Créer son outil de contact avec l’IA

Faire créer un programme qui prépare vos prises de contact et tester son fonctionnement.

Votre activité

Demandez à l’IA de créer votre outil en Python, lancez une simulation et vérifiez les messages obtenus.

Créez un dossier d’essai dans l’agent utilisé au jour 2. Téléchargez contacts-exemple.csv et placez-le dans ce dossier. Reprenez votre message validé dans la leçon précédente, ou utilisez l’exemple fourni. Vous allez faire créer votre propre script. Dans un chat simple, enregistrez le code fourni dans contact.py. Nabil peut accompagner les premières manipulations Python à l’oral.

Comprendre avant de pratiquer

Pendant sa recherche d’emploi, Nabil a utilisé un script pour contacter automatiquement des recruteurs. Son ancien outil peut ne plus fonctionner aujourd’hui. L’idée à retenir : vous pouvez demander à l’IA de créer votre propre outil, puis le faire évoluer avec elle.

Un script Python est un programme qui exécute des consignes. L’IA peut l’écrire et expliquer comment le lancer. Un agent navigateur agit sur les pages ; un script peut répéter une opération définie sur une liste. Sans appel à un modèle, son exécution ne consomme pas de tokens ; une personnalisation par IA à chaque ligne, elle, peut en consommer.

Préparer un brouillon et envoyer un message sont deux actions distinctes. On teste d’abord le programme sur des données fictives, puis un envoi témoin vers sa propre boîte si le compte le permet. LinkedIn interdit les outils tiers automatisant l’activité sur son site : l’exercice utilise donc des messages email.

Exemple fictif si besoin

Le CSV contient des contacts et entreprises entièrement fictifs. Lea et Sami ont chacun une adresse d’exemple ; Lea apparaît deux fois, et Noa n’a pas d’adresse. Les noms en .example servent uniquement à la simulation. Le message fourni illustre le fonctionnement du programme, pas une candidature réelle.

Guidé : les étapes et les prompts

1. Préparer les données et Python

Ouvrez le CSV : quatre lignes fictives, dont un doublon et une adresse manquante. Préparez votre message de base ou utilisez l'exemple. Demandez à l'IA de vérifier votre environnement avant de proposer une installation.

Prompt à adapter

Je débute. Je veux créer un petit outil Python de prise de contact dans ce dossier. Mon système est [Windows, macOS ou Linux]. Aide-moi à vérifier si Python est déjà disponible. Explique une action à la fois, son résultat attendu et comment retrouver les fichiers. Si une installation est nécessaire, utilise une source officielle et attends mon accord avant de la lancer.

2. Faire créer une première version

Commencez avec les contacts fictifs. Les champs du message seront remplis à partir du CSV, sans nouvel appel IA à chaque contact.

Prompt à adapter

Crée contact.py pour lire contacts-exemple.csv (UTF-8, séparateur point-virgule). Mon message validé : [message avec {prenom}, {entreprise} et {motif}]. Crée un brouillon TXT lisible par contact complet dans brouillons/ et un suivi CSV. Détecte les doublons par id ou email et signale les adresses absentes sans les inventer. La simulation est le mode par défaut : aucun envoi ni accès réseau. Une deuxième exécution ne doit pas créer de doublons ni écraser mes brouillons. Utilise seulement Python standard. Explique comment lancer et vérifier le résultat.

Pas d'idée ? Exemple prêt à essayer

Crée contact.py pour lire contacts-exemple.csv (UTF-8, point-virgule). Message de démonstration : « Bonjour {prenom}, votre activité chez {entreprise} m'intéresse, notamment {motif}. Je souhaite découvrir les compétences utiles dans votre équipe. Seriez-vous disponible pour un bref échange ? » Crée un brouillon TXT par contact complet dans brouillons/ et un suivi CSV. Détecte les doublons par id ou email. Signale les adresses absentes sans les inventer. Simulation par défaut, aucun réseau ni envoi. Une deuxième exécution ne doit ni dupliquer ni écraser les brouillons. Python standard uniquement ; explique comment lancer et vérifier.

