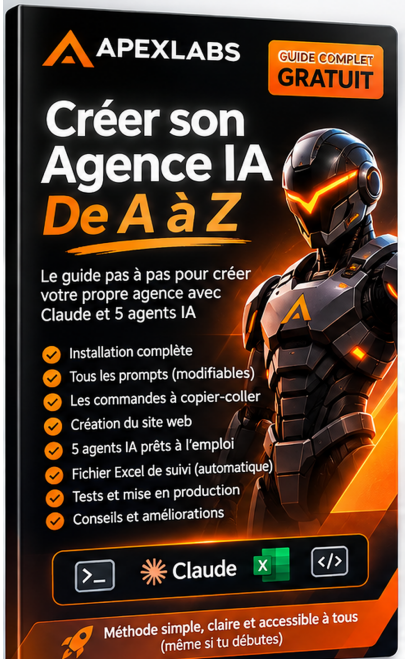


APEXLABS / ÉDITION DÉBUTANT

CRÉER SON AGENCE IA DE A À Z

Le guide complet pour créer un site, cinq agents IA et un tableau Excel de suivi, depuis un terminal Windows.

Avec des commandes à copier-coller, des prompts personnalisables, des exemples fictifs et des tests simples.

> Important : ce guide crée un prototype d'agence. Un site et des agents ne prouvent pas qu'une entreprise est légalement créée, qu'elle possède des clients ou qu'elle génère des revenus. Les tests utilisent des données fictives et demandent une vérification humaine.

Édition ApexLABS / ARMORRED • Novembre 2026 • Guide pédagogique gratuit • Aucun revenu promis

Bienvenue : comment utiliser ce guide

Imagine une petite équipe de cinq assistants numériques. Tu leur donnes une mission; chacun exécute un morceau du travail. Ils préparent des idées et des documents, mais tu restes responsable des décisions. Voilà ApexLABS.

Tu peux suivre les étapes sans savoir programmer. Il suffit de lire les encadrés, de copier les commandes et de vérifier le résultat avant de continuer. Certaines installations nécessitent l'aide d'un adulte responsable, notamment la gestion des comptes, des achats et des données de clients.

Les trois couleurs du guide

- Orange : une action à faire ou une étape essentielle.
- Gris : une commande ou du texte à copier dans le terminal.
- Encadré clair : une précaution ou un point de contrôle.

Ton résultat final

- Un site vitrine ApexLABS consultable en local.
- Un agent Marché qui prépare une analyse en distinguant faits et hypothèses.
- Un agent Marketing qui prépare des idées de communication.
- Un agent Support qui rédige des réponses à vérifier.
- Un agent Reporting qui crée un classeur Excel local.
- Un agent Superviseur qui coordonne et vérifie le travail.
- Un dossier de démonstrations et un guide réutilisable.

> Aucune étape ne doit envoyer des emails à de vraies personnes, acheter un service, publier un site ou collecter des données privées sans autorisation explicite.

Sommaire

01 Comprendre le projet
02 Préparer Windows et les outils
03 Créer les dossiers et les sauvegardes
04 Écrire un bon prompt
05 Construire le site vitrine
06 Agent 1 : analyse de marché
07 Agent 2 : marketing
08 Agent 3 : support
09 Agent 4 : Excel local
10 Agent 5 : superviseur

11 Démo complète et contrôle qualité
12 Les commandes essentielles
13 Personnaliser son agence
14 Résoudre les problèmes courants
15 Pack vidéo : que filmer et où ranger
16 Publier le guide et protéger les données

01 / Comprendre le projet

Une agence numérique aide d'autres entreprises à améliorer leur présence sur Internet. Ici, le prototype ApexLABS propose trois services : un site de présentation, des automatisations simples et un assistant de réponses fréquentes.

Que veut dire "agent IA" ?

Un agent est un petit programme qui reçoit une mission et produit un résultat. Par exemple : « Prépare une idée de publication pour un salon de coiffure ». Son résultat est un brouillon, pas une vérité automatique.

Élément	Explication très simple
Site web	La vitrine de l'agence : les visiteurs découvrent les services.
Terminal	Une fenêtre où on écrit des commandes au lieu de cliquer.
Python	Un langage qui permet de créer les agents.
Claude Code	Un outil qui peut aider à écrire et tester du code.
Excel	Le carnet de bord : il note ce qui s'est passé.
Superviseur	Le chef d'orchestre : il coordonne les tâches et signale les erreurs.

