

Prompt 1:

Fais-moi un script PowerShell (.ps1) pour Windows, relançable sans risque, qui prépare un environnement Python pour l'API MetaTrader5 :

1. Vérifie Python 3.10 64-bit. Absent → télécharge et installe en silencieux, en profil utilisateur, sans droits admin. Refuse le 32-bit.
2. Vérifie pip. Cassé → répare via ensurepip, puis met à jour pip.
3. Vérifie et installe si besoin : MetaTrader5, numpy, pandas, scipy, pytz. Affiche les versions.
4. Détecte le terminal MT5 (terminal64.exe) via le registre et le disque. N'installe PAS MT5. Si absent, arrête tout avant les autres installs et explique qu'il faut installer le terminal de son broker et l'avoir lancé une fois.
5. Affiche un résumé final avec les points bloquants.

Options : -CheckOnly (audit seul), -TestMT5 (teste la connexion au terminal et le symbole XAUUSD).

Fichier en ASCII pur (encodage PowerShell 5.1). Ajoute un .bat de lancement à double-clic avec -ExecutionPolicy Bypass.

Prompt 2:

Fais-moi un script Python mt5_check.py qui audite un compte MT5 avant de développer une stratégie :

1. Se connecte au terminal MT5 en détectant automatiquement le chemin de terminal64.exe. Reprend la session en cours si le terminal est déjà connecté, sinon utilise login / password / serveur (mot de passe en variable d'environnement, jamais en dur). Vérifie que l'AlgoTrading est actif.
2. Affiche le compte : login, serveur, devise, solde, equity, levier, demo ou réel.
3. Liste les paires forex du broker et résume en une ligne le ou les prefixes et suffixes exacts utilisés (EURUSD, EURUSD.a, EURUSD.pro...), avec digits, spread et lot minimum.
4. Sur EURUSD, en résolvant son vrai nom chez le broker, affiche uniquement : la date de la barre M30 la plus ancienne disponible, la profondeur d'historique en années, et le décalage horaire du serveur par rapport à UTC. Pas d'export CSV, pas de chargement complet en

memoire.

Donne-moi la commande notepad, le code complet, et la commande de lancement.

Prompt 3:

Fais-moi un EA Python pour MetaTrader 5 (fichier unique xau_fvg_ea.py).

1. STRATEGIE

Instrument : XAU/USD, timeframe M15.

Biais : direction du dernier BOS en H1 (structure par fractales).

Entree : ordre limit sur retest d'un FVG M15 dans le sens du biais.

Un FVG haussier est un gap $low[i] > high[i-2]$; baissier si $high[i] < low[i-2]$. La couleur des bougies est ignoree.

Filtres : taille du FVG ≥ 3 % de l'ADR14 D1, spread $\leq 0,40$ \$, une seule position a la fois, pas de reentree sur un FVG deja utilise.

Session : 09h-22h heure de Paris, DST gere, pas le week-end.

2. RISQUE

1 % de l'equity par trade.

SL fixe 5,00 \$ (50 pips or).

TP1 a 5,00 \$: ferme 50 % du lot et deplace le SL au BE + 0,10 \$.

Runner : trailing sur le dernier swing M15, cloture forcee a 22h.

Kill switch : arret jusqu'au lendemain a -3 % sur la journee, 5 trades par jour maximum.

Magic number dedie : ne touche jamais aux positions manuelles.

3. CONNEXION

Chemin terminal force via `TERMINAL_PATH`, avec repli sur la detection auto si le chemin n'existe pas.

`DEMO_ONLY` : refuse de demarrer sur un compte reel.

`DRY_RUN` : bascule simulation / execution réelle.

Resolution automatique du suffixe broker du symbole.

Verification du `stops_level` avant de placer le SL.

4. LANCEMENT

`AUTO_DETACH` : le script se relance dans une fenetre PowerShell dediee et rend la main tout de suite.

Verrou reseau anti double-lancement.

Logs console + fichier `xau_fvg_ea.log`.

Contraintes : Python 3.10, librairie MetaTrader5, messages de log en ASCII pur (console Windows). Donne-moi le fichier complet.

Prompt 4 :

Rends mon EA Python MT5 capable de tourner 24/5 sans surveillance.

Ajoute :

1. Une boucle principale incassable : toute exception est loguee et l'EA repart apres 30 s. Au bout de 20 erreurs consecutives, il ferme les positions et s'arrete proprement.
2. Une reconnexion automatique si le terminal MT5 devient injoignable (reseau coupe, terminal ferme).
3. Une ligne de vie toutes les 15 min dans le log, au format :
VIVANT | heure Paris | equity | nb positions | nb ordres | en session
4. Un script PowerShell check-ea.ps1 qui affiche en une commande :
 - le processus Python actif ou non, avec son temps de fonctionnement
 - un avertissement si plusieurs processus Python tournent
 - le terminal MT5 actif ou non
 - la fraicheur du fichier de log (alerte au-dela de 20 min)
 - les 15 dernieres lignes du log
 - le nombre d'erreurs dans le log completSortie coloree : vert si OK, rouge si bloquant.

