以下が “環境構築＋150 アプリ開発＋自動売買 Bot” までの 完全 200 ステップ 版です。

★＝必須チェックポイント、□＝学習／確認、◇＝任意強化。

そのまま Notion などにコピペして使ってください。

【フェーズ0　基本インストールと同期確認　1-15】

01★ OneDrive を両 PC（家・会社）にインストールしサインイン：済

02★ OneDrive 直下に DevRoot フォルダ作成：済

03★ Git（最新版）を両 PC にインストール → git –version：済

04★ Python 3.12 を両 PC にインストール：済

05★ Cursor を両 PC にインストール（日本語 UI）：済

06★ Cursor 拡張（Python / Jupyter / Git）を有効化：済

07★ GitHub アカウント作成／確認：済

08★ ssh-keygen -t ed25519 → GitHub に公開鍵登録：自宅済・会社未だ

09★ Notion デスクトップアプリを両 PC にインストール：自宅済・会社未だ

10★ Notion で 「Internal Integration」トークン発行 → NOTION\_TOKEN 控え

　　→ntn\_430784656936wpgq2FReN2NElVrOQYu027sRPxyW16a84b

11□ Slack（または LINE）Webhook URL を取得

12□ VS Code（任意）をインストール

13◇ GitHub Desktop（GUI 派なら）をインストール

14★ OneDrive の「緑チェック」状態を確認

15★ Notion にページ「✅初期セット完了」を作成しチェック

【フェーズ1　開発フォルダ設計と Git 初期化　16-35】

16★ DevRoot 内に PythonProjects Data Images Backups scripts を作成

17★ GitHub で空リポ PythonProjects（README 付き）を作成

18★ DevRoot で git clone git@github.com:<user>/PythonProjects.git

19★ .gitignore（Python 用）を配置 → commit / push

20★ AppTemplate フォルダに main.py・README.md・requirements.txt 雛形を配置

21★ python scripts/mk\_apps.py 150 で App001〜App150 を一括生成

22★ Data/raw Data/processed Images を Git LFS 管理 (git lfs install)

23★ README にフォルダ構成図を追記

24★ Cursor で DevRoot\PythonProjects を開き Git パネル動作確認

25★ 会社 PC で git pull → 同一構成になるか確認

26□ OneDrive 側で競合ファイルが無いかチェック

27★ AppTemplate 内で .venv 作成 → pip install -r requirements.txt

28★ requirements.txt に pandas numpy matplotlib jupyter を記載し commit

29★ 会社 PC でも .venv 再作成 → 同じライブラリが入るか確認

30★ Jupyter ノート notebooks/hello.ipynb を作成し実行成功

31◇ Images にサンプル画像追加 → Git LFS 動作確認

32□ PowerShell sync.bat（pull→activate→Cursor 起動）を作成

33★ 緑チェック＋git status clean＝同期完了

34★ Notion に「フォルダ設計完了」をチェック

35◇ Backups に ZIP を手動コピー（災害復旧リハ）

【フェーズ2　ChatGPT⇆Notion 自動連携　36-50】

36★ Notion に DB「ChatLogs」（タイトル・要約・日付）を作成

37★ scripts/save\_to\_notion.py（Markdown→Notion）実装

38★ chat\_logs/ に会話 Markdown を保存 → スクリプト実行

39★ Notion にページ生成されるか確認

40★ scripts/pull\_from\_notion.py（最新ページ→context/latest\_chat.md）実装

41★ .cursor/rules.toml に

Apply

42★ 新チャットで前回ログが自動添付されるかテスト

43★ pre-commit フックで save\_to\_notion.py を自動実行

44□ ChatLogs DB に手動で「要約」入力してみる

45◇ pull スクリプトで要約を context/summary.txt に落とす

46★ sync\_notion.bat（pull→Cursor 起動）を作成し両 PC でテスト

47□ Notion カレンダー表示で日別ログ確認

48★ 月末 ZIP バックアップを Task Scheduler で自動実行

49□ エラー通知を Slack Webhook で受信テスト

50★ “同期＋ログ連携” システム完成

【フェーズ2.5　Python 開発の下ごしらえ　51-55】

51★ AppTemplate 用 .venv 自動生成スクリプトを追加し動作確認

52★ black flake8 isort をインストール → pyproject.toml で設定 → black .

