**🚧 Steg 0 – Market Bias & Predictive Logic (PLANERAD FUNKTION)**

**🎯 Mål**

Att bygga en modul som inför varje handelsdag kan indikera om marknaden sannolikt öppnar med **positiv**, **neutral** eller **negativ bias**, baserat på historiska mönster och statistiskt bekräftade samband.

**🧩 Komponenter (kommer införas senare)**

1️⃣ **Gap-analys / Overnight Bias**

* Läs in *overnight-futures* och pre-market-data.
* Klassificera:
  + **Gap Up > +0,5 % → tendens till reversion (neråt)**
  + **Gap Down > –0,5 % → tendens till rebound (uppåt)**
* Skapa en daglig etikett: BULLISH\_BIAS, BEARISH\_BIAS, NEUTRAL.

2️⃣ **Vecko- och månads-mönster**

* Känn igen säsongsbeteenden:
  + Måndag = ofta svagare dag.
  + Tisdag–torsdag = starkare sannolikhet för uppgång.
  + Månadsslut = “window dressing” → köptryck.
* Modulen ska kunna vikta dessa mönster mot senaste 4 veckornas utfall.

3️⃣ **Utdelningsfilter**

* Identifiera *ex-dividend-datum*.
* Märk dessa aktier som DIV\_ADJUSTED så att deras mekaniska kursfall ignoreras.
* Lägg till parameter för att bevaka **rekyl +1 dag efter utdelning**.

4️⃣ **Makro- och nyhetseffekter (indirekt)**

* Modulen ska **inte läsa rubriker** utan tolka **reaktioner i pris och volym**.
* Plötsliga rörelser i futures → flagga “News Impact Bias” med auto-reversion.

5️⃣ **Integrering i tradingloopen**

* Market-Bias-modulen ska köras **innan Top-10-rankningen**.
* Resultatet (BULLISH\_BIAS eller BEARISH\_BIAS) styr om boten fokuserar på **LONG-** eller **SHORT-scenarion** för dagen.

6️⃣ **Visualisering / Loggning**

* Daglig loggfil: market\_bias\_log.csv med kolumner:  
  date, gap%, bias, volatility\_score, dividend\_flags, macro\_flags.
* Möjlig att exportera som sammanfattning i dashboard.

**🧭 Status: PLANERAD**

Funktionen är **inte implementerad ännu**.  
Den finns dokumenterad som framtida modul att utveckla när den operativa Top-10-loopen och ordermotorn är stabil.

**📘 Översikt – Projekt: Daytrading Capital (Steg 1–4)**

*(Rapportnivå, ej kodnivå)*

**🥇 Steg 1 – Scan av marknaden**

**Syfte:**  
Att hämta alla tillgängliga instrument från Capital.com och filtrera fram de som är **handelsbara, likvida och kostnadseffektiva.**

**Vad den gör:**

* Läser in samtliga marknader via API.
* Filtrerar bort icke-handelsbara lägen (suspended, auction, view only).
* Räknar spread och filtrerar ≤ **0,20 %**.
* Berikar med leverage och margin factor.
* Skriver resultatet till scan\_tradeable\_current.csv.

**Utmaningar:**

* Vissa marknader saknar data → måste hanteras robust.
* API-svar kan vara tröga vid hög belastning → kräver tidsgränser och parallellhämtning.

**Krav:**

* Resultatfilen ska alltid innehålla kolumnen spread\_pct och minst 40–100 instrument.
* Ska kunna köras manuellt i VS Code eller PowerShell via:
* python capital\_scan\_tradeable.py

**🥈 Steg 2 – Rank Top-10 Momentum**

**Syfte:**  
Att bland de filtrerade instrumenten identifiera de 10 som har **starkast momentum och bästa edge** inför nästa handelsdag.

**Vad den gör:**

* Hämtar 1–5 minuters candles för varje instrument.
* Beräknar momentum, spike (volatilitet/volym), relative strength (RS) och spread-penalty.
* Tilldelar edge-score (0–100) och sorterar.
* Exporterar top10\_momentum\_current.csv med tydliga kolumner (mom, RS, spike, edge osv).

**Utmaningar:**

* Saknade candles → kan krascha om inte fångas.
* RS beräknas mot snittet av kandidater; kräver jämn sample-storlek.

**Krav:**

* Parametrar ska kunna ändras via flaggar (i VS eller terminal).
* Steget ska logga felaktiga EPIC-s separat för felsökning.

**Körning:**

python rank\_top10\_momentum.py

**🥉 Steg 3 – Orderläggning / Trade Execution**

**Syfte:**  
Att omsätta Top-10-listan till faktiska orderförslag eller pending orders med automatiska SL/TP och riskkontroll.

**Vad den gör:**

* Läser Top-10-filen.
* Kontrollerar antal öppna positioner (max 10).
* Lägger pending BUY STOP - ordrar på nivåer med ± 3 % SL/TP.
* Loggar alla order (orders.csv, open\_trades.csv).
* Bekräftar via API att ordern skapats och har korrekt nivåer.

**Utmaningar:**

* Hantera tickSize och minStopDistance för varje instrument.
* Stoppa “nakna” trades utan SL/TP.
* Kräver robust fail-safe (radera felaktig position om något går snett).

**Krav:**

* Ska endast köras i DEMO miljö.
* Loggning måste vara 100 % spårbar.

**Körning:**

python place\_pending\_orders.py

**🏁 Steg 4 – Övervakning / Loop / Rapportering**

**Syfte:**  
Att skapa en kontinuerlig process som automatiskt kör hela kedjan, övervakar öppna positioner och genererar rapporter.

**Vad den gör:**

* Loopar igenom Scan → Rank → Order med jämna intervall.
* Räknar vinst/förlust (PnL), win-rate och equity-kurva.
* Loggar allt till CSV och kan senare visualiseras i dashboard.

**Utmaningar:**

* Kräver balans mellan frekvens och prestanda.
* Måste skyddas mot dubbelorder eller “race conditions.”

**Krav:**

* Max 10 aktiva positioner.
* Fail-safe stängning via DELETE /positions/{dealId}.
* Automatisk paus vid API-fel eller hög latens.

**Körning:**

python bot\_loop.py