スピーディーな本作りと
カスタマイズ可能な本作りのための
Vivliostyle Themes プロジェクト

CSS組版 Vivliostyle ユーザーと開発者の集い 2020秋

2020.10.24　やましー

やましー @yamasy1549

- スパイスカレーづくりが趣味の学生🍛
- プログラミング言語かるたをつくった
- Vivliostyle を応援する者
  - Vivliostyle を使って卒論を書いた
  - Vivliostyle Themes に関わっている

Vivliostyle Themes とは

"**スピーディー**" な本づくりと "**カスタマイズ可能**" な本づくり

Themes がめざす 2 つのこと

① スピーディー

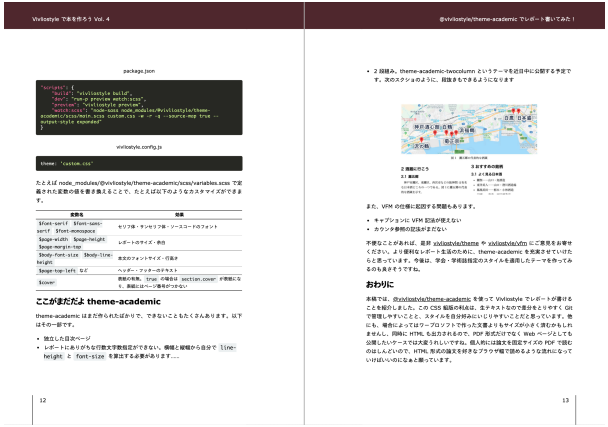
- 構造化された文章を書けば、スタイル（見た目）を意識せずにいい感じの本ができる
- さまざまなスタイルを適用できる→**ワンソース・マルチユース**

② カスタマイズ可能

- ユーザーが自由にスタイルを編集できる
- Themes を通してスタイルのカスタマイズ方法が統一されている

公式 Themes の紹介

- https://github.com/vivliostyle/themes
- 基本的なスタイルが定義された CSS ≒ **スタイルテンプレート**
- Themes を使って作られた本などの例 ↓



『Vivliostyle で本を作ろう Vol.4』

このスライド!

公式 Themes の紹介

- 現在は公式 Themes が 4 つ
  - 文庫 theme-bunko
  - スライド theme-slide
  - 技術書 theme-techbook
  - レポート theme-academic
- さらに増えます
  - 段組みレポート
  - 欧文文庫
  - 論文
  - etc...

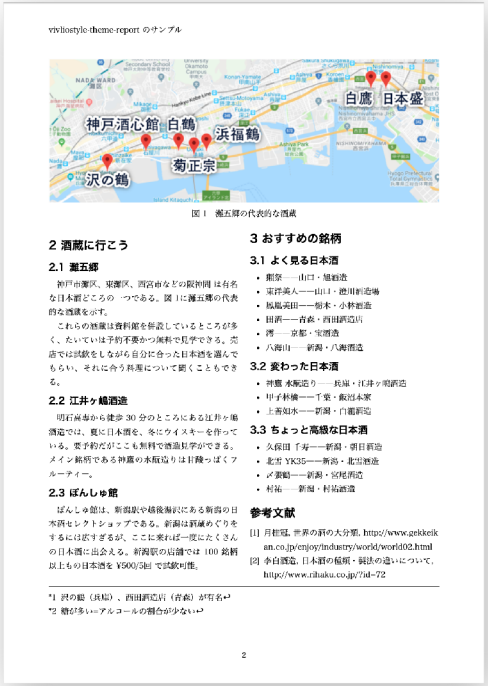

vivliostyle-theme-report のサンプル

図 1 灘五郷の代表的な酒蔵

2 酒蔵に行こう

2.1 灘五郷

神戸市灘区、東灘区、西宮市などの阪神間 は有名な日本酒どころの 一つである。図 1に灘五郷の代表的な酒蔵を示す。

これらの酒蔵は資料館を併設しているところが多く、たいていは予約不要かつ無料で見学できる。売店では試飲をしながら自分に合った日本酒を選んでもらい、それに合う料理について聞くこともできる。

2.2 江井ヶ嶋酒造

明石海峡から徒歩 30 分のところにある江井ヶ嶋酒造では、夏に日本酒を、冬にウイスキーを作っている。要予約だがここも無料で酒造見学ができる。メイン銘柄である神鷹の水廻造りは甘酸っぱくフルーティー。