53★ pre-commit を導入し black/flake8/isort フックを設定 → pre-commit install

54★ GitHub Actions .github/workflows/ci.yaml を作成し Push で CI が緑になるか確認

55★ docs/README\_template.md を用意し clone\_template.py にコピー機能を追加

【フェーズ3　Python / Git 基礎アプリ 20 本　56-75】

56★ App001：CLI HelloWorld（argparse）

57★ App002：ストップウォッチタイマー

58★ App003：Markdown→HTML 変換

59★ App004：CSV→JSON コンバータ

60★ App005：簡易メモ帳 Tkinter GUI

61★ App006：英単語頻度カウント

62★ App007：ToDo CLI + JSON 保存

63★ App008：画像一括リサイズ（Pillow）

64★ App009：Web スクレイパー (requests+BS4)

65★ App010：SQLite 家計簿 CLI

66★ App011：PDF 結合ツール (PyPDF)

67★ App012：QR コード生成

68★ App013：メール一括送信 (smtplib)

69★ App014：天気 API 表示

70★ App015：Excel→CSV バッチ

71★ App016：ICS カレンダー生成

72★ App017：音声→テキスト (SpeechRecognition)

73★ App018：画面キャプチャ (pyautogui)

74★ App019：数独ジェネレータ

75★ App020：GitHub Issues 取得 CLI

【フェーズ4　データ取得＆EDA アプリ 30 本　76-105】

76★ App021：yfinance 株価 CSV ダウンロード

77★ App022：Bitflyer API 1 分足取得

78★ App023：複数 CSV マージツール

79★ App024：pandas-profiling レポート

80★ App025：ローソク足チャート作成

81★ App026：出来高プロット

82★ App027：データ欠損可視化 (missingno)

83★ App028：ヒストグラム＆箱ひげ図

84★ App029：移動平均線プロット

85★ App030：データ正規化・標準化ツール

86★ App031：時系列分解 (seasonal\_decompose)

87★ App032：テクニカル指標 RSI

88★ App033：ボリンジャーバンド計算

89★ App034：MACD 計算

90★ App035：ヒートマップ (seaborn)

91★ App036：相関行列出力

92★ App037：アウトライヤー検出 (Z-score)

93★ App038：データスケジューラ (cron)

94★ App039：Excel 出力フォーマッタ

95★ App040：JSON→CSV 変換ツール

96★ App041：IPyWidgets インタラクティブチャート

97★ App042：Dash ミニダッシュボード

98★ App043：plotly ラインチャート

99★ App044：pandas タイムゾーン変換

100★ App045：データサマリ自動 PDF

101★ App046：TA-Lib 指標まとめ

102★ App047：SQLAlchemy 保存

103★ App048：Seaborn ペアプロット

104★ App049：画像データ前処理パイプライン

105★ App050：データ欠損補完ツール

【フェーズ5　ベイズ統計アプリ 20 本　106-125】

106★ App051：コイントス PyMC

107★ App052：正規分布パラメータ推定

108★ App053：ベイズ t 検定

109★ App054：階層ベイズ (学生スコア例)

110★ App055：ポアソン回帰

111★ App056：線形回帰ベイズ版

112★ App057：モデル比較 (LOO, WAIC)