La règle d'or

Toujours distinguer trois types d'informations : fait vérifié (avec sa source), hypothèse (une idée plausible à vérifier) et démonstration fictive (un scénario inventé pour apprendre).

02 / Installer les outils sur Windows

Les exemples ci-dessous utilisent PowerShell. Pour l'ouvrir : bouton Démarrer → rechercher « PowerShell » → ouvrir. Une fenêtre avec « PS C:\...> » apparaît.

Étape 1 — Vérifier Python

Python fait tourner les agents. Tape la commande suivante :

```
python --version
```

Si tu vois « Python 3.x », c'est bon. Si Windows ne trouve pas Python, consulte https://www.python.org/downloads/ et l'installation officielle. Pendant l'installation, vérifie l'ajout de Python au PATH.

Étape 2 — Vérifier Node.js

Node sert à certaines animations, notamment Remotion. Il n’est pas nécessaire pour les agents Python eux-mêmes.

```
node --version
npm --version
```

Si ces commandes échouent, consulte le site officiel : https://nodejs.org/.

Étape 3 — Vérifier Claude Code

Claude Code est l’assistant qui aide à programmer. Sa disponibilité, ses modalités d’accès et ses limites dépendent de ton compte et de sa configuration actuelle.

```
claude --version
claude --help
```

Ne colle jamais une clé privée dans un prompt ou une capture d’écran. Pour installer ou configurer Claude Code, utilise la documentation officielle de Claude Code et vérifie ses instructions avant toute commande d’installation.

Étape 4 — Vérifier l’éditeur de texte

VS Code est utile pour visualiser le code et les rapports, mais un simple éditeur suffit pour lire des fichiers.

```
code --version
```

Si « code » n’est pas reconnu, ouvre VS Code depuis le menu Démarrer puis choisis Fichier → Ouvrir un dossier.

03 / Créer le dossier de l’agence

Pour suivre cet exemple tel quel sur le PC du créateur de la vidéo, voici le chemin du projet. Sur ton ordinateur, remplace le nom du compte et le chemin par ton propre dossier.

```
$project = "$HOME\Desktop\Mon_Agence_IA"
New-Item -ItemType Directory -Force -Path $project | Out-Null
Set-Location $project
```

Crée maintenant les sous-dossiers essentiels :

```
$folders = @("BUSINESS\site_web", "BUSINESS\agents", "BUSINESS\tests",
  "BUSINESS\reports", "BUSINESS\data", "BUSINESS\config",
  "01_RUSHS", "02_SCREEN_RECORDINGS", "03_SCREENSHOTS",
  "04_AUDIO", "05_IMAGES", "06_SCRIPT", "08_EXPORTS")
foreach ($f in $folders) {
  New-Item -ItemType Directory -Force -Path (Join-Path $project $f) | Out-Null
}
Get-ChildItem $project
```

Le résultat attendu : une liste de dossiers, sans message d’erreur. L’important est de travailler dans le dossier du projet, jamais dans C:\WINDOWS\system32.

Schéma de l’architecture

Dossier	Rôle
BUSINESS/site_web	Les fichiers du site HTML, CSS et JavaScript
BUSINESS/agents	Les scripts Python des cinq agents
BUSINESS/reports	Les rapports et le classeur APEXLABS_SUIVI.xlsx
BUSINESS/tests	Les tests automatisés
03_SCREENSHOTS	Les images à utiliser dans la vidéo
01_RUSHS	Les rushs face caméra si tu réalises une vidéo

Sauvegarder son travail

Un ordinateur peut planter. Copie régulièrement le projet sur un autre support et utilise Git si tu le connais. Un dépôt Git en local ne remplace pas une sauvegarde des vidéos et documents.

```
git init
git status
```

> Avant Git, crée un fichier .gitignore et exclus les fichiers secrets, les vidéos lourdes, node_modules et les fichiers temporaires. Ne publie pas de clés d’accès ou de documents privés.