3. Lancer et contrôler

Suivez la commande adaptée à votre ordinateur. Avec le fichier fourni, attendez deux brouillons, un doublon écarté et une adresse manquante signalée. Ouvrez les deux fichiers et vérifiez prénom, entreprise et motif. Relancez : les deux brouillons doivent rester uniques.

4. Faire corriger un problème

Donnez à l'IA le résultat observé, le résultat attendu et le message d'erreur exact si nécessaire. Retestez après correction.

Prompt à adapter

> Résultat observé : [ce qui se passe ou erreur, sans secret]. Résultat attendu : deux brouillons uniques avec le fichier d'essai, un doublon écarté, une adresse manquante signalée, aucun envoi. Explique la cause et corrige le programme sans effacer mes fichiers. Fais-moi refaire le test, y compris une deuxième exécution.

5. Préparer l'envoi de test

Une fois la simulation vérifiée, vous pouvez ajouter la fonction d'envoi avec le prompt dans Aller plus loin. Commencez par votre propre adresse. La configuration dépend de la messagerie ; aucun envoi réel n'est nécessaire pour valider ce premier atelier. Si vous l'essayez, vérifiez la réception dans la boîte témoin.

Vérifiez votre résultat

- Vous savez retrouver le programme, les données, les brouillons et le suivi.
- Les deux messages reprennent les bons champs, sans contact ni fait inventé.
- Le doublon et l'adresse absente sont traités ; relancer ne duplique pas les brouillons.
- Vous distinguez simulation, brouillon préparé, envoi accepté par le service et réception constatée.

Accompagné : construisez votre demande

- Demandez une première version simple, puis ajoutez une fonction à la fois.
- Copiez les erreurs en retirant mots de passe et clés. L'IA peut vous aider à les comprendre.

Autonome : votre défi

Définissez votre outil de contact, faites-le créer par l'IA et vérifiez-le sur le CSV fictif. Ajoutez une amélioration utile et prouvez qu'elle fonctionne. Si vous ajoutez l'envoi, testez seulement votre propre adresse avant toute utilisation réelle.

Pour poursuivre

Ajoutez un envoi témoin vers votre propre boîte, avec une méthode permise par votre fournisseur. Si cette configuration est inaccessible, améliorez la prévisualisation ou les filtres. Pour un usage réel ultérieur, choisissez et relisez les destinataires et messages, respectez les règles du service et les demandes de ne plus être contacté.

Prompt du prolongement

La simulation de mon outil fonctionne. Aide-moi à ajouter un envoi email de test. Demande d'abord quelle messagerie j'utilise et vérifie sa documentation officielle ; propose une méthode autorisée par ce service. Ne demande jamais mon mot de passe ou ma clé dans le chat ni dans le code : explique la configuration locale adaptée. Garde la simulation par défaut. Le mode envoi doit demander une action explicite, montrer le destinataire et le message, autoriser uniquement ma propre adresse de test et refuser les adresses fictives. Envoie au maximum un message témoin après ma validation, sans relance automatique. Distingue dans le journal les brouillons, les envois acceptés par le service et les erreurs ; une réponse ambiguë impose une vérification avant de retenter. Empêche un deuxième envoi du même message. Explique comment vérifier moi-même sa réception et désactiver le mode envoi.

Fichiers utiles

Contacts fictifs pour créer et tester votre outil: https://formation.nabdata.fr/ressources/contacts-exemple.csv

Outils et sources

Installer Python depuis le site officiel : https://www.python.org/downloads/

Python : écrire et envoyer des emails : https://docs.python.org/3/library/email.examples.html

LinkedIn : activité automatisée : https://www.linkedin.com/help/linkedin/answer/a1340567/automated-activity-on-linkedin?lang=en