2.3 ぼんしゅ館

ぼんしゅ館は、新潟県や越後海沢にある新潟の日本酒セレクトショップである。新潟は酒蔵めぐりをするには広すぎるが、ここに来れば一度にたくさんの日本酒に出会える。新潟駅の店舗では 100 銘柄以上もの日本酒を ¥500/5回 で試飲可能。

3 おすすめの銘柄

3.1 よく見る日本酒

- 鶴齢——山口・旭酒造
- 東洋美人——山口・澄川酒造場
- 福鳳来田——愛知・小野酒造
- 田酒——青森・西田酒造店
- 澤——京都・宝酒造
- 八海山——新潟・八海酒造

3.2 変わった日本酒

- 神鷹 水廻造り——兵庫・江井ヶ嶋酒造
- 甲子柿撰——千葉・飯沼本家
- 上善如水——新潟・白龍酒造

3.3 ちょっと高級な日本酒

- 久保田 千寿——新潟・朝日酒造
- 北雪 YK35——新潟・北雪酒造
- 〆張鶴——新潟・宮尾酒造
- 村祐——新潟・村祐酒造

参考文献

[1] 月桂冠. 世界の酒の大分類. http://www.gekkeikan.co.jp/enjoy/industry/world/world02.html
[2] 争白酒造. 日本酒の価格・製法の違いについて. http://www.rihaku.co.jp/?id=72

*1 沢の鶴（兵庫）、西田酒造店（青森）が有名↩
*2 糖が多い⇔アルコールの割合が少ない↩

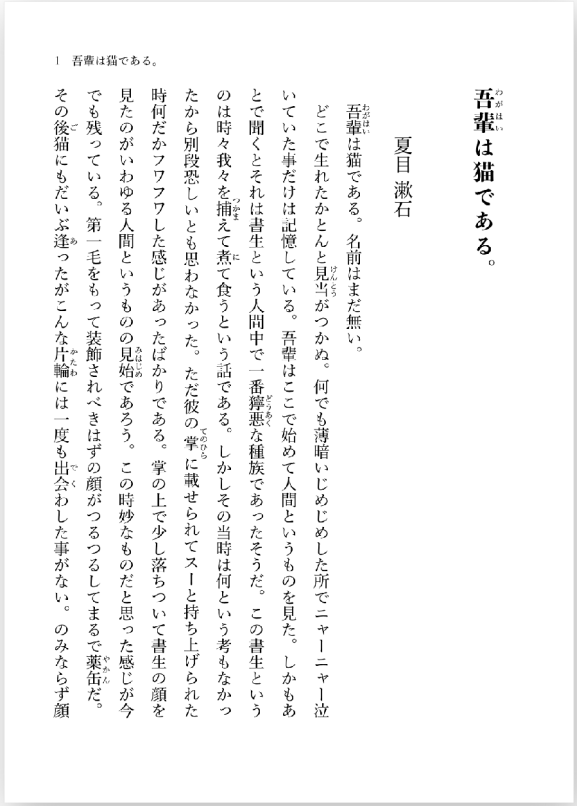
2

段組みレポートの例

文庫本スタイル `theme-bunko`

- @vivliostyle/theme-bunko
- スタンダードな文庫本
- 明朝体
- 縦書き
  - **ルビ**もOK
  - **縦中横**もばっちり

吾輩は猫である
ここを強調、
2020年、最高の夏



吾輩は猫である。

夏目 漱石

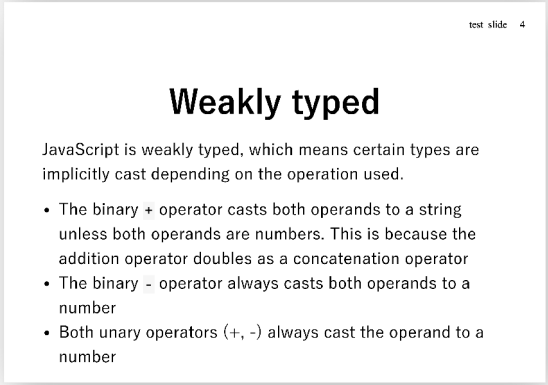
吾輩は猫である。名前はまだ無い。

どこで生れたかとんと見当がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。吾輩はここで始めて人間というものを見た。しかもあとで聞くとそれは書生という人間中で一番獰悪な種族であったそうだ。この書生というのは時々我々を捕えて煮て食うという話である。しかしその当時は何という考もなかったから別段恐しいとも思わなかった。ただ彼の掌に載せられてスーと持ち上げられた時何だかフワフワした感じがあったばかりである。掌の上で少し落ちついて書生の顔を見たのがいわゆる人間というものの見始であろう。この時妙なものだと思った感じが今でも残っている。第一毛をもって装飾されべきはずの顔がつるつるしてまるで薬缶だ。その後猫にもだいぶ逢ったがこんな片輪には一度も出会わした事がない。のみならず顔

