

## IA et Quantique

Référence : **PYIA004**

Durée : **1 jour (7 heures)**

Certification : **Aucune**

### CONNAISSANCES PRÉALA

Connaissances de base de la Programmation Python  
Avoir des notions de base en probabilités  
Avoir une curiosité pour les mathématiques et la physique

### PROFIL DES STAGIAIRES

Développeurs, data scientists, ingénieurs logiciels souhaitant découvrir l'informatique quantique par la pratique  
Professionnels curieux des nouvelles technologies quantiques et de leur potentiel concret

### OBJECTIFS

Comprendre les différences clés avec l'informatique classique  
Comprendre intuitivement les concepts de qubit et superposition d'états  
Créer et exécuter des circuits quantiques sur simulateur et matériel réel  
Évaluer les possibilités et limitations actuelles de l'informatique quantique

### CERTIFICATION PRÉPARÉE

Aucune

### MÉTHODES PÉDAGOGIQUES

Mise à disposition d'un poste de travail par stagiaire  
Remise d'une documentation pédagogique papier ou numérique pendant le stage  
La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions  
Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

### FORMATEUR

Consultant-formateur expert IA

### MÉTHODES D'ÉVALUATION DES ACQUIS

Auto-évaluation des acquis par le stagiaire via un questionnaire  
Attestation des compétences acquises envoyée au stagiaire  
Attestation de fin de stage adressée avec la facture

### CONTENU DU COURS

#### 1. Le générateur aléatoire quantique

Premier contact avec Qiskit : Installation et configuration de l'environnement (Jupyter + Qiskit). Premier circuit quantique :  $|0\rangle \rightarrow H \rightarrow$  mesure. Observation des résultats : vraiment 50/50 ?

- 💡 Générateur de nombres aléatoires quantique simple
- 💡 Comparaison avec `random.choice()` - même résultat, même mécanisme ?

#### 2. Du bit au qubit : le qubit comme unité de calcul

Analogie avec le bit classique  
Opérations de base  
États  $|0\rangle$ ,  $|1\rangle$  et la superposition d'états  
Visualisation sur la sphère de Bloch avec Qiskit  
La mesure quantique : effondrement de la fonction d'onde

- 💡 Visualiser les états avec `plot_bloch_multivector()`
- 💡 Expérimenter avec différents angles de rotation

### 3. Introduction aux portes quantiques, porte de Hadamard : créer la superposition parfaite

Portes X, Y, Z : rotations sur la sphère de Bloch

Composition de portes dans Qiskit

💡 *Créer des superpositions avec différentes amplitudes*

💡 *Visualiser les circuits avec `circuit.draw('mpl')`*

### 4. Circuits multi-qubits et intrication

Passage à deux qubits

Systèmes à deux qubits, espace d'états en 4 dimensions

Porte CNOT : création d'intrication

États de Bell :  $|00\rangle + |11\rangle$  et ses variants

💡 *Créer et mesurer les 4 états de Bell*

💡 *Mesurer les corrélations quantiques*

### 5. Manipulation avancée : portes contrôlées : CNOT, CZ, Toffoli

Circuits réversibles : propriété fondamentale du quantique

💡 *Construire une porte SWAP avec des CNOT*

💡 *Téléportation quantique (circuit complet)*

### 6. Transition vers le matériel réel

Plateformes physiques : ions piégés, qubits supraconducteurs, photons, avantages et inconvénients

Différences entre simulation et matériel réel : bruit et décohérence

Architecture des machines Rigetti

💡 *Même circuit sur simulateur local et machine Rigetti*

💡 *Observer et quantifier les effets du bruit*

### 7. Algorithme de Simon

Problème de Simon : motivation et enjeux

Trouver la période cachée  $s$  dans  $f(x) = f(x \oplus s)$

Interférences pour révéler la structure

💡 *Cas simple  $s = "11"$*

### 8. Construction progressive de l'algorithme, implémentation de $f(x)$ $f(x \oplus s)$

Registres d'entrée et de sortie

Interférences quantiques : amplifier les bonnes réponses

💡 *Implémenter l'oracle pour  $s = "11"$*

💡 *Circuit complet Simon  $s = "11"$  sur simulateur*

### 9. Passage à la complexité réelle : $s = "1111"$

Scaling à 4 qubits : 16 possibilités classiques

Construction de l'oracle 4 qubits

Extraction de l'information : résolution du système linéaire

💡 *Algorithme de Simon complet  $s = "1111"$  sur simulateur*

💡 *Analyser les résultats et probabilités de mesure*

### 10. Validation sur matériel quantique réel : adaptation pour les contraintes matérielles Rigetti

Gestion du bruit et stratégies de mitigation

Optimisation du circuit (profondeur, connectivité)

💡 *Simon  $s = "1111"$  sur machine quantique physique*

💡 *Comparaison quantitative simulateur / matériel réel*

## 11. Bilan et perspectives

Analyse des résultats obtenus

Avantage quantique observé : 16 essais classiques contre 4-5 mesures quantiques

Impact du bruit sur l'algorithme : succès / échecs

Techniques de correction d'erreur

💡 *Mesurer précisément l'impact du bruit*

## 12. Écosystème quantique actuel : plateformes disponibles (IBM, Google, Rigetti, IonQ)

Langages et frameworks (Qiskit, Cirq, Forest)

Applications prometteuses à court terme

Ressources pour continuer : projets pour approfondir

Communautés et formations avancées. Veille technologique

💡 *Créer son propre défi quantique*

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:0171197030) ou par mail à [referent.handicap@edugroupe.com](mailto:referent.handicap@edugroupe.com) pour vos éventuels besoins d'aménagements, afin de vous offrir la meilleure .