

# テックブログ文豪への道

Road to tech-blog "Bunngou"

# ken7253

Frontend developer



技術記事を書いたりするのが趣味。

最近はNext.jsを使ったアプリケーションを書いています。

インターフェイス設計やアクセシビリティ・SSG関連の技術に興味があります。


 <https://github.com/ken7253>





 <https://zenn.dev/ken7253>


 <https://dairoku-studio.com>




技術記事書いてますか？

# 閲覧数少ないけど書いてます



[投稿する](#)




**ken7253**  
フロントエンドエンジニア  
439 Likes 25 Followings 24 Followers  


Edit profile

Articles 19   Scraps 28   Comments


TECH



Chrome120から実現する  
details要素のアクセシビリティ  
ターン

28日前   ♡ 2


TECH



なぜHTMLコメントにはイベント  
リスナーが付与できるのか

2ヶ月前   ♡ 7


TECH



UIのテストにアクセシビリティ情  
報が使われている理由

6ヶ月前   ♡ 4


TECH



私がthrowを使わない理由

6ヶ月前   ♡ 106


TECH



VSCodeで特定のnpm-scripts  
を非表示にする方法

2023/03/21   ♡ 3


TECH




Nuxt3でOGPを定義する

2023/01/11   ♡ 13


TECH



TECH



TECH



# なぜ記事を書くのか

最初はインプットとアウトプットのバランスを取るために始めた。

- 記事を書くことで技術の理解を深める
- 自分の考えを広めたい
- 文章を書くのが好き
- 他のアウトプットは敷居が高い

記事を書くことを通じて **インプットした知識を洗練させる** ことができる。

どのように記事を書いているか

## 題材集め

自分の知識は広く浅くなので、そこまで特定の分野について色々を書けるわけではない。

とにかく題材になりそうなものはメモしておいてあとから肉付けして記事にする。

- times
- X(twitter)
- Zennのスクラップ機能

など、後から見返すことのできる媒体に保存することが大切

**専門分野・得意分野がはっきりしていない人の戦略** として有効

いい題材・よく思われない題材



## いい題材

基本的に独自性があって自身の体験が含まれているものがベスト

**誰とも被らない内容** や **誰もまだ書いていない知識** は真っ先に書くべき

- チームで〇〇を導入してよかったこと・悪かったこと
- サービスを作ってみて〇〇の理解が進んだ話
- レビューで指摘された〇〇を深掘りしてみた

新規性がなくても自分たちで考えた結果や、努力をアピールする場所でいい。

きっかけは他人の記事や発言であっても **自分で調べて検証する** ことが大切。

いい題材

**解説記事** などは独自性が無くても大丈夫な部類ではある。

- 過去の自分がどこで躓いたのかを思い出しながら書く
- どの程度の知識を持っている人向けの解説なのかを決めておく

めちゃくちゃ基礎的な内容でも深掘りしてみると意外な知見がある。

- `flex` と `grid` の使い分け
- `table` 関連要素の使い方
- `${任意のライブラリ名}` とは何か

よく思われない題材

こういう内容の記事を書くな！という話ではない。

個人的にに **質の高い記事に仕上げるのが難しい** と思っている内容がある

- 参考になった記事〇〇選
- エンジニアが読んでおくべき本〇〇選

なぜか？

- どうしてもリンク集になりがち
- 自分の意見を入れるのが難しい
- リンク先の記事の内容が正しいのか精査するのがめんどい

よく思われない題材

- 特定のライブラリや技術に対してネガティブな内容を含むもの
- 技術に関する用語が微妙に間違っている

前者は、比較記事やライブラリの移行記事とかでも気を遣うほうがベター

後者は結構自分もやりがちだけど、ここにこだわると文章の印象が違う。

自分の中での表記ルールとかを持っておくと更にいい。

- React.jsではなくReact
- 要素とタグという用語の使い分け

最初は **指摘されたら修正すればいい** ぐらいの感覚でOK

## 記事を書く流れ

# 記事を書く流れ

自分の場合はこういう流れで書いてます。

**書き方に正解はない** のであくまで一例として。

1. 着地点を決める
2. 見出しを作る
3. とにかく書く
4. 文章を削る

## 着地点を決める

何も考えずに文を書き出すと **作文中に迷子** になる。

文章に矛盾や迷いが生じがちになるので、着地点を決めておく。

解説記事とかだと要約とかで締めることが多いかも。

## 見出しを作る

フロントエンドなら記事も正しくマークアップする。

文章を構造化することで **自分が今何を書いているのか** が分かりやすくなる。



## とにかく書く

見出しと着地点に沿ってとにかく文章を書いていく。

構造化は先に済ませているので **書きやすいセクションから** 書いていく。

このフェーズで見出しの構造を変更したり着地点を変えたりもする。

## 文章を削る

小学生の読書感想文ではないので文字数稼ぎは不要

勢いに任せて文章を書くと文章が長くなりがちなので文を絞る

- 無くても伝わる表現は削る
- 語順を並び替えて括弧を削る
- 意味が曖昧な表現があったら書き直す

**文章モリファクタリング** することで直すべき箇所が分かってくるようになる。

個人的に楽しい時間なので時間をかけがち、やり過ぎると永遠に記事が公開できない。



## おすすめ書籍

### 日本語の作文技術

- 句読点の打ち方
- 修飾語の使い方
- 語順の整理のしかた

などなど、相手に伝えるための文章の書き方が勉強できる。

## 最後に

アドカレや普段の記事を書きたい人向けの相談とかも受け付けています。

- 記事のレビュー
- 題材が決まらない人向けの相談
- 記事の構成についての相談
- もっと記事のクオリティを上げたい

などなど **Times**や**DMで連絡** いただければ相談時間確保します！