1 吾輩は猫である。

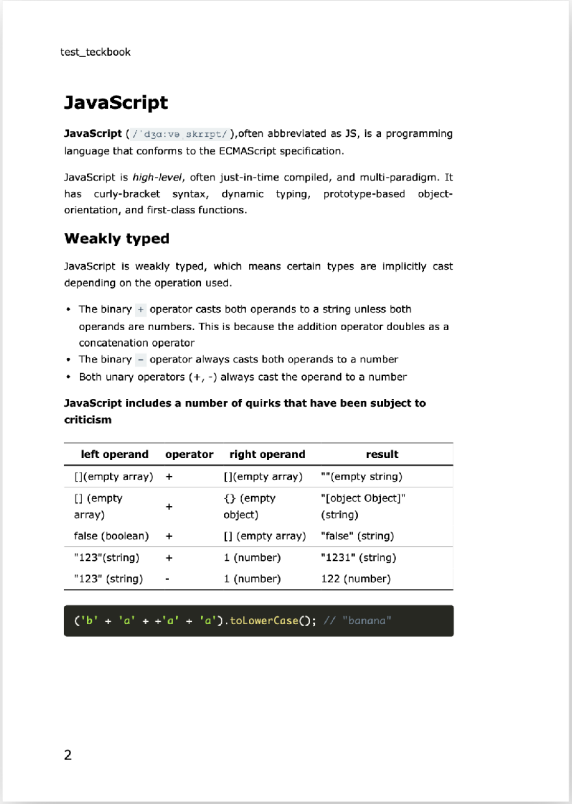
スライドスタイル `theme-slide`

- @vivliostyle/theme-slide
- この資料のようなスライド
- ゴシック体
- 横書き
- **表紙ページ**はスタイルが変わる
  - 青背景
  - 中央寄せ
  - ページ番号非表示



技術書スタイル `theme-teckbook`

- @vivliostyle/theme-techbook
- **印刷を意識**した技術書
  - 小口・ノドの余白調整
- ゴシック体
- 横書き
- **ソースコード**や**目次**もばっちり
- **脚注**が使える
- SCSS で書かれているのでカスタマイズしやすい



test_teckbook

JavaScript

JavaScript (/ˈdʒɑːvə skrɪpt/),often abbreviated as JS, is a programming language that conforms to the ECMAScript specification.

JavaScript is *high-level*, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Weakly typed

JavaScript is weakly typed, which means certain types are implicitly cast depending on the operation used.

- The binary `+` operator casts both operands to a string unless both operands are numbers. This is because the addition operator doubles as a concatenation operator
- The binary `-` operator always casts both operands to a number
- Both unary operators (+, -) always cast the operand to a number

JavaScript includes a number of quirks that have been subject to criticism

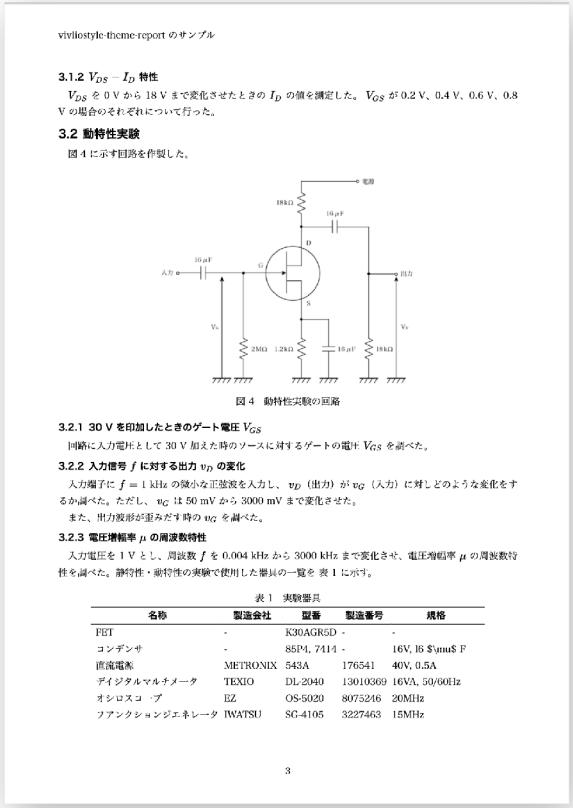
left operand	operator	right operand	result
[](empty array)	+	[](empty array)	""(empty string)
[] (empty array)	+	{} (empty object)	"[object Object]" (string)
false (boolean)	+	[] (empty array)	"false" (string)
"123"(string)	+	1 (number)	"1231" (string)
"123" (string)	-	1 (number)	122 (number)

```
('b' + 'a' + +'a' + 'a').toLowerCase(); // "banana"
```

