

「とほほのWWW入門」を 29年間継続してきた 秘訣とは

2025年2月5日
杜甫々（とほほ）

自己紹介

- 「とほほのWWW入門」管理人
 - 1996年9月からなので、もう**29年目**
- ハンドル：**杜甫々**（とほほ）
 - 本名は違います
- 広島生まれ。広島在住。カープファン
 - 昨年の観戦成績は4勝5敗1分
- インターネット歴
 - 1988年からなので**37年**
- インターネット老人会会長をめざしています(ウツ)



「とほほのWWW入門」

とほほのWWW入門 検索 (By Google)

とほほの娘がLINEスタンプはじめました



メニュー

- HOME
- フレーム版
- ダウンロード
- ラウンジ
- URLの広場

はじめに

- ご使用上の注意 (1)
- 管理者へのメール (1)
- 主な更新履歴 (24)
- 自己紹介 (1)

基本編

- 用語集 (82)
- 逆引きリファレンス (32)
- Webページ作成入門 (7)

フォーマット

- HTML (400)
- HTML5 (7)
- XHTML (1)
- MathML (1)
- DTD (1)
- JSON (1)
- SVG (1)
- VML (1)
- GIF (1)
- CSV (1)
- セマンティック・ウェブ (1)

CSS

- CSS (ABC順)(719)
- Bootstrap (61)
- Less (1)
- Sass (1)
- Tailwind CSS (1)
- CSSフレームワーク (1)
- リセットCSS (1)

プログラミング言語

- JavaScript (39)
- TypeScript (1)
- Java (25)
- Perl (4)
- PHP (14)
- Ruby (11)
- Python (13)
- パッチ (1)
- PowerShell (1)

<https://www.tohoho-web.com/>

- 1996年開設
- HTML/JavaScript/CSSなどWeb関連技術紹介サイト

Copyright (C) 1996-2025 杜甫々 
<https://www.tohoho-web.com/www.htm>



「とほほの〇〇入門」

とほほのWWW入門 検索 (By Google)

とほほの娘がLINEスタンプはじめました



メニュー

- HOME
- フレーム版
- ダウンロード
- ラウンジ
- URLの広場

■ はじめに

- ご使用上の注意 (1)
- 主な更新履歴 (24)
- 管理者へのメール (1)
- 自己紹介 (1)

■ 基本編

- 用語集 (82)
- Webページ作成入門 (7)
- 逆引きリファレンス (32)

■ フォーマット

- HTML (400)
- HTML5 (7)
- XHTML (1)
- MathML (1)
- DTD (1)
- JSON (1)
- SVG (1)
- VML (1)
- GIF (1)
- CSV (1)
- XMLテクニック・ウェブ (1)

■ CSS

- CSS (ABC順) (719)
- Bootstrap (61)
- Less (1)
- Sass (1)
- Tailwind CSS (1)
- CSSフレームワーク (1)
- リセットCSS (1)

■ プログラミング言語

- JavaScript (39)
- TypeScript (1)
- Java (25)
- Perl (4)
- PHP (14)
- Ruby (11)
- Python (13)
- Go (1)
- C言語 (1)
- C# (1)
- Rust (1)
- Scala (1)
- Haskell (1)
- Kotlin (1)
- Bash (1)
- AWK (1)
- バッチ (1)
- PowerShell (1)
- FORTRAN (1)
- Lisp (1)
- 8ビットアセンブラ (1)

Web関連技術を中心に紹介しています

■ フレームワーク

- Ruby on Rails (1)
- Drupal (18)
- Django (3)
- Laravel (1)
- jQuery (16)
- Node.js (1)
- Express (1)
- AngularJS (1)
- Angular (1)
- Vue.js (1)
- React (1)
- Next.js (1)
- Flask (1)
- AMP (1)

■ ライブラリ

- Chart.js (2)
- Jspreadsheet (1)
- gRPC (1)
- MathJax (1)

■ CGI

- CGI (7)
- カウンター設置 (1)
- メール送信フォーム (1)
- 掲示板設置 (1)
- 検索フォーム (1)
- とほほのCGIソフト集 (6)

■ Linux

- UNIX/Linux (1)
- CentOS (1)
- CentOS 8 (1)
- RHEL9 (1)
- AlmaLinux (1)
- Yum (1)
- Vim (1)
- SELinux (1)