04 / Apprendre à écrire un prompt

Un prompt est une consigne. Pour que Claude travaille correctement, donne-lui cinq informations : le rôle, le but, les fichiers à produire, les limites et la vérification attendue.

```
Tu es [ROLE].
Objectif : [CE QUE JE VEUX].
Contexte : [MON ACTIVITÉ ET MON PUBLIC].
Livrables : [FICHIERS A CRÉER].
Contraintes : [CE QUI EST INTERDIT OU OBLIGATOIRE].
Tests : [COMMENT VÉRIFIER QUE CA FONCTIONNE].
Présente ton plan avant de modifier des fichiers.
```

Exemple facile

Mauvais prompt : « Fais-moi une agence ». Bon prompt : « Crée le prototype local d’une agence de sites vitrines pour petites entreprises, en français, avec un formulaire de démonstration qui n’envoie rien, une page mobile et des tests ».

> Les passages entre [CROCHETS] sont des zones à personnaliser. Remplace-les avant l’envoi.

05 / Créer le site vitrine ApexLABS

Le site présente les services. Il doit être accessible localement, facile à lire et adapté aux téléphones. Pour commencer, pas besoin de serveur complexe ni de paiement.

Prompt prêt à personnaliser : site web

```
Tu es développeur web. Dans BUSINESS/site_web, crée un site vitrine
local pour [NOM_AGENCE]. Public : [CLIENTS_CIBLES].
Services : [SERVICE_1], [SERVICE_2], [SERVICE_3].
Crée index.html, assets/css/style.css et assets/js/main.js.
Design : noir, anthracite, blanc et orange, mobile-first.
Sections : accueil, services, méthode, démonstrations, FAQ, contact.
La FAQ et le formulaire sont des DEMOS : aucun envoi réel.
Aucun prix, client, témoignage ou revenu inventé.
Teste la navigation, le mobile et l accessibilité au clavier.
Ne publie pas le site. Donne les commandes pour le lancer.
```

Lancer le site en local

```
python -m http.server 8080 --bind 127.0.0.1 --directory BUSINESS\site_web
```

Ouvre http://127.0.0.1:8080 dans ton navigateur. Si la page s’affiche, tu as réussi. Tape Ctrl+C dans PowerShell pour arrêter le serveur.

Contrôle avec les yeux

- Le texte est lisible et il n’y a pas de débordement.
- Les liens du menu mènent aux bonnes sections.
- La version téléphone ne coupe pas les boutons.
- Les exemples fictifs sont indiqués comme fictifs.
- Le formulaire de test n’envoie pas de vraies données.

06 / Agent 1 : analyser un marché

L’agent Marché sert à poser des questions utiles : quel problème existe ? Qui pourrait avoir ce problème ? Quelles alternatives sont déjà proposées ? Quelles informations faut-il vérifier ?

Exemple fictif

Un salon de coiffure fictif souhaite présenter ses horaires et ses prestations sur un site web. L’agent doit proposer une première analyse, pas affirmer que ce salon existe réellement.

Prompt : création de l’agent Marché

```
Crée BUSINESS/agents/market.py, un agent terminal Python.
Entrées : activite, zone générale, public, besoin.
Sorties : BUSINESS/reports/market_report.md + JSON.
Ajoute un mode --dry-run sans réseau, clairement marqué SIMULATION.
Mode réel facultatif : cherche des sources publiques uniquement si autorisé.
Sépare les faits vérifiés, hypothèses et recommandations.
Ne fabrique jamais de source ou de concurrent.
Ajoute tests, gestion d erreurs, limite de temps et budget explicite.
Ne contacte aucune entreprise. Fournis les commandes PowerShell.
```

Tester sans appel externe

```
python BUSINESS\agents\main.py market --dry-run
```

Dans le projet ApexLABS de la vidéo, cette commande existe. Si tu pars d’un projet neuf, elle ne fonctionnera qu’après que Claude aura réellement créé main.py et l’agent Marché.

Préparer un éventuel test réel

```
python BUSINESS\agents\main.py market --reel --plan --modele sonnet --budget 0.50
```

Cette commande affiche le plan prévu dans le projet ApexLABS. Vérifie les permissions et les éventuelles consommations de quota. Le plafond n’équivaut pas nécessairement à une garantie de facturation nulle. Ne lance pas le mode réel sans comprendre ce qu’il va faire.