2

レポートスタイル `theme-academic`

- @vivliostyle/theme-academic
- 学生が書くようなレポート
- 明朝体
- 横書き
- 自動で**章・節番号**がつく
- **数式**もばっちり
- SCSS で書かれているのでカスタマイズしやすい



vivliostyle-theme-report のサンプル

3.1.2 $V_{DS} - I_D$ 特性

V_{DS} を 0 V から 18 V まで変化させたときの I_D の値を測定した。V_{GS} が 0.2 V、0.4 V、0.6 V、0.8 V の場合のそれぞれについて行った。

3.2 動特性実験

図 4 に示す回路を作製した。

図 4　動特性実験の回路

3.2.1 30 V を印加したときのゲート電圧 V_{GS}

回路に入力電圧として 30 V 加えた時のソースに対するゲートの電圧 V_{GS} を調べた。

3.2.2 入力信号 f に対する出力 v_D の変化

入力端子に $f = 1$ kHz の微小な正弦波を入力し、v_D（出力）が v_G（入力）に対しどのような変化をするか調べた。ただし、v_G は 50 mV から 3000 mV まで変化させた。

また、出力波形が歪みだす時の v_G を調べた。

3.2.3 電圧増幅率 μ の周波数特性

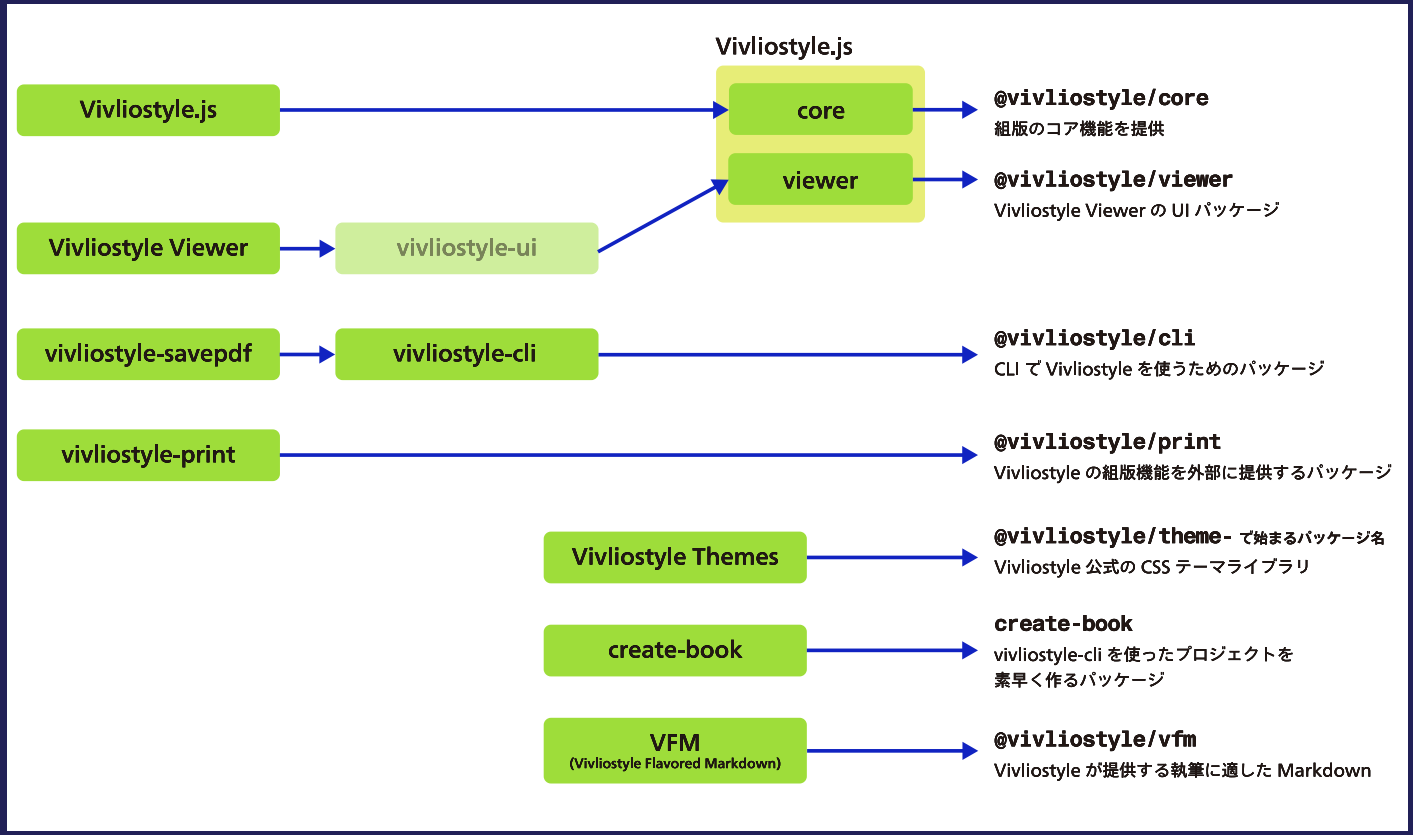
入力電圧を 1 V とし、周波数 f を 0.004 kHz から 3000 kHz まで変化させ、電圧増幅率 μ の周波数特性を調べた。静特性・動特性の実験で使用した器具の一覧を 表 1 に示す。

