

La fiche Python de seconde aide à réviser et à s'entraîner

Leçon de Python en seconde, exercices progressifs, correction détaillée et PDF à imprimer pour comprendre vite et s'entraîner en autonomie.

Éducation lycée — méthodes, fi

Seconde

Prénom : _____

Date : ___ / ___ / ___

Version imprimable

Une fiche Python pour la seconde résume les bases du langage : variables, types, conditions, boucles et fonctions simples. Elle permet de comprendre rapidement une leçon, puis de s'exercer avec des exemples corrigés, des exercices progressifs et un PDF à imprimer chez soi.

Devant un écran, beaucoup d'élèves lisent `print("Bonjour")` sans difficulté, puis hésitent dès qu'il faut créer une variable ou corriger une erreur de syntaxe. En seconde, Python s'apprend mieux avec une progression très simple : un rappel court, une méthode pas à pas, deux exemples résolus, puis des exercices de difficulté croissante. Cette fiche aide l'élève à travailler seul, le parent à suivre les notions essentielles et l'enseignant à disposer d'un support propre à imprimer. On y retrouve le vocabulaire utile, les commandes de base, une version élève, une correction séparée et un format clair pour réviser avant une activité ou une évaluation.

Le mémento Python de Seconde ; : objectif, prérequis et vocabulaire essentiel

Niveau ; : Seconde · **Cycle** ; : à vérifier avant publication · **Matière** ; : NSI · **Domaine** ; : Python. Cette **fiche python** aide à reconnaître les **bases python**, lire un court script et écrire des instructions simples. Le PDF A4 à imprimer garde un cadre net ; : « ; Prénom ; : _____ ;», « ; Date ; : _____ ;», page élève puis correction séparée.

Télécharger le PDF

Voir la correction

Repère rapide ; : ce **mémento python** sert à comprendre un petit programme avant de s'entraîner, en initiation de **programmation** liée à **Numérique et sciences informatiques**.

Je sais lire quelques lignes en **langage Python**, dire ce qu'elles font et écrire une instruction correcte avec variable, fonction et indentation.

Prérequis ; : distinguer texte et nombre ; ; suivre une consigne ligne par ligne ; ; accepter l'essai-erreur.

Mot	Repère
variable	nom qui stocke une valeur
type	nature d'une donnée

À retenir ; : comprendre un script court d'abord.

`print("Bonjour")` affiche un texte.

△ Ne pas confondre affectation et test d'égalité ; ; l'indentation compte.

Python, ici *Python (langage)*, n'est ni le serpent ni une référence culturelle ; : c'est un langage **interprété** et **multiplateforme**. Vocabulaire utile ; : **chaîne de caractères**, **booléen**, **fonction**, **argument**, **indentation**. Les repères d'**Éduscol** invitent à rester simple en Seconde ; : utile pour débiter, moins pour aller déjà vers une NSI approfondie.

Bases de Python : méthode pas à pas pour écrire et tester un petit programme

Pour écrire un petit programme en **Python**, gardez une routine fixe ; : lire, nommer, écrire, tester. Cette **méthode python** évite les erreurs de débutant. Elle aide à choisir des **variables python** claires, à repérer les bons **types python** et à avancer sans se perdre dans une fiche plus dense, du type *Banque PT*.

1. Lisez la consigne et relevez les données utiles ; : ce que l'on connaît, ce qu'il faut demander et ce que le programme doit afficher.
2. Nommez les variables avec des mots simples, comme *age*, *note* ou *prenom*, car une variable stocke une valeur qu'on doit reconnaître d'un coup d'œil.
3. Écrivez quelques lignes avec **print()** pour afficher, **input()** pour saisir, puis convertissez si besoin avec *int* ou *float* ; ; dans une fonction, un **argument python** est l'information placée entre parenthèses.

4. Lancez, relisez la sortie, puis corrigez calmement les parenthèses, l'**indentation python** et la différence entre =, qui affecte une valeur, et ==, qui teste une égalité.

Un réflexe simple aide beaucoup ; : tester après chaque ligne ajoutée. En Seconde, c'est plus utile que d'aller vite. Un support vidéo peut être placé juste après cette section, si une ressource fiable a été validée.

I

python • Les variables • Ce qu'il faut savoir & erreurs à ne pas faire • int float et string - lycée — jaicompris Maths

Exemples résolus : afficher, calculer, tester une condition

En seconde, le premier déclic arrive souvent avec trois lignes seulement. C'est assez. Un bon **exemple python** montre le code, la sortie attendue, puis une **correction expliquée** qui nomme simplement chaque action. Ici, on travaille les **bases du langage** de *Python (langage)* : créer une **variable**, stocker une **chaîne de caractères**, afficher un résultat avec **print python**, puis utiliser une **condition** pour décider. Rien de spectaculaire. Mais c'est précisément ce qui rassure, car l'élève voit que le programme lit les lignes dans l'ordre et exécute une consigne à la fois, à condition que l'écriture soit exacte.

Exemple 1

```
prenom = "Lina"
print(prenom)
```

Sortie attendue : Lina

Explication : la première ligne crée la variable **prenom** et lui donne un texte entre guillemets. La seconde ligne affiche ce contenu. Sans guillemets, Python cherche un nom de variable et renvoie une erreur.

Exemple 2

```
nombre = 5
if nombre > 0 : print("positif")
```

Sortie attendue : positif

Explication : nombre vaut 5. La ligne **condition if python** teste si ce nombre est supérieur à 0 ; comme c'est vrai, la ligne indentée s'exécute. Si le nombre valait , rien ne s'afficherait ici.

