

SYLLABUS OFFICIEL

Examen ITS Python

Programmation Python
(IT Specialist niveau fondations)

Certification : IT Specialist: Python

Niveau : Foundation / Entry-level | Public : Étudiants / Développeurs juniors / Data analysts débutants

1. Présentation de la certification

L'examen **IT Specialist: Python (ITS Python)**, délivré par **Pearson** via Certiport, valide les compétences fondamentales en la programmation Python : syntaxe, structures de données, fonctions et général coding. Cette certification atteste de votre capacité à maîtriser les concepts et pratiques essentiels du domaine, selon les bonnes pratiques de l'industrie.

La réussite de cet examen unique conduit à l'obtention de la certification **IT Specialist en Python**, reconnue internationalement par les employeurs et les institutions éducatives. Les ITS sont des certifications **entry-level** posées comme première brique du parcours IT, souvent validées avant les certifications constructeurs (Microsoft, Cisco, AWS, Google).

Informations clés

Code de l'examen	ITS Python (ITS-Python)
Intitulé officiel	IT Specialist: Python
Certification obtenue	IT Specialist en Python
Technologie ciblée	Python 3.x
Éditeur officiel	Pearson (Pearson VUE / Certiport)
Centre de test	Certiport (Pearson VUE) — OpenCertif est centre Certiport autorisé
Niveau	Foundation / Entry-level
Programme	IT Specialist (ITS) — certifications industry-recognized
Format de l'examen	QCM scenarios + drag-and-drop + matching items + hot spot questions
Durée de l'examen	50 minutes
Nombre de questions	Environ 40 questions
Score requis	700 sur 1000 (≈ 70 %)
Prérequis recommandé	Environ 150 heures de pratique (recommandation Pearson / Certiport)
Langue de l'examen	Anglais (autres langues selon disponibilité régionale)

Âge minimum recommandé	13 ans et plus
Validité de la certification	Permanente sur la version passée (liée à la version technologique ciblée)
Politique de reprise	Délai d'attente de 24 heures avant la 1re reprise (voucher retake à utiliser sous 60 jours)
Modalité	En centre Certiport agréé (CATC) avec Compass — OpenCertif est centre Certiport autorisé
Badge numérique	Badge officiel délivré via Credly après réussite
Position dans le catalogue	Brique fondamentale du parcours IT — souvent passée avant les certifications constructeurs (Microsoft, Cisco, AWS, etc.)

2. Profil du candidat

En tant que candidat à l'examen ITS Python, vous développez et validez des compétences fondamentales en python. Vous êtes capable de :

- Comprendre la syntaxe Python et les conventions PEP 8.
- Déclarer et utiliser les **data types** : int, float, str, bool, list, tuple, dict, set.
- Effectuer des opérations sur les variables avec les **opérateurs** arithmétiques, comparaison, logiques.
- Gérer le **type casting** et la conversion entre types.
- Créer des structures de control flow : **if / elif / else**.
- Créer des **loops** : for, while, break, continue.
- Gérer les **inputs / outputs** avec input() et print(), file I/O (open, read, write).
- Écrire et appeler des **fonctions** avec paramètres, return, arguments par défaut.
- Comprendre les **scopes** : global, local, nonlocal.
- Documenter le code avec **docstrings** et commentaires.
- Gérer les **exceptions** avec try / except / else / finally.
- Utiliser les **modules** standards : math, datetime, random, os, sys.
- Importer des modules : import, from, as.
- Manipuler les **strings** : slicing, méthodes (split, join, replace, format).
- Débugger un script Python (lecture de la stack trace).

L'examen évalue spécifiquement les familles de compétences essentielles à tout débutant en python :

- Opérations sur data types et opérateurs
- Flow Control avec décisions et boucles
- Input / Output operations
- Structure et documentation du code
- Troubleshooting et exception handling
- Modules et tools

3. Prérequis et public cible OpenCertif

Aucun prérequis académique formel n'est exigé. Pearson et Certiport recommandent toutefois :

- **Environ 150 heures de pratique Python** (recommandation officielle).
- Accès à Python 3.x installé (python.org) avec IDLE ou un IDE (VS Code, PyCharm).
- Notions algorithmiques de base (variables, conditions, boucles).
- Familiarité avec un terminal / invite de commande.
- Anglais niveau scolaire suffisant.
- Aucun prérequis académique formel.

Public cible OpenCertif

- Étudiants en informatique, data science, ingénierie.
- Lycéens en NSI (Numérique et Sciences Informatiques).
- Développeurs juniors et candidats junior Python.
- Data analysts et scientists débutants (Python = langage de référence en data).
- Profils en reconversion vers le développement.
- Auto-didactes formalisant leurs compétences Python.
- Profils techniques (DevOps, SysAdmin) souhaitant valider le scripting Python.

4. Domaines de compétences mesurées

L'examen est structuré autour de 6 grands domaines de compétences. Le tableau ci-dessous indique le poids relatif de chaque domaine dans l'évaluation finale (version 2026 — aligné sur la version actuelle ITS Pearson). Les pondérations sont des estimations issues du guide officiel Unity / Certiport.