表 1　実験器具

名称	製造会社	型番	製造番号	規格
FET	-	K30AGR5D	-	-
コンデンサ	-	85P4, 7414	-	16V, 16 μ F
直流電源	METRONIX	543A	176541	40V, 0.5A
デイジタルマルチメータ	TEXIO	DL-2040	13010369	16VA, 50/60Hz
オシロスコープ	EZ	OS-5020	8075246	20MHz
ファンクションジエネレータ	IWATSU	SG-4105	3227463	15MHz

3

Themes を使って最速本づくり



Themes を使って最速本づくり

- Create Book を通して使う
  - Create Book で同人誌を作ろう!
  - チュートリアルガイド
  - FAQ: テーマをカスタマイズするには
- 本づくりの流れ
  1. **Themes からテーマを選択**
  2. VFM で原稿を執筆
  3. **カスタマイズ用スタイルを定義**
  4. `$ vivliostyle build`
  5. PDF と HTML ができる!

```
./my_book/
├── manuscripts/ (原稿)
│   ├── cover.md
│   ├── chapter01.md
│   └── chapter02.md
├── my_theme/ (カスタマイズ)
│   ├── package.json
│   └── theme.css
├── node_modules/
├── package.json
├── vivliostyle.config.js
└── .vivliostyle
```

これからの Themes

"もっとスピーディー" に、"もっとカスタマイズ可能" に

もっとスピーディー

- Themes の種類を増やす
  - [themes#17] 学術用のマルチカラム可能なテーマ theme-academic-twocolumn を作る
  - [themes#16] 欧文用テーマ theme-gutenberg を作る
  - 論文!?　請求書!?　新聞!?
- issueで提案できます😃
  - ニッチなものでも大丈夫!
  - 既存 Themes の問題点・改善案もお待ちしています
- pull req 大歓迎💻
  - **新しい用途の Themes は公式 Themes として採用**したい

もっとカスタマイズ可能

- 原稿ファイルごとに異なる CSS を指定できるように
  - [themes#9] How to customize the style of themes?
- よく使う設定項目は CSS を書かなくても設定できるように
  - [themes#19] テーマに共通して、ユーザ指定可能なパラメータを洗い出す
  - [vivliostyle-cli#76] CLIにSCSSのトランスパイル機能を含める
  - ページサイズ・フォントサイズ・行間など
  - 将来的には Vivliostyle Pub で、CSS の知識がなくても GUI で設定できるようにしたい
- issueで提案できます😃

Vivliostyle Themes まとめ

- "**スピーディー**" な本づくりと "**カスタマイズ可能**" な本づくり
  - Create Book を通して Themes（テンプレート）を使う
  - CSS を書いて Themes を上書きできる
- これからやっていくこと
  - まずは Themes の種類を増やす
  - CSS を書かなくてもカスタマイズできる仕組みを整える
- issueで提案できます😀
- 🖋 Happy writing!