Pièges à éviter : oublier les guillemets autour d'un texte, mélanger texte et nombre sans conversion, écrire = au lieu de == quand on compare, ou décaler l'indentation après **if**. Ces erreurs sont courantes ; elles servent justement de repères avant les exercices.



Exercices progressifs à imprimer sur la fiche python

Oui ; : une bonne **fiche python pdf** de Seconde doit proposer des **exercices python** brefs, gradués et pensés pour le **pdf à imprimer** ; ; l'élève lit du code, complète une ligne, prévoit une sortie, puis termine par un défi bonus bien séparé. Au **Lycée Condorcet**, ce format fonctionne bien en initiation, car il rassure le débutant tout en gardant un vocabulaire scolaire stable, y compris dans l'esprit de *Pixees*. Ce support aide surtout à automatiser les bases ; ; un élève déjà très autonome préférera parfois un mémento plus serré.

Action	Écriture utile
Afficher	<code>print(x)</code>
Saisir	<code>nom = input("...")</code>
Convertir	<code>n = int(input("..."))</code>

À retenir ; : cette *fiche révision python* va du repérage à l'écriture, avec renvoi final vers la leçon liée, les exercices liés et l'évaluation si ces pages existent.

Exemple minute ; : `x = 4`, puis `print(x)` affiche 4.

⚠ Confondre texte et nombre, ou oublier l'indentation après `if`.

Correction détaillée, erreurs fréquentes et à retenir

La **correction python** n'est utile que si elle suit exactement la fiche élève ; : mêmes numéros, même ordre, réponse brève, puis explication claire. Dans le **pdf correction**, la solution gagne à apparaître en **gras coloré**, suivie d'une phrase simple qui justifie le choix ; : variable bien nommée, type reconnu, résultat produit par `print()`, condition indentée sans erreur. Rien de décoratif. Ce format rend la *fiche python corrigée* vraiment exploitable par un élève de Seconde qui révise seul, mais aussi par un parent ou un enseignant qui veut vérifier vite. Le balisage final pourra intégrer une FAQPage, sans alourdir la lecture.

À retenir ; : pour l'*à retenir python*, gardez cinq repères ; : une variable stocke une valeur ; ; les types de base sont `int`, `float`, `str` et `bool` ; ; `print()` affiche ; ; une condition teste une situation ; ; l'indentation organise le programme. Le PDF premium gardera un footer avec URL canonique une fois validée, ressources liées réellement disponibles, *branding* discret, et un fichier texte-sélectionnable, autonome, imprimable, utile même sans la page web.

Les interrogations courantes

Quel type de langage est Python ?

Python est un langage de programmation de haut niveau, interprété et généraliste. Cela signifie qu'il est proche du langage humain, qu'il s'exécute sans compilation complexe pour débiter, et qu'on peut l'utiliser dans des domaines variés : calcul, web, automatisation, intelligence artificielle ou encore traitement de données. En lycée, on l'emploie surtout pour apprendre les bases de l'algorithmique.

Comment créer des variables sur Python ?

Pour créer une variable en Python, j'écris un nom, puis le signe `=`, puis une valeur. Par exemple : `age = 15` ou `nom = "Lina"`. Python crée la variable au moment de l'affectation. Il ne faut pas mettre d'espace dans le nom, éviter les accents, et choisir un nom clair pour comprendre le programme.

C'est quoi un argument en Python ?

Un argument est une valeur donnée à une fonction au moment où on l'utilise. Par exemple, dans `print("Bonjour")`, la chaîne "Bonjour" est l'argument de la fonction `print`. Les arguments servent à préciser ce que la fonction doit faire ou sur quelle donnée elle doit agir. Une même fonction peut recevoir un ou plusieurs arguments.

Comment définir un mot sur Python ?

En Python, un "mot" s'écrit comme une chaîne de caractères. Pour le définir, je le place entre guillemets simples ou doubles : `mot = "bonjour"` ou `mot = 'bonjour'`. Python comprend alors qu'il s'agit de texte et non d'un nom de variable. C'est la bonne manière de stocker un prénom, une réponse ou un message.

Comment créer une variable en Python ?

Créer une variable en Python consiste à faire une affectation. J'écris par exemple `x = 4`, `prenom = "Nora"` ou `note = 17.5`. Le signe `=` signifie ici "reçoit la valeur". On peut ensuite réutiliser la variable dans des calculs, des tests ou des affichages. Une variable peut aussi changer de valeur au cours du programme.

Qu'est-ce qu'un argument en Python ?

Un argument est l'information transmise à une fonction quand on l'appelle. Dans `len("chat")`, "chat" est l'argument donné à la fonction `len`. Grâce à lui, la fonction sait sur quelle donnée travailler. Je conseille de repérer les parenthèses : ce qui est dedans correspond souvent aux arguments utilisés par la fonction.

Comment définir un « mot », donc une chaîne de caractères, en Python ?

Pour définir une chaîne de caractères en Python, il faut entourer le texte avec des guillemets. Par exemple : `ville = "Paris"`. Même si le texte contient un seul mot, Python le traite comme une chaîne. On peut ensuite l'afficher, la comparer ou la concaténer avec une autre chaîne. Sans guillemets, Python chercherait une variable de ce nom.

Quelle différence entre `=` et `==` en Python ?

En Python, `=` sert à affecter une valeur à une variable, alors que `==` sert à comparer deux valeurs. Par exemple, `x = 5` donne 5 à x, tandis que `x == 5` vérifie si x vaut 5. La comparaison renvoie `True` ou `False`. C'est une distinction essentielle pour éviter les erreurs.

Dernière mise à jour : 10/06/2026

[Continue sur lycee-condorcet.fr](https://lycee-condorcet.fr)

Lycée Condorcet - Document pédagogique