Domaine	Intitulé	Pondération
1	Opérations sur data types et opérateurs	20 %
2	Flow Control avec décisions et boucles	20 %
3	Input / Output operations	15 %
4	Structure et documentation du code	15 %
5	Troubleshooting et exception handling	15 %
6	Modules et tools	15 %

Remarque : l'examen UCU Programmer dure environ 50 minutes pour 40 questions, soit environ 1 minute 15 par question. La gestion du temps est essentielle. Le score requis pour valider est de **500 sur 700** (sur une échelle officielle Unity de 200 à 700 points).

5. Détail des compétences mesurées

Cette section détaille de manière exhaustive l'ensemble des compétences couvertes par l'examen ITS Python, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur la version actuelle ITS Pearson).

1 Opérations sur data types et opérateurs

20 %

1.1 Variables et data types

- ▶ Déclarer des variables et utiliser les data types : int, float, str, bool.
- ▶ Manipuler les **collection types** : list, tuple, dict, set.
- ▶ Effectuer du **type casting** : int(), float(), str(), bool(), list().
- ▶ Comprendre la mutabilité vs immutabilité.

1.2 Opérateurs et expressions

- ▶ Opérateurs arithmétiques (+, -, *, /, //, %, **).
- ▶ Opérateurs de comparaison (==, !=, <, >, <=, >=).
- ▶ Opérateurs logiques (and, or, not).
- ▶ Opérateurs d'assignation (=, +=, -=, *=, /=).
- ▶ Opérateurs d'identité (is, is not) et d'appartenance (in, not in).

2 Flow Control avec décisions et boucles

20 %

2.1 Décisions conditionnelles

- ▶ Structure **if / elif / else**.
- ▶ Nested if et combined conditions.
- ▶ Opérateur ternaire : ``x if condition else y``.

2.2 Boucles

- ▶ **for** loops avec `range()` et itérables.
- ▶ **while** loops et conditions de sortie.
- ▶ **break, continue, else** sur boucles.
- ▶ Nested loops et complexité.

2.3 List comprehensions

- ▶ Créer des list comprehensions : ``[x*2 for x in nums]``.
- ▶ Filtrer avec condition : ``[x for x in nums if x > 0]``.
- ▶ Dict et set comprehensions.

3 Input / Output operations

15 %

3.1 Console I/O

- ▶ Lire du clavier avec **input()** et convertir le type.
- ▶ Afficher avec **print()** : `sep, end, flush`.
- ▶ Formattage : f-strings, `.format()`, % operator.

3.2 File I/O

- ▶ **open()** avec modes : `r, w, a, r+, b`.
- ▶ Lire : `read(), readline(), readlines(), iteration`.
- ▶ Écrire : `write(), writelines()`.
- ▶ Context manager : **with open(...) as f**.

4 Structure et documentation du code

15 %

4.1 Fonctions

- ▶ Définir avec **def**.
- ▶ Paramètres positionnels, keyword arguments, *args, **kwargs.
- ▶ Valeurs de retour avec **return**.
- ▶ Default parameter values.
- ▶ Lambda functions.

4.2 Scope et visibilité

- ▶ Variables **global**, **local**, **nonlocal**.
- ▶ Closures et nested functions.

4.3 Documentation

- ▶ **Docstrings** et conventions PEP 257.
- ▶ Commentaires avec #.
- ▶ PEP 8 (style guide) : indentation, naming conventions.

5 Troubleshooting et exception handling

15 %

5.1 Types d'erreurs

- ▶ Syntax errors vs runtime errors (exceptions).
- ▶ Exceptions communes : ValueError, TypeError, IndexError, KeyError, ZeroDivisionError, FileNotFoundError.

5.2 Gestion des exceptions

- ▶ **try / except** blocks.
- ▶ Multiple except clauses.
- ▶ **else** et **finally**.
- ▶ Raise une exception : **raise ValueError(...)**.
- ▶ Créer des custom exceptions.

6 Modules et tools

15 %

6.1 Import et modules

- ▶ Importer : **import module**, **from module import name**, **as alias**.
- ▶ Modules standards : **math**, **datetime**, **random**, **os**, **sys**, **json**.
- ▶ Créer ses propres modules.

6.2 Bonnes pratiques

- ▶ Convention de nommage : snake_case pour variables et fonctions.
- ▶ PascalCase pour classes.
- ▶ Bonnes pratiques de structure : `if __name__ == "__main__":`.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au ITS Python à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en Python 3, IDLE, variables, data types, fonctions, modules, file I/O, exceptions et accompagnement tutoré.