■ データベース

- MySQL (3)
- PostgreSQL (1)
- MongoDB (1)
- Cassandra (1)
- OpenLDAP (1)

■ ツール

- VirtualBox (1)
- Vagrant (1)
- WSL (1)
- Docker (18)
- Kubernetes (1)
- OpenSSL (1)
- Pacemaker (1)
- Git (1)
- GitHub (1)
- GitLab (1)
- Zabbix (1)
- SNMP (1)
- ELK (1)
- Prometheus (1)
- tmux (1)
- tcpdump (1)
- jq (1)

■ プロトコル

- HTTP (1)
- Cookie (1)

■ 技術知識

- 文字コード (1)
- Unicode一覧 (1)
- フォント (1)
- 正規表現 (1)
- 色入門・色覚本 (1)
- 拡張子 (1)
- 画像(GIF/JPEG) (1)
- 暗号化 (1)
- 単位 (1)
- 数値フォーマット (1)
- マルチメディア (4)
- ライセンス (1)
- クリエイティブ・コモンズ (1)

■ セキュリティ

- セキュリティ (1)
- CORS (1)
- 同一生成元ポリシー (1)

■ 注意事項

- HP作成上の注意 (1)
- 著作権 (1)
- アクセス向上施策 (1)

■ アラカルト

- アラカルト (42)
- How To (39)
- トラブルシューティング (2)
- ショートカット (1)
- 競演ツール集 (10)
- 競演リンク集 (1)
- 社畜の本 (1)
- スペシャルサンクス (1)

■ 番外編

- 点字 (1)
- 映画(2021年) (1)
- 珍しい苗字 (1)
- 陶磁器 (1)
- 仏教 (1)
- 韓国語 (1)
- 中国語 (1)
- 洋楽 (1)
- 確定申告 (1)
- 相続 (1)
- 資産運用 (1)
- 法会 (1)
- 広島ラーメン (1)
- 広島お好み焼き「八重」 (1)
- タイ料理 (1)
- お酒 (1)
- お茶・紅茶 (1)
- 三大〇〇 (1)
- とほほの雑記帳 (1)

Copyright (C) 1996-2024 社博々
https://www.tohoho-web.com/www.htm



「とほほの○○入門」

• 下記などの入門記事を紹介

プログラミング言語系

HTML入門 (1996年)
JavaScript入門 (1996年)
CSS入門 (1997年)
Perl入門 (2007年)
Java入門 (2004年)
PHP入門 (2013年)
Ruby入門 (2014年)
Python入門 (2014年)
Go言語入門 (2020年)
C言語入門 (2020年)
C#入門 (2020年)
Rust入門 (2020年)
Bash入門 (2020年)
Haskell入門 (2020年)
FORTRAN入門 (2020年)
Scala入門 (2021年)
AWK入門 (2021年)
Kotlin入門 (2021年)
Lisp入門 (2022年)
8bitアセンブラ入門 (2022年)

フレームワーク系

jQuery入門 (2013年)
Bootstrap入門 (2015年)
AngularJS入門 (2015年)
Node.js入門 (2016年)
Django入門 (2018年)
Angular入門 (2018年)
Vue.js入門 (2018年)
React入門 (2018年)
Laravel入門 (2020年)
Ruby on Rails入門 (2022年)
Next.js (2024年)

その他系

文字コード入門 (1996年)
色入門 (1997年)
Cookie入門 (1997年)
著作権入門 (1998年)
HTTP入門 (2005年)
フォント入門 (2012年)
Docker入門 (2016年)
セキュリティ入門 (2018年)
暗号化入門 (2021年)
正規表現入門 (2021年)
DevTools (2024年)

どれがどの言語だったか覚えていられなくて、
コーディングする時は for文ひとつ書くにも
自分のリファレンスをにらめっこしてます...

本書の内容は...