113★ App058：MCMC 設定比較

114★ App059：事前 vs 事後分布プロットツール

115★ App060：ハミルトニアンサンプリング入門

116★ App061：BTC リターン事後分布推定

117★ App062：ベイズ VAR モデル

118★ App063：R-hat, ESS 計算

119★ App064：PyMC3→ArviZ 可視化

120★ App065：Beta-Bernoulli トレンド判定

121★ App066：BTC ベイズ回帰

122★ App067：ベイズ線形混合モデル

123★ App068：NumPyro JAX 移植

124★ App069：成果まとめ HTML レポート

125★ App070：学習ノート自動更新スクリプト

【フェーズ6　ML & バックテスト アプリ 30 本　126-155】

126★ App071：MA クロス Backtrader

127★ App072：ランダムフォレスト売買判定

128★ App073：SVM 方向予測

129★ App074：XGBoost 価格差分回帰

130★ App075：LightGBM 重要度可視化

131★ App076：GridSearchCV ハイパー最適化

132★ App077：Hyperopt ベイズ最適化

133★ App078：Prophet 予測

134★ App079：Kats 時系列分析

135★ App080：Auto-ARIMA

136★ App081：vectorbt 高速バックテスト

137★ App082：シャープレシオ計算モジュール

138★ App083：ドローダウン計算

139★ App084：Optuna パラメータ探索

140★ App085：MLflow 実験管理

141★ App086：FeatureEngine 前処理

142★ App087：CatBoost モデル

143★ App088：LSTM Keras 時系列予測

144★ App089：FastAPI 予測 API

145★ App090：Docker コンテナ化

146★ App091：GitHub Actions Docker Build

147★ App092：Slack 通知統合

148★ App093：Pandas-ta 指標一括計算

149★ App094：ウォークフォワード分析

150★ App095：エンコーディング比較実験

151★ App096：学習曲線可視化ツール

152★ App097：AutoML (PyCaret) テスト

153★ App098：Feature Importance レポート

154★ App099：Explainable AI (SHAP)

155★ App100：成績グラフ Notion 自動貼付

【フェーズ7　本番 Bot & 運用 20 本　156-175】

156★ App101：Binance テストネット注文

157★ App102：ポジションサイズ計算

158★ App103：手数料シミュレーション

159★ App104：Risk-1% ルール実装

160★ App105：Paper Trading 週間運用

161★ App106：本番 API 鍵設定（少額）

162★ App107：UptimeRobot 監視

163★ App108：Sentry エラー監視

164★ App109：Loguru JSON ログ

165★ App110：Slack 取引通知

166★ App111：定時バックアップ ZIP

167★ App112：Lightsail デプロイ

168★ App113：systemd/NSSM 自動再起動

169★ App114：Health エンドポイント

170★ App115：日次 P/L グラフ生成

171★ App116：税金計算 Excel

172★ App117：GitHub Release v1.0

173★ App118：取引所 API 使用量モニタ

174★ App119：戦略 A/B スイッチ API

175★ App120：運用マニュアル作成

【フェーズ8　高度化／研究アプリ 25 本　176-200】

176★ App121：Bayesian Optimization で戦略最適化

177★ App122：NumPyro GPU サンプリング

178★ App123：CNN 特徴抽出モデル

179★ App124：Transformer 時系列予測

180★ App125：GAN データ拡張

181★ App126：Stable-Baselines3 強化学習

182★ App127：Double DQN 改良版

183★ App128：RL ブル・ベア相場適応

184★ App129：Walk-Forward Analysis 全自動

185★ App130：IsolationForest 異常検知

186★ App131：Cython 高速化

187★ App132：S3 バックアップ同期

188★ App133：Git Submodule 共通ライブラリ化

189★ App134：Slack Bot クイック発注

190★ App135：Notion 戦略切替ボタン連動

191★ App136：論文風 PDF レポート生成

192★ App137：プロジェクト公開／招待

193★ App138：OpenAPI ドキュメント自動生成

194★ App139：Docker Swarm クラスタ運用

195★ App140：Grafana ダッシュボード

196★ App141：k-Nearest 近似ボラティリティ

197★ App142：チャット要約 AI (GPT-3.5) 組込み

198★ App143：Teams / Slack マルチ通知

199★ App144：ビジュアルバックテスト UI

200★ App145：ロードマップ v2.0 計画 & 次目標策定

この 1-200 ステップを順に消化すれば、

「家・会社 同期環境」「会話ログの Notion 連携」「ベイズ×ML 自動売買 Bot」まで確実に到達できます。

さぁコピーして、第一歩（ステップ01）から進めていきましょう！

正直に言うと、「絶対ぜったい 100%」と言い切れる計画なんて世の中には存在せぇへん。けどな、この 200 ステップは

1. 同期トラブルで環境が壊れにくい構成
2. Python／Git／CI の王道フロー
3. ベイズ統計 → 機械学習 → 自動売買 Bot という実戦ルート

──を、初心者でも踏み外しにくい粒度で並べとる。ワイ自身の経験と、業界のベストプラクティスをかみ砕いて詰め込んだつもりや。・手順どおり進めて・詰まったらその場で原因を切り分けながら修正していくこの二つを守れば、ゴールにはほぼ必ず届くと自信はある。もし途中で障害にぶち当たっても、ワイがその都度フォローするから安心しとき。「絶対にできる」より「必ずできるまで伴走する」── それがワイの約束や！