couches (modèle pratique)

Niveaux d'abstraction

**OSI:** Couches Application, Présentation, Session

**TCP/IP:** Regroupées en une seule couche Application

Utilisation actuelle

**OSI:** Référence théorique pour l'enseignement

**TCP/IP:** Standard de facto d'Internet

#### **4. Exemples pratiques en entreprise**

**Situation 1:** Problème de connectivité réseau

L'approche OSI aide à diagnostiquer méthodiquement chaque couche

**Situation 2:** Configuration firewall d'entreprise

La compréhension des couches permet de bloquer le trafic aux bons niveaux

#### **Questions 5-6: Applications pratiques**

##### **5. Association couches OSI - équipements**

**Couche 1 (Physique):** Hub, câble Ethernet

**Couche 2 (Liaison):** Switch, pare-feu, navigateur web

**Couche 3 (Réseau):** Routeur

##### **6. Envoi d'e-mail selon OSI**

**Explication simple du processus:**

1. **Application:** L'utilisateur écrit l'e-mail dans le client
2. **Présentation:** Chiffrement et formatage du message
3. **Session:** Établissement de la session SMTP
4. **Transport:** TCP segmente et garantit la livraison
5. **Réseau:** IP route vers le serveur de destination
6. **Liaison:** Ethernet encapsule les trames
7. **Physique:** Transmission électrique sur le réseau

#### **Questions bonus: Adressage IP et serveurs**

**Adresse IP publique vs privée**

**IP Publique:** 203.0.113.42 (accessible depuis Internet)

**IP Privée:** 192.168.1.100 (réseau local uniquement)

### **Types de serveurs**

- **Web:** Apache, Nginx
- **Base de données:** MySQL, PostgreSQL

### **Protocoles de communication réseau**

HTTP/HTTPS

Navigation web sécurisée

FTP/SFTP

Transfert de fichiers

DNS