

プログラミング学習の心得

コード is 量！

自己紹介 (Twitter: @taroosg)

```
{
  "name": "大杉太郎 (たろさん・たろ先生) ",
  "works": [
    {
      "work": "🎓 G's ACADEMY FUKUOKA 主任講師",
      "skills": ["JavaScript", "React", "PHP", "Laravel"]
    },
    {
      "work": "🎓 エンジニア",
      "skills": ["Laravel", "JavaScript", "画面設計", "DB設計"]
    }
  ],
  "like": ["💻", "📚", "✈️ 🚗 🏠 🚶", "🍹 🍷 🍺", "👤"]
}
```

はじめに

せっかくチャレンジするのだから，成果を出しましょう！！

成果を出すためにはどうするのかを共有します！

学ぶことが目的ではない

- 「アイデアを実現するプロダクトをつくる」
- 「プロダクトをつくれる人間になる」

プロダクトをつくれるようになるには. . .

- 「プロダクトをつくる」しかない
- 時間が必要
- 量 >>> 越えられない壁 >>> 質

「量」が大事！！！！

もくじ

- 時間はなぜ必要か
- 時間をどうやって使うか
- 時間をどうやって作るか
- まとめ

時間が必要な理由

人間の構造が原因

よくある例

- Q1：コードが理解できません！！
- Q2：なんで動かないんですか！！
- A：（人間の）仕様です！

「理解」とはなにか

理解 = 「多くの具体例から法則を見出すこと」

- 「コード」と「動作結果」から法則を見出す
- わからないときは「具体例」が足りない

そのため．．．

「時間」が具体例の「量」と「多様性」を生み出す！

コードがわかるようになるには時間が必要！

もくじ

- ~~時間はなぜ必要か~~
- 時間をどうやって使うか
- 時間をどうやって作るか
- まとめ

時間をどうやって使うか

時間が必要なことはわかった。

時間をどのように使えばいいのか．．??

「めっちゃコード書きまくればいい」ではない！！

プロダクトをつくるときの雑な流れ

1. つくりたいものを決める.
2. 必要な処理を細かく分解する.
3. 試したり調べたりしながら実装する.

「分解」が大事

「分解」が大事

- 「複雑な処理」は「単純な処理」の組み合わせ.
- 単純な処理なら書ける.
- 簡単な処理に分解できれば地道に実装すれば OK !

つまり. . .

時間の使い方には 2 種類ある.

- よく考えたり日本語で書き出したりで「分解」する時間.
- 分解した内容をコードで書く時間.