07 / Agent 2 : préparer le marketing

Cet agent rédige des idées de communication : une phrase de présentation, un post pour les réseaux et quelques sujets à publier. Ce n’est pas un système de spam ou d’envoi automatique.

Prompt personnalisable

```
Crée BUSINESS/agents/marketing.py, un agent terminal Python.
Il reçoit un rapport Marché au format JSON et la liste des services.
Il propose : 3 slogans, 2 posts, 1 présentation courte,
1 mini-calendrier de contenu sur 7 jours.
Chaque texte doit être marqué BROUILLON A VALIDER.
Ne publie rien, n envoie aucun message, ne prétends pas avoir
des clients ni des résultats non démontrés.
Enregistre marketing_report.md et marketing_report.json.
Ajoute un mode simulation et des tests locaux.
```

Exercice

Personnalise la consigne : remplace « salon de coiffure » par « fleuriste », « boulangerie » ou « atelier de réparation » et demande à l’agent deux angles de communication différents.

08 / Agent 3 : préparer le support

Le support répond à une question simple, mais il ne doit pas inventer un tarif, une promesse de livraison ou des règles contractuelles. Les réponses sont des brouillons examinés par un humain.

Prompt personnalisable

```
Crée BUSINESS/agents/support.py, un agent terminal Python.
Entrées : question d un client fictif, FAQ, services connus.
Sorties : brouillon de réponse, catégorie, niveau de confiance,
besoin de vérification humaine, statut, rapport JSON et Markdown.
Si la FAQ ne contient pas la réponse : indique INCERTAIN.
Aucun envoi, aucune promesse de prix, aucune donnée sensible.
Ajoute 5 exemples fictifs et des tests reproductibles.
```

Exemple de question fictive

« Bonjour, j’ai un salon de coiffure et je voudrais une page avec mes horaires. Comment fonctionne votre accompagnement ? » L’agent peut expliquer la méthode, mais pas donner un délai définitif non validé.

09 / Agent 4 : créer le fichier Excel

Cet agent consigne les résultats de chaque tâche dans un fichier .xlsx sur l’ordinateur. Il n’a pas besoin de Google Sheets ni d’un compte Google.

Les trois feuilles

Feuille	Ce qu'on y trouve
EXECUTIONS	Historique daté des tâches (agent, mission, statut, durée, sorties).
DEMANDES_DEMO	Demandes fictives avec leur besoin et leur état.
TABLEAU_DE_BORD	Compteurs et graphiques calculés à partir des lignes réellement enregistrées.

Prompt personnalisable

```
Crée BUSINESS/agents/reporting.py. Génère un classeur Excel local :
BUSINESS/reports/APEXLABS_SUIVI.xlsx.
Feuilles : EXECUTIONS, DEMANDES_DEMO, TABLEAU_DE_BORD.
Thème : noir, anthracite, orange, blanc. Ajoute filtres,
ligne de titres figée, couleurs de statut et graphiques simples.
Écris les journaux en conservant les run_id et sans doublons.
N invente aucune donnée ni aucun chiffre de réussite.
Si les lignes sont fictives, indique SIMULATION en évidence.
Prévois le cas où Excel est ouvert et verrouille le fichier.
Garde aussi les JSON/CSV de traçabilité, et crée des tests.
```

Excel : comment le vérifier

Le fichier final à filmer doit avoir été créé par le programme Reporting, pas simplement copié depuis un exemple. Dans Excel : ouvre les trois feuilles, vérifie les lignes et essaie les filtres.

```
Test-Path "BUSINESS\reports\APEXLABS_SUIVI.xlsx"
Invoke-Item "BUSINESS\reports\APEXLABS_SUIVI.xlsx"
```

Si PowerShell affiche True et si Excel ouvre bien les trois feuilles, le fichier existe. Vérifie ensuite que les chiffres du tableau de bord correspondent aux données de la feuille EXECUTIONS.

> Pour écrire de nouvelles lignes, ferme le fichier dans Excel avant de relancer l'agent : sinon Windows peut refuser la modification. Conserve une sauvegarde avant les gros tests.

Quel thème graphique choisir ?