- 下記で発表した内容も多く含んでいますので、すでに聞かれたことのある方はすみません
 - オープンセミナー2022@岡山 ... (2022年8月20日)
 - YAPC::Hiroshima 2024 ... (2024年2月10日)
 - ファインディ様カンファレンス ... (2024年4月16日)
 - 情報処理学会 ... (2024年8月10日)
- 質問の時間もとりたいと思います。

個人的歷史

生まれ（1965年）

- 1965年生まれ（59歳）
- 広島県呉市出身



戦艦大和



呉冷麺



コンピューターとの出会い（1981年）

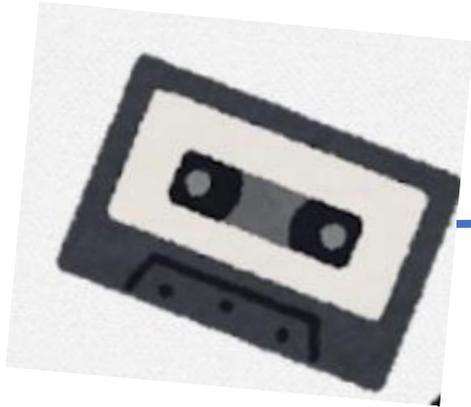
- コモドール社 VIC-1001
 - ブラウン管テレビに接続して使用
 - CPU：初代ファミコンと同じ 6502
 - **メモリ：5KB**
- 兄弟で金を出し合い、3KB の拡張メモリパックを追加



VIC-1001 By Cbmeeks is licensed under CC BY-SA 3.0
https://commons.wikimedia.org/wiki/File:Commodore_VIC_20_computer1.jpg

コンピューターとの出会い（1981年）

- 外部媒体は**カセットテープ**
 - ゲームのローディングに15分くらい
 - ローディング中にエラーになることも



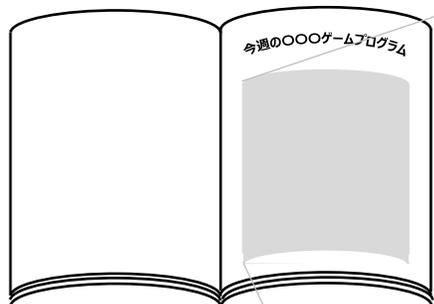
ぴー-----っ、
があ-----っ！

音を聞くとどの
プログラムかわかる



マイコン雑誌にはゲームプログラムが印字されていました

雑誌にまだ CD-ROM なんて添付できない時代。



```
0000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 : 0a
0000020 03 00 3e 00 01 00 00 00 10 6b 00 00 00 00 00 : 0d
0000040 40 00 00 00 00 00 00 00 58 1e 02 00 00 00 00 : 08
0000060 00 00 00 00 40 00 38 00 0d 00 40 00 1e 00 1d 00 : 00
0000100 06 00 00 00 04 00 00 00 40 00 00 00 00 00 00 : 0a
0000120 40 00 00 00 00 00 00 00 40 00 00 00 00 00 00 : 00
0000140 d8 02 00 00 00 00 00 00 d8 02 00 00 00 00 00 : 04
0000160 08 00 00 00 00 00 00 00 03 00 00 00 04 00 00 00 : 0f
0000200 18 03 00 00 00 00 00 00 18 03 00 00 00 00 00 00 : 06
0000220 18 03 00 00 00 00 00 00 1c 00 00 00 00 00 00 00 : 07
0000240 1c 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 : 0d
0000260 01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 : 05
0000300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : 00
0000320 10 35 00 00 00 00 00 00 10 35 00 00 00 00 00 00 : 0a
0000340 00 10 00 00 00 00 00 00 01 00 00 00 05 00 00 00 : 06
0000360 00 40 00 00 00 00 00 00 00 40 00 00 00 00 00 00 : 00
0000400 00 40 00 00 00 00 00 00 81 32 01 00 00 00 00 00 : 04
0000420 81 32 01 00 00 00 00 00 10 00 00 00 00 00 00 00 : 04
0000440 01 00 00 00 04 00 00 00 00 80 01 00 00 00 00 00 : 06
0000460 00 80 01 00 00 00 00 00 00 80 01 00 00 00 00 00 : 02
0000500 80 74 00 00 00 00 00 00 80 74 00 00 00 00 00 00 : 08
0000520 00 10 00 00 00 00 00 00 01 00 00 00 06 00 00 00 : 07
0000540 70 ff 01 00 00 00 00 00 70 0f 02 00 00 00 00 00 : 01
```