「分解の時間」 「コードの時間」 がある！

もくじ

- ~~時間はなぜ必要か~~
- ~~時間をどうやって使うか~~
- 時間をどうやって作るか
- まとめ

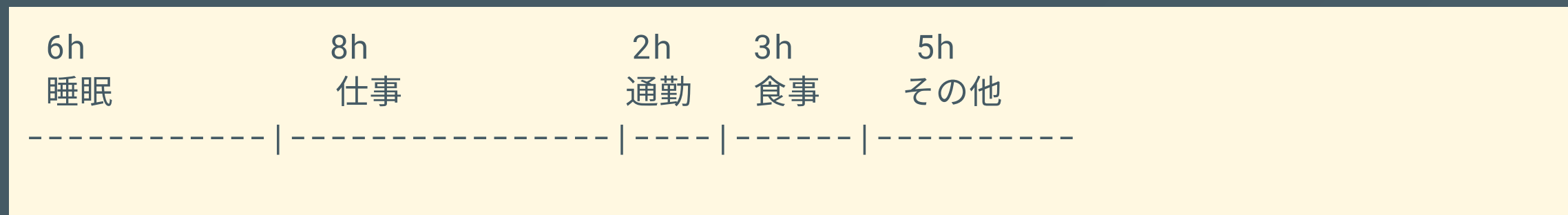
時間をどうやって作るか

時間はない。

人間の 24 時間は埋まっている。

引き算だッ！

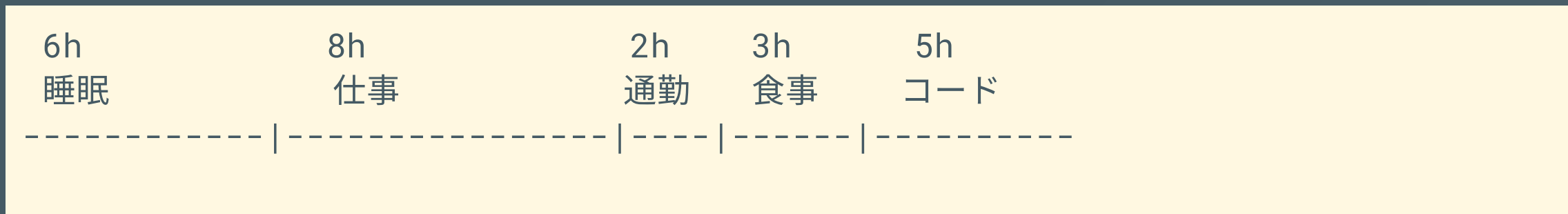
1日のスケジュールを見てみましょう！



この中になくてもいい時間はあるでしょうか？

なくともいい時間をなくす

「その他」をコードに充ててみましょう。

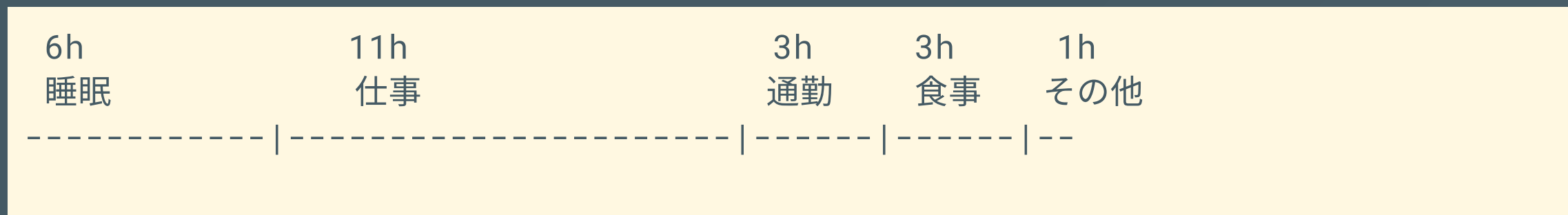


いい感じですね！

「やらないこと」を決めるのが大事！

とはいえ．．．

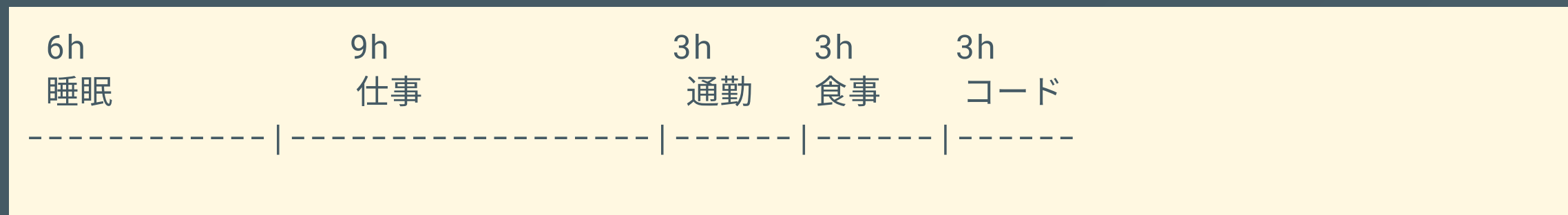
そんな暇はない！



これは心もとない(´・ω・`)

時短するんだッ！

すでにあるものをより短時間で終わらせましょう！



これが「質を高める」ことです！

質を高めるために

質を高めるには経験値が必要です。

1. 量（経験値）を準備する。
2. 「経験値」を用いて時間を「圧縮」する。

だから「量 > 質」

優先順位（やらないこと）を決める。
質を高めることで時間を作り出せる。

もくじ

- ~~時間はなぜ必要か~~
- ~~時間をどうやって使うか~~
- ~~時間をどうやって作るか~~
- まとめ

コード is 量！
時間が必要！

時間が多種多様な具体例を生む！
具体例が法則を見出せる（= 理解）！

時間は「分解する時間」「コードを書く時間」

やらないことを決めよう！
すでにやっていることの質を高めよう！



Enjoy!