Format de la formation

Durée recommandée	150 heures de pratique Python recommandées par Pearson / Certiport (OpenCertif structure ce parcours sur 50 à 70 heures de formation tutorée complétées par 80 à 100 heures de projet et exercices)
Modalité	100 % distanciel asynchrone, ou blended (distanciel + classes virtuelles)
Support pédagogique	Unity Certified User Courseware officiel (GMetrix) + ressources OpenCertif (modules Rise 360, scénarios immersifs)
Plateforme LMS	Imsopecertif.fr (Moodle) — accès 24/7 pendant 12 mois
Encadrement	Tutorat asynchrone par expert Unity certifié + classes virtuelles bimensuelles
Pratique requise	Au moins 150 heures de pratique Unity (recommandation officielle Unity Technologies)
Évaluations	Quiz formatifs par module, 3 projets pratiques Unity, examens blancs CertPREP
Certification finale	Passage de l'examen ITS Python en centre OpenCertif (CATC Certiport)

Parcours d'apprentissage proposé

- **Module 1** : Introduction à Python — IDLE, syntaxe.
- **Module 2** : Variables et data types numériques.
- **Module 3** : Strings et manipulation.
- **Module 4** : Listes, tuples, dictionnaires, sets.
- **Module 5** : Opérateurs et expressions.
- **Module 6** : Structures conditionnelles if / elif / else.
- **Module 7** : Boucles for et while.
- **Module 8** : Console I/O — input() et print().
- **Module 9** : File I/O et context manager.

- **Module 10** : Fonctions et lambda.
- **Module 11** : Scope, closures, *args, **kwargs.
- **Module 12** : Documentation et PEP 8.
- **Module 13** : Exception handling.
- **Module 14** : Modules standards.
- **Module 15** : List comprehensions.
- **Module 16** : Mini-projet — script Python complet.
- **Module 17** : Examen blanc CertPREP.

7. Ressources d'étude officielles

En complément du parcours OpenCertif, les ressources officielles Unity Technologies et Certiport suivantes sont fortement recommandées :

- Documentation officielle Python (sources éditeur).
- Page Certiport officielle : certiport.pearsonvue.com/Certifications/IT-Specialist.
- Page Pearson IT Specialist : [pearson.com / itspecialist](https://pearson.com/itspecialist).
- **CertPREP Practice Tests (GMetrix)** — examens blancs Certiport pour ITS.
- **LearnKey courses** pour IT Specialist — self-paced video learning.
- Tutoriels gratuits et documentation Python.
- Communautés : **Stack Overflow**, **GitHub**, **Reddit** (selon technologie).
- Plateformes d'apprentissage : **Coursera**, **edX**, **Udemy**, **Pluralsight**.
- Badge officiel délivré via **Credly** (credly.com).
- Page Pearson VUE pour la réservation : home.pearsonvue.com.
- Pages OpenCertif dédiées : opencertif.fr / [its](https://its.pearsonvue.com).

8. Modalités de passage de l'examen

Inscription	Via OpenCertif ou directement auprès d'un centre Certiport
Centre d'examen	OpenCertif — Centre Certiport Authorized Testing Center (CATC) / Pearson VUE
Mode de passage	En centre uniquement (Unity n'autorise pas l'examen OnVUE à distance pour les certifications UCU — présence sur site requise)
Pièce d'identité	1 pièce d'identité avec photo obligatoire le jour de l'examen (pour les mineurs : autorisation parentale et CNI / passeport)
Aménagements	Demande possible auprès de Certiport (temps additionnel, assistance technique)
Résultat	Score communiqué immédiatement à la fin de l'examen (échelle 200-700, seuil de réussite 500)
Validité de la certification	3 ans à partir de la date de réussite — attribuée une seule fois (stackable, pas de renouvellement payant requis)
Politique de reprise	Délai d'attente de 24 heures avant la 1re reprise. Voucher retake à utiliser sous 60 jours après l'échec.

Badge numérique

Badge officiel délivré via Credly et intégrable à LinkedIn, CV, portfolio, sites de recrutement

9. Contact et inscription

Pour toute information complémentaire, demande de devis ou inscription à la formation préparatoire au ITS Python, l'équipe OpenCertif reste à votre disposition. OpenCertif est un Centre Certiport Authorized Testing Center (CATC) habilité à délivrer les certifications Unity Certified User.



10. Mentions légales et version

Ce syllabus est établi par OpenCertif sur la base des Objective Domains officiels publiés par Certiport pour la certification ITS Python, dans sa version applicable (version 2026 — aligné sur la version actuelle ITS Pearson). Les compétences mesurées, les pondérations et les objectifs présentés reflètent fidèlement la structure de l'examen telle que publiée par Unity Technologies via Certiport.

Pearson, le logo Pearson, Pearson VUE, Certiport, CertPREP, GMetrix, Compass et IT Specialist (ITS) sont des marques déposées de Pearson Education Inc. Python est une marque déposée de la Python Software Foundation. LearnKey est une marque déposée de LearnKey Inc. Credly est une marque déposée de Pearson Education Inc.

OpenCertif n'est pas affilié à Unity Technologies. Ce document est fourni à titre informatif. Pour la version officielle et à jour des Objective Domains, consulter certiport.pearsonvue.com/Certifications/Unity et unity.com/products/unity-certifications.

Version du syllabus : 2026.05 — Édition mai 2026

Source officielle Certiport : certiport.pearsonvue.com/Certifications/Unity/Certified-User/Certify

Source officielle Unity : unity.com/products/unity-certifications/user-programmer

Page OpenCertif : opencertif.fr/unity-user-programmer