印刷された 16進ダンプをひたすら打ち込む



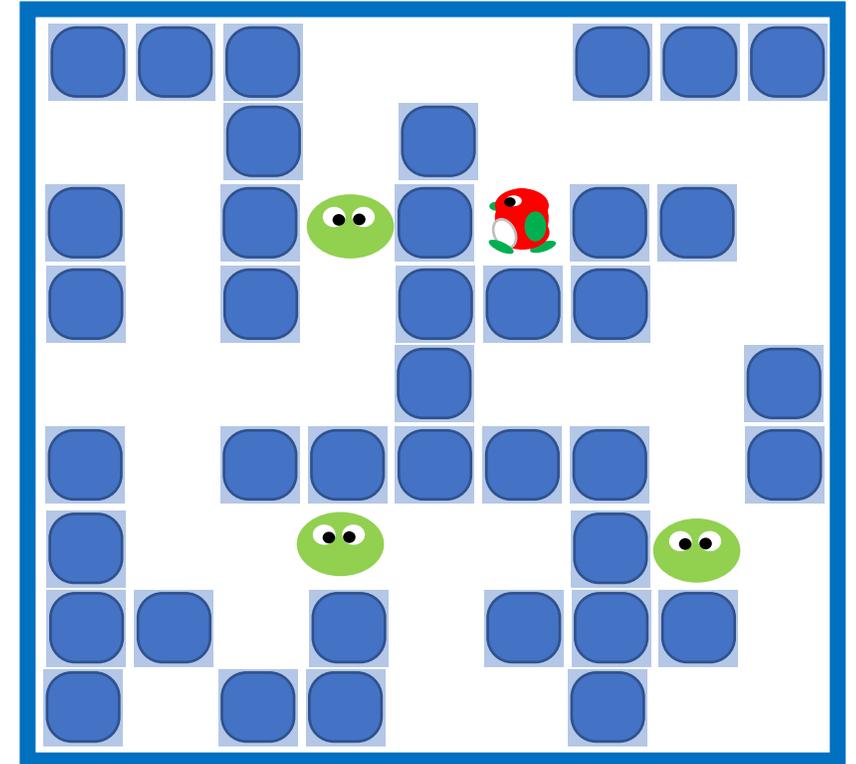
VIC-1001のプログラミング（1982年頃）

- BASIC

- 変数は極力1文字

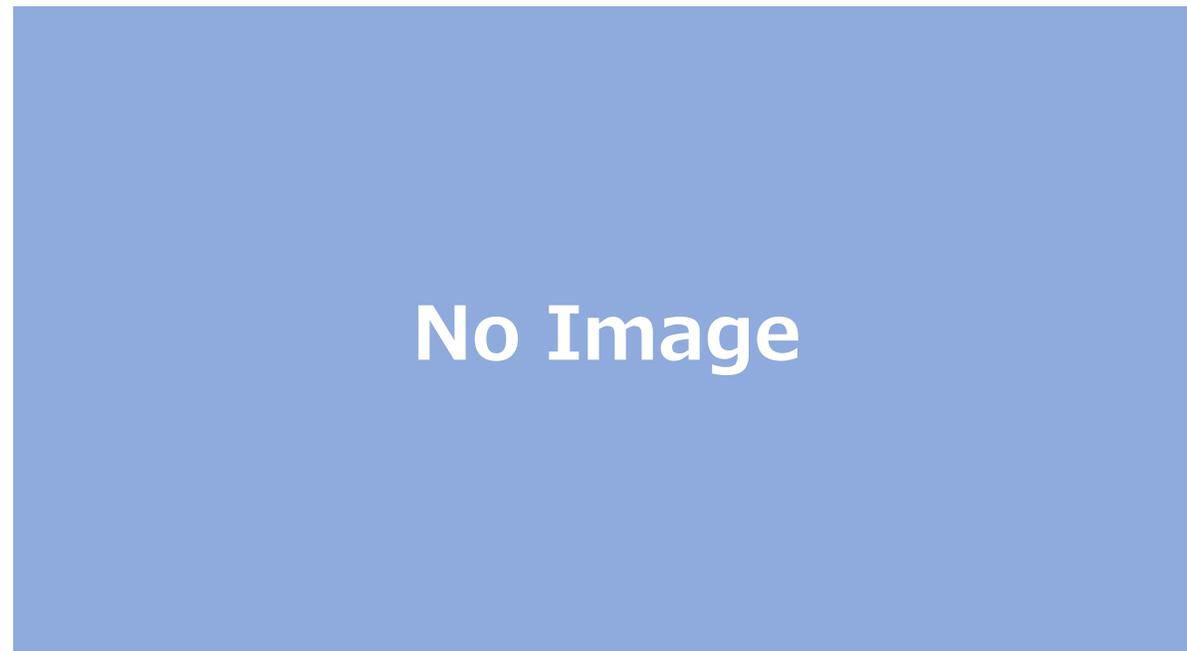
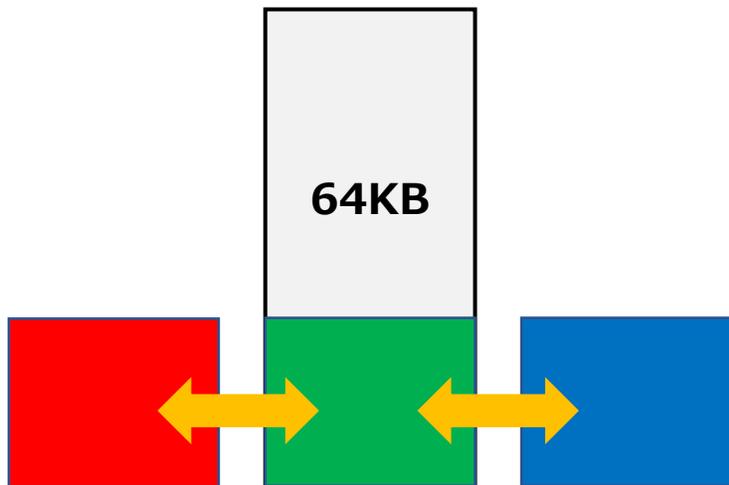
- 「ペンゴ」もどきゲーム作ったりして遊んでました

- 遊んでいると、スタック領域が増えていき、**データ領域を破壊**
 - キャラクターデータが破壊されて**世界の崩壊**が始まりました



PC-8801 mkII SR (1985年) 名機

- NECの PC88 シリーズ
- Z80A (ゼッパチ) 互換CPU
 - 8080系8ビットCPU
- メモリは**64KB** 10倍以上!
 - VIC-1001から10倍以上の増加
 - カラー画面はバンク切り換えで対応



