勉強環境セットアップ

1 導入

この講義で最重要なのは、**最初に記法と運用規則を固めて、あとで全体を崩さないこと**です。
学習ノートは1枚だけなら多少書き方がぶれても困りません。しかし枚数が増えると、表記のぶれ、リンク形式のぶれ、frontmatter の欠落がそのまま管理の難しさになります。だからセットアップでは、書き始める前に基準を決めます。

2 用語と定義

まず主役になる用語をそろえます。
Gloss[読/よ]みつき[表記/ひょうき] とは、漢字の語に読みを添えて、本文を読みやすくするための書き方です。
frontmatter とは、ノートの先頭にある管理情報です。この環境では date、tags、type、status、related、source_answer、source_scan をそろえます。
関連カード Card link とは、本文の中で他のノートへつなぐ参照です。この環境では @card の形式を使います。

3 方針

環境セットアップでは、まず見本を1つ決めて、そこから外れないように書くのが基本です。この環境では sample.n.md がその基準です。
つぎに、ノート本体と運用の役割分担をはっきりさせます。ノートには内容を書き、prompt には書き方の規則を書き、tools と src-cli には生成や lint の責務を持たせます。こうしておくと、どこを直せば全体に効くかが分かりやすくなります。

4 導出

4.1 1. なぜ見本を先に決めるのか

書き方を後からそろえるのは、最初にそろえるよりずっと大変です。なぜなら、表記のずれは1箇所ではなく、全てのノートに散らばるからです。
だから sample.n.md を見本にして、Gloss、数式、frontmatter、関連リンクの書き方を最初に固定します。こうすると、新しいノートを足すたびに迷わずに済みます。

4.2 2. なぜ frontmatter を 省略 しないのか

frontmatter は見た目のためではなく、管理のためにあります。date があれば最新更新日が分かり、tags があれば分野ごとに探せて、related があれば内容のつながりを追えます。
つまり frontmatter を欠くと、そのノート 1 枚だけの問題ではなく、一覧化、検索、関連付けまで崩れます。
だから必須項目は 省略 しません。

4.3 3. なぜ本文リンクと運用ツールを分けるのか

本文では @card を使って人間が読める 形 でつなぎます。一方で、lint や index 更新は tools と src-cli に任せます。
この分担を守ると、内容を直す作業と機械的な整形を分離できます。その結果、ノートはノートとして読みやすく、運用は運用として安定します。

5 どこまで成り立つか

この方針は、ノートが少ない段階でも有効ですが、数が増えるほど効果が大きくなります。
ただし、規則だけ決めても守られなければ意味がありません。だから prompt に方針を書き、lint で機械的に検査し、手動確認で 補 うという 3 段階で維持します。

6 最終形

勉強 環境 のセットアップでは、見本、prompt、lint、運用ツールの役割を先に決めます。
frontmatter は 省略 せず、関連リンクは @card でそろえ、本文の書き方は sample.n.md に合わせます。

7 一言でいうと

- 最初に基準を決めると、あとで全体を直す手間が減ります。
- 内容はノートに、規則は prompt に、検査は lint に分けるのが基本です。
- 管理しやすい 学習 環境 は、書き方をそろえるところから始まります。

8 関連リンク

関連講義・関連ノート

→ README
https://study.bem130.com/README/

→ index
https://study.bem130.com/index/

元の解答

なし

スキャン画像(がぞう)

なし