- Bandeau anthracite avec le nom APEXLABS.
- Orange pour les titres, les indicateurs et les liens visuels.
- Fond clair pour les lignes de données : meilleur confort de lecture.
- Vert pour « Terminé », orange pour « À vérifier », rouge pour « Échec ».
- Un grand avertissement « DÉMONSTRATION FICTIVE » si les chiffres sont simulés.

10 / Agent 5 : le superviseur

Le superviseur est le chef d'orchestre. Il reçoit un objectif, vérifie les résultats de chaque agent et produit un rapport. Il ne doit pas décider tout seul d'envoyer un email ou de payer quelque chose.

Prompt personnalisable

```
Crée BUSINESS/agents/supervisor.py, superviseur Python.
Mission : orchestrer market -> marketing -> support -> reporting.
Attribue un run_id, valide les fichiers JSON produits,
distingue FAITS / HYPOTHESES / DEMOS et conserve les journaux.
Au maximum 1 nouvelle tentative par erreur technique récupérable.
Si un agent échoue, arrête proprement les tâches dépendantes.
Aucune publication, aucun paiement, aucun email automatique.
Produis supervisor_report.md + JSON avec le verdict
VALIDE / A REVOIR / ECHEC et les motifs du verdict.
Crée tests et commande : python BUSINESS/agents/main.py demo --dry-run.
```

Pourquoi contrôler les réponses ?

Une IA peut être persuasive même quand elle se trompe. Le superviseur doit vérifier que les sorties existent, qu'elles respectent un format et qu'elles n'affirment pas un fait sans source. Pour les décisions importantes, l'humain reste responsable.

11 / Démonstration complète : un salon fictif

Imagine une entreprise de coiffure fictive nommée « Salon Démo ». Elle souhaite présenter ses prestations sur Internet. Cette demande n’est pas une vraie commande cliente.

Ordre	Action
1 / Superviseur	Enregistre la mission et distribue les tâches.
2 / Marché	Prépare les besoins et les hypothèses du secteur.
3 / Marketing	Crée des idées de présentation et de publication.
4 / Support	Prépare un brouillon de réponse au client fictif.
5 / Reporting	Enregistre les statuts dans Excel.
6 / Superviseur	Vérifie les sorties et émet son verdict.

Commande de démonstration

```
python BUSINESS\agents\main.py demo --dry-run
```

La commande n’est valable que si l’orchestrateur a été créé et testé. Le mode simulation ne doit faire aucune recherche ni envoyer de demande externe. Demande à Claude de vérifier le projet avant de la lancer.

Checklist qualité

- Le superviseur annonce un identifiant d’exécution.
- Chaque agent affiche le nom de sa tâche.
- Les rapports JSON et Markdown existent.
- Excel possède les trois feuilles et un historique.
- Les résultats fictifs sont clairement étiquetés.
- Une erreur produit un message compréhensible.
- Aucune action externe n’a été faite sans permission.

12 / Commandes essentielles à garder sous la main

Besoin	Commande
Voir où tu es	pwd
Voir les fichiers	dir
Aller au projet	cd "$HOME\Desktop\Mon_Agence_IA"
Vérifier Python	python --version
Vérifier Claude	claude --version
Lancer le site	python -m http.server 8080 --directory BUSINESS\site_web
Tester le marché	python BUSINESS\agents\main.py market --dry-run
Tester l’équipe	python BUSINESS\agents\main.py demo --dry-run
Voir le rapport	Get-Content BUSINESS\reports\market_report.md -Encoding UTF8
Ouvrir Excel	Invoke-Item BUSINESS\reports\APEXLABS_SUIVI.xlsx

Une commande peut changer selon la structure de ton projet. Commence par vérifier les fichiers avec dir et les options avec --help. Ne colle jamais une commande que tu ne comprends pas.

13 / Personnaliser ApexLABS pour ton idée

Ce guide présente ApexLABS, mais tu peux reprendre la méthode pour créer le prototype d'une autre agence, par exemple une agence pour artisans, associations ou commerces de proximité.