PC-8801 mkII SR

PC-8801 mkII SR (1985年) **名機**

- 2024年8月8日、電波新聞社から復刻ミニサイズ版発売が発表されました
- 2025年春発売予定。予定価格：3万3,000円



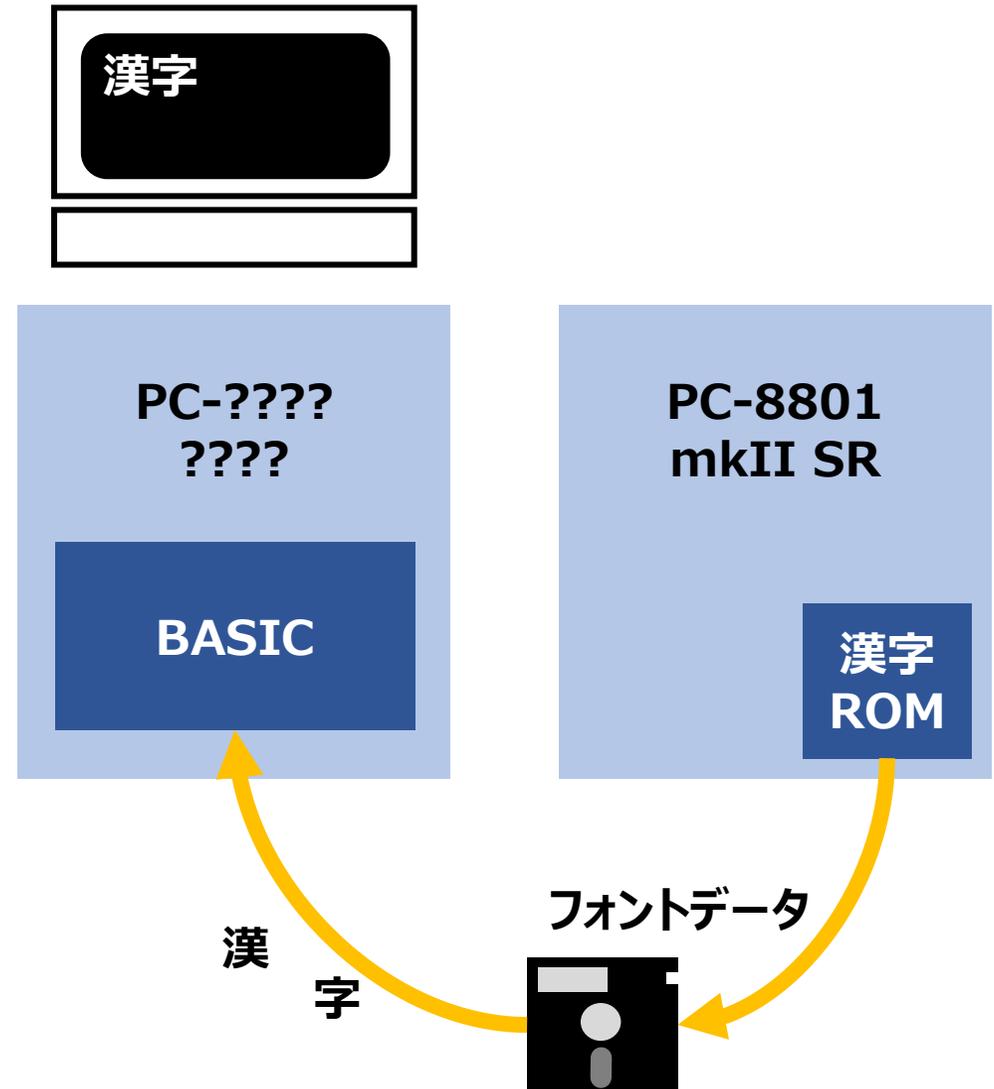
引用元：4Gamer.net「一世を風靡した8ビットホビーマシンが小さくなって帰ってくる。
「PasocomMini PC-8801 mkII SR」発表。詳細については8月8日に明らかに」
<https://www.4gamer.net/games/800/G080048/20240518005/>



引用元：4Gamer.net「往年の名機の復刻版小型ゲーム機
「PasocomMini PC-8801 mkII SR」は2025年春に発売！」
<https://www.4gamer.net/games/800/G080048/20240808033/>

友人につくってあげた漢字BASIC（1985年）

- 漢字ROMを持たないパソコンの友人に漢字情報をフロッピーディスクから読みこみながら表示できるように、BASICを拡張
 - 今思えば、著作権的にまずかったかも...
- 1文字画面に表示する度にフロッピーディスクをランダムアクセスするので、ちょっと長文を表示するとディスクが壊れそう...



PC-9801 VM2 (1986年)

名機

- PC-9801 VM
- CPU : 8086互換 (V30)
- メモリ : 384KB 6倍!
384MB じゃないよ
- 追加でハードディスクも購入
 - HDD : 11MB



htomari, CC BY-SA 2.0
<https://creativecommons.org/licenses/by-sa/2.0>
via Wikimedia Commons

プログラミングとの出会い

- **VIC-1001 BASIC (1981年)**
- **PC-8801 BASIC (1984年)**
- **Z80 (ゼッパチ) アセンブラ**
- **FORTRAN (1986年)**

プログラミングのアルバイト（1986年）

- 行きつけのパソコン関連ショップで、いきなりの**プログラミングバイト**開始
 - 社長の知り合いの個人会社の化粧品管理システム。
 - BASIC のはずが、いつの間にか C言語で開発することに・・・
- COBOLの**キーパンチャーのアルバイト**
 - PCが1人1台の時代ではないので、プログラムは紙と鉛筆で作成
 - 作成されたプログラム（紙）を、ただひたすらPCに打ち込む仕事
 - 1行2円だったかな？ 間違えると10円の罰金

ゲームをつくらうぜ！（1986年）

- ・バイト先の社長が宴会中にいきなりゲームをつくるぞ宣言。
- ・アイデアを募集してたので、考えてみました。

桃栗3年柿8年ゲーム



桃・栗・柿の木を育てるゲーム
1ゲームに8年かかってしまうので没
世界初の育成ゲームになってたかも？

たまごっち：1996年

ライトペンを使ったダーツゲーム



色々なものが
壊れそうなので没



Gregory Lloyd, CC BY 2.0
via Wikimedia Commons



リアルタイム将棋



フリー素材.com

早く駒を動かした方が勝ち。
没にはならなかったけど、
社長のゲーム作るぞ熱が
さめちゃってました...

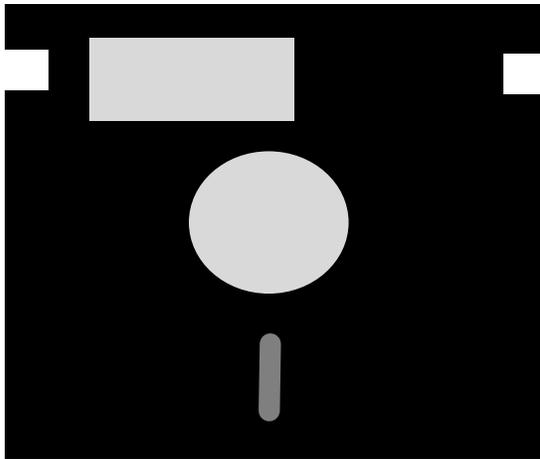
大学研究室ではこんなのを使っていました（1987年）

• Apple-II

貧乏研究室だったので…

- 1977年発売
- CPU はファミコンと同じ 6502
- フロッピーは 1DD
 - フロッピーディスクの片面しか使えないので、表裏ひっくり返して使っていました

こっちにも切り込みを
入れると両面使える

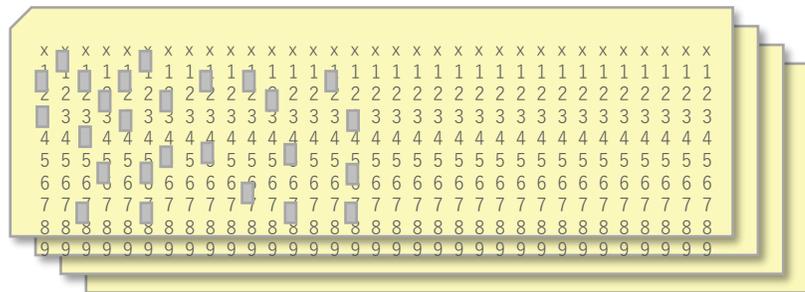


User Maury Markowitz on en.wikipedia,
Copyrighted free use, via Wikimedia Commons