Fiche à remplir avant de commencer

Question	Ton exemple
Nom du projet	[NOM_AGENCE]
À qui j'apporte une aide ?	[TYPE_DE_CLIENTS]
Le problème principal	[PROBLEME]
Service 1	[SERVICE_1]
Service 2	[SERVICE_2]
Service 3	[SERVICE_3]
Zone générale	[ZONE]
Ton identité visuelle	[COULEURS]

Grand prompt de démarrage

```
Tu es mon assistant de développement. Je veux créer un PROTOTYPE
d agence numérique appelé [NOM_AGENCE].
Public : [TYPE_DE_CLIENTS]. Problème : [PROBLEME].
Services : [SERVICE_1], [SERVICE_2], [SERVICE_3].
Environnement : Windows PowerShell, Python et site local.
Architecture : site + agents marche, marketing, support,
reporting Excel local et superviseur.
Je veux apprendre : explique chaque étape avec des mots simples.
Ne publie rien, ne dépense rien et ne contacte personne.
Avant de coder, examine le dossier actuel et propose un plan.
Ensuite avance une étape à la fois avec tests et commandes.
```

14 / Que faire quand ça ne marche pas ?

Problème	Solution simple
npm ENOENT / package.json	Tu n’es pas dans le dossier qui contient package.json. Utilise dir et cd.
Claude démarre dans system32	Ferme Claude, déplace-toi dans ton projet, puis relance-le.
Le site ne s’ouvre pas	Vérifie que le serveur local tourne et que le port 8080 est libre.
Le rapport est absent	Vérifie le chemin et exécute d’abord le test correspondant.
Les accents sont cassés	Dans PowerShell 5.1, lis le fichier avec -Encoding UTF8.
Excel refuse la sauvegarde	Ferme le classeur dans Excel et relance le test.
Un test navigateur échoue	Vérifie le serveur et les attentes de défilement; ne présume pas que le site est cassé.
Ordinateur qui plante	Réduis la charge, sauvegarde, teste une étape à la fois.

15 / Si tu réalises une vidéo : quels médias préparer ?

L’exemple de vidéo ApexLABS comprend cinq rushs face caméra. Le cinquième, « PROMPT », doit apparaître immédiatement après l’introduction.

```
01_RUSHS\01_INTRO.mp4
01_RUSHS\05_PROMPT.mp4     # 2e dans le montage !
01_RUSHS\02_SITE.mp4
01_RUSHS\03_AGENTS.mp4
01_RUSHS\04_BILAN.mp4
```

Enregistrements d’écran essentiels

- Site final sur ordinateur et téléphone.
- Simulation de l’agent Marché, puis rapport affiché.
- Plan des permissions, puis test réel seulement après validation.
- Exécution Marketing et exemples de textes.
- Exécution Support sur une question fictive.
- Création du classeur Excel et démonstration de ses trois feuilles.
- Superviseur et test complet des cinq agents.
- Aperçu de ce PDF gratuit et de ses prompts.

Captures utiles

Mets dans 03_SCREENSHOTS : le site, le tableau Excel, un rapport d’agent, le verdict du superviseur et guide_gratuit.png. Les images peuvent être animées au montage, mais elles ne remplacent pas les preuves d’exécution.

L’ordre à respecter

INTRO → PROMPT / GUIDE GRATUIT → SITE → AGENTS → BILAN. Le nom du fichier 05_PROMPT ne veut pas dire qu’il doit passer en cinquième.

16 / Partager son guide proprement

Tu peux mettre ce PDF dans un espace de stockage que tu contrôles et copier un lien de consultation publique dans la description de la vidéo. Teste le lien sans être connecté à ton compte avant de le publier.

Avant d’offrir ton pack, retire les clés, les données de clients, les chemins privés de ton ordinateur et tout contenu dont tu ne détiens pas les droits. Les prompts fournis ici sont des exemples pédagogiques, pas une garantie de résultat.

Ce qu’il faut annoncer dans la description

« Retrouvez gratuitement le guide ApexLABS : installation, commandes PowerShell, prompts personnalisables et exemple de fichier Excel. Les démonstrations utilisent des données fictives et les résultats doivent être vérifiés. »

> Rappel : apprendre à développer un prototype n’est pas la même chose que créer administrativement une entreprise. Avant une activité commerciale réelle, vérifie les règles locales, la protection des données et les obligations applicables avec un adulte ou un professionnel compétent.