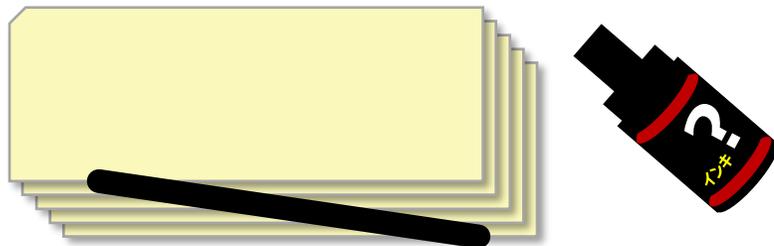
大学研究室ではこんなのを使っていました（1987年）

• パンチカード

- 磁気ディスクを信用しない教授の授業はフロッピーディスク使用禁止



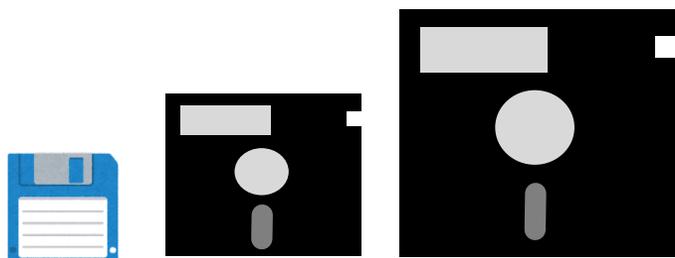
- バラバラになっても元に戻せるように斜め線をひいておく



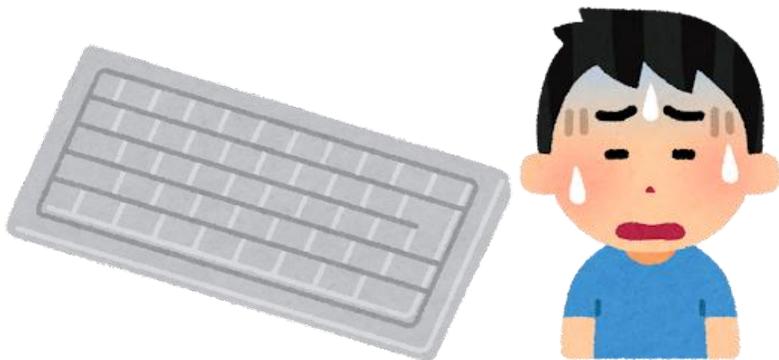
User Maury Markowitz on en.wikipedia,
Copyrighted free use, via Wikimedia Commons

そのころ：とほほママもショールームでこんな体験してました

8インチフロッピーディスクを
折りたたんで送る新人さん



キーボードの ABC... の順番が
バラバラだと電話してくるお客さん



「環境設定の問題かも...」という
「隣の部屋に移動したけど直らない」と...



「マウスを上に動かして」というと
マウスを上に持ち上げるシニアさん



入社

入社：1988年

- ちょっと研究所っぽい部署に配属



Sun Workstation

Fourdee at English Wikipedia,
Public domain,
via Wikimedia Commons



MicroVAX

Emiliano Russo, Associazione Culturale VerdeBinario,
Public domain, via Wikimedia Commons

インターネットとの出会い（1988年）

- ・インターネット（TCP/IP）グループに配属される...

今度、社内講習があるから
大丈夫だよ

という訳で、その講師よろしくね

講師やるとなったら、全力で覚えるっしょ
よろしくー

インターネット（TCP/IP）なんて、
全然わからないんですが...

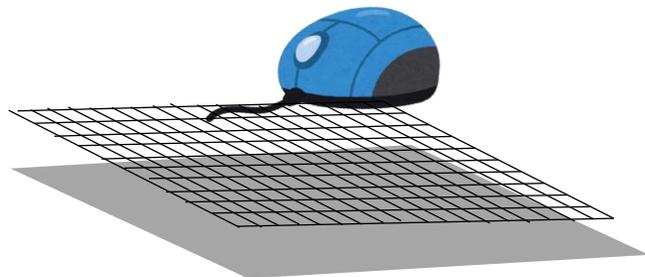
ほっ

えw？

えw？ えw？

入社：職場には色々な人が居て**刺激満載**

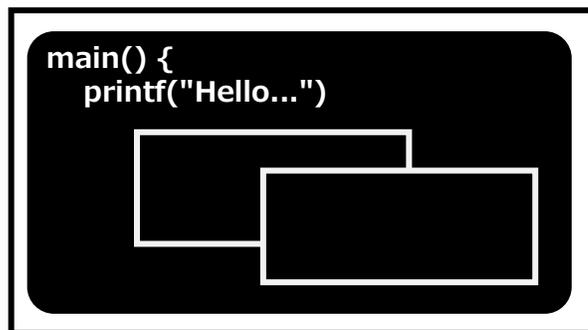
光学式マウスのマウスパッドを
OHPシートで自作を試みる先輩



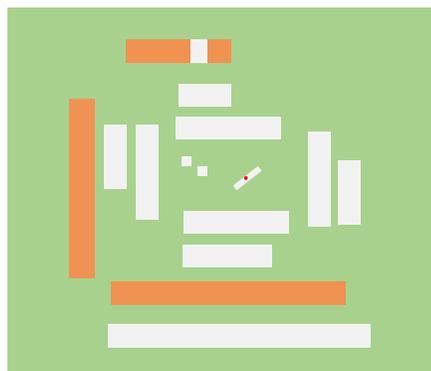
大学時代からすでに
プログラミング本を出版している同輩



テキストエディタ (Emacs) 上に
罫線お絵かきソフトを試作する先輩



他UNIX上のマージャンソフトを
自社UNIX上でコンパイルする先輩



1989年に
日本とアメリカ出張者間で
マージャン

その頃のインターネット・・・