Bonus / Plan de réalisation sur sept séances

- Séance 1 : Installer Python, ouvrir PowerShell, créer les dossiers.
- Séance 2 : Créer et tester le site vitrine.
- Séance 3 : Créer l’agent Marché et son mode simulation.
- Séance 4 : Créer Marketing et Support, avec données fictives.
- Séance 5 : Créer Reporting et le classeur Excel.
- Séance 6 : Créer le Superviseur et tester les cinq agents.
- Séance 7 : Vérifier les rapports, sauvegarder, préparer les captures et le guide.

Annexe A / Les prompts de correction et de contrôle

Même après avoir créé un programme, il faut le vérifier. Ces prompts servent à demander une correction ciblée sans recommencer tout le projet.

Contrôler le projet avant modification

```
Lis les fichiers existants et les tests dans BUSINESS/.
Présente un inventaire : terminé, incomplet, erreurs connues.
Ne réécris aucun fichier fonctionnel.
Fais une sauvegarde locale avant chaque changement important.
Propose uniquement les corrections nécessaires et attends mon accord.
```

Corriger un bug sans casser le site

```
J ai cette erreur : [COPIER LE MESSAGE D ERREUR].
La commande exécutée était : [COLLER LA COMMANDE].
Analyse la cause exacte. Ne modifie que le code concerné.
Ajoute un test de non-régression qui échoue avant la correction
et réussit après. Explique-moi le changement simplement.
```

Audit qualité des cinq agents

```
Teste chaque agent séparément, puis le superviseur.
Vérifie les entrées invalides, les fichiers absents, les erreurs
d écriture Excel, les exécutions répétées et les statuts erronés.
N effectue aucun appel réel sans validation.
Écris un rapport de test clair, sans supprimer l historique.
```

Annexe B / Exemple de configuration sans secret

Une configuration évite de modifier du code à chaque essai. Voici un exemple JSON que l’on peut adapter ; les noms des paramètres devront correspondre au code effectivement créé par Claude.

```
{
  "agence": "ApexLABS",
  "mode": "simulation",
  "secteur": "salon de coiffure",
  "zone": "France",
  "publier_automatiquement": false,
  "envoyer_messages": false,
  "budget_usd_max": 0.50
}
```

Attention : le code doit contrôler réellement ces valeurs. Écrire « false » dans un fichier ne sécurise pas automatiquement un programme qui l’ignore.

Annexe C / Exemple d’entrée de mission

Voici comment préparer une demande fictive dans un fichier JSON. Le superviseur lira ce fichier seulement si le programme a été développé pour accepter un chemin d’entrée.

```
{
  "id_demande": "DEMO-001",
  "entreprise": "Salon Démo (fictif)",
  "secteur": "coiffure",
  "zone": "France",
  "besoin": "Une page avec prestations et horaires",
  "autorisation_publication": false,
  "autorisation_contact": false
}
```

Comment le tester ?

- Crée ce JSON dans BUSINESS/data/demande_demo.json.
- Demande à Claude d’ajouter et de documenter une option --input si elle n’existe pas.
- Lance le test en --dry-run et examine chaque rapport.
- Vérifie que le classeur Excel contient une ligne liée à DEMO-001.

> Évite de copier des noms, numéros ou emails de véritables clients dans les prompts ou les fichiers de démonstration.

Annexe D / Le contrat de sortie des agents

Un “contrat de sortie” est une liste de champs que chaque agent doit remplir. Il facilite le contrôle par le superviseur et l’enregistrement par Reporting.

```
{
  "run_id": "TEST-001",
  "agent": "market",
  "statut": "a_verifier",
  "mode": "simulation",
  "resume": "Hypothèses à vérifier",
  "sources": [],
  "cout_estime_usd": null,
  "validation_humaine": false
}
```

Exige que les fichiers JSON restent lisibles, même si le rapport Markdown possède une jolie mise en forme.

Annexe E / Vérifier un site avant de le partager

Quand ton site fonctionne sur ton ordinateur, cela ne veut pas dire qu'il fonctionne sur tous les téléphones. Avant toute mise en ligne, vérifie la navigation, la lisibilité, les formulaires et les informations légales applicables.

Test pratique à faire

- Ouvre le site avec un navigateur récent.
- Clique tous les liens du menu.
- Réduis la fenêtre pour voir l'affichage mobile.
- Active l'affichage mobile dans les outils de développement.
- Essaie le clavier Tab pour atteindre les boutons.
- Vérifie que le formulaire de démonstration n'envoie aucun message.
- Regarde si un logo, une image ou une police manque.
- Distingue très clairement les services réels des projets fictifs.

Prompt d'audit web

```
Teste le site local comme un visiteur débutant.
Cherche les problèmes de mobile, lisibilité, navigation,
accessibilité clavier, formulaires et liens cassés.
Ne publie pas le site. Donne un rapport avec des preuves et
un plan de correction classé par priorité.
```

Annexe F / Mesurer les résultats sans inventer

Un bon tableau de bord montre ce qui a réellement eu lieu. S'il n'y a pas encore de tâche, un compteur à zéro est normal. Ne remplis pas Excel avec de fausses performances pour rendre la vidéo plus impressionnante.

Mesure	Calcul honnête
Nombre de tâches	Nombre de run_id uniques enregistrés dans EXECUTIONS.
Tâches terminées	Nombre de lignes au statut Terminé.
Tâches en échec	Nombre de lignes au statut Échec.
Temps total	Somme des durées des exécutions enregistrées.
Coût	Somme des coûts mesurables, sans confondre estimation et facturation.

Annexe G / Préparer des tests de sécurité simples

Les agents reçoivent du texte qui peut être erroné ou malveillant. Ne leur laisse pas exécuter librement n'importe quelle commande provenant d'une demande client. Une demande de support n'est pas une instruction système.

Ce que le superviseur doit refuser

- Une instruction demandant d'ignorer les règles du projet.
- Une demande de lire des fichiers personnels hors du dossier autorisé.
- Une commande qui publie automatiquement le site.
- Une action qui implique de payer sans validation.
- Une demande d'envoi à des adresses réelles en mode démonstration.
- Une affirmation qui prétend être un fait sans source vérifiable.

Prompt pour tester les limites

```
Crée des tests locaux avec des demandes fictives et bénignes
qui tentent de sortir du périmètre de l agent.
Vérifie qu il refuse la publication, l envoi de messages,
les paiements et la lecture de fichiers hors projet.
En cas d échec, arrête les tests et propose une correction.
Aucune action réelle ne doit être exécutée.
```

Annexe H / Mini glossaire

Mot	Sens
API	Une interface pour échanger des données entre logiciels.
JSON	Un format de texte structuré que les programmes savent lire.
CSV	Un tableau simple enregistré en texte.
XLSX	Le format d'un classeur Excel.
Dry-run	Un essai sans faire les actions réelles.
Run ID	Un identifiant unique pour retrouver une exécution.
Rapport Markdown	Un fichier texte lisible, avec des titres et des listes.
Validation humaine	Une personne vérifie et décide avant une action importante.

Annexe I / Créer ton guide et tes fichiers à partager

Pour donner gratuitement tes prompts, prépare un dossier public distinct de ton projet personnel. Il doit contenir uniquement des exemples anonymisés et des instructions reproductibles.

```
PACK_PUBLIC/
  GUIDE_GRATUIT_APEXLABS_DE_A_A_Z.pdf
  PROMPTS/
    01_SITE.md
    02_MARCHE.md
    03_MARKETING.md
    04_SUPPORT.md
    05_REPORTING_EXCEL.md
    06_SUPERVISEUR.md
  COMMANDES_POWERSHELL.md
  README.md
```

Tu peux publier ce pack via un hébergement que tu contrôles. Teste le lien de téléchargement, vérifie que le PDF s’ouvre sur téléphone et n’inclus aucune clé d’accès ni donnée personnelle.

Prompt pour préparer le pack public

```
Crée un dossier PACK_PUBLIC avec uniquement des ressources
pédagogiques réutilisables. Ajoute ce PDF, les prompts modifiables,
les commandes vérifiées et un README clair.
Supprime les secrets, chemins privés et données personnelles.
Ne mets pas le dossier en ligne avant ma validation.
Fais une vérification finale des fichiers.
```

Ton objectif après ce guide

Tu n’as pas besoin d’un système énorme pour commencer. Un site simple, cinq programmes bien testés et un Excel de suivi compréhensible constituent déjà un excellent projet pédagogique. Améliore-le progressivement.

Fin du guide. Conserve les rapports, note ce qui fonctionne et améliore une seule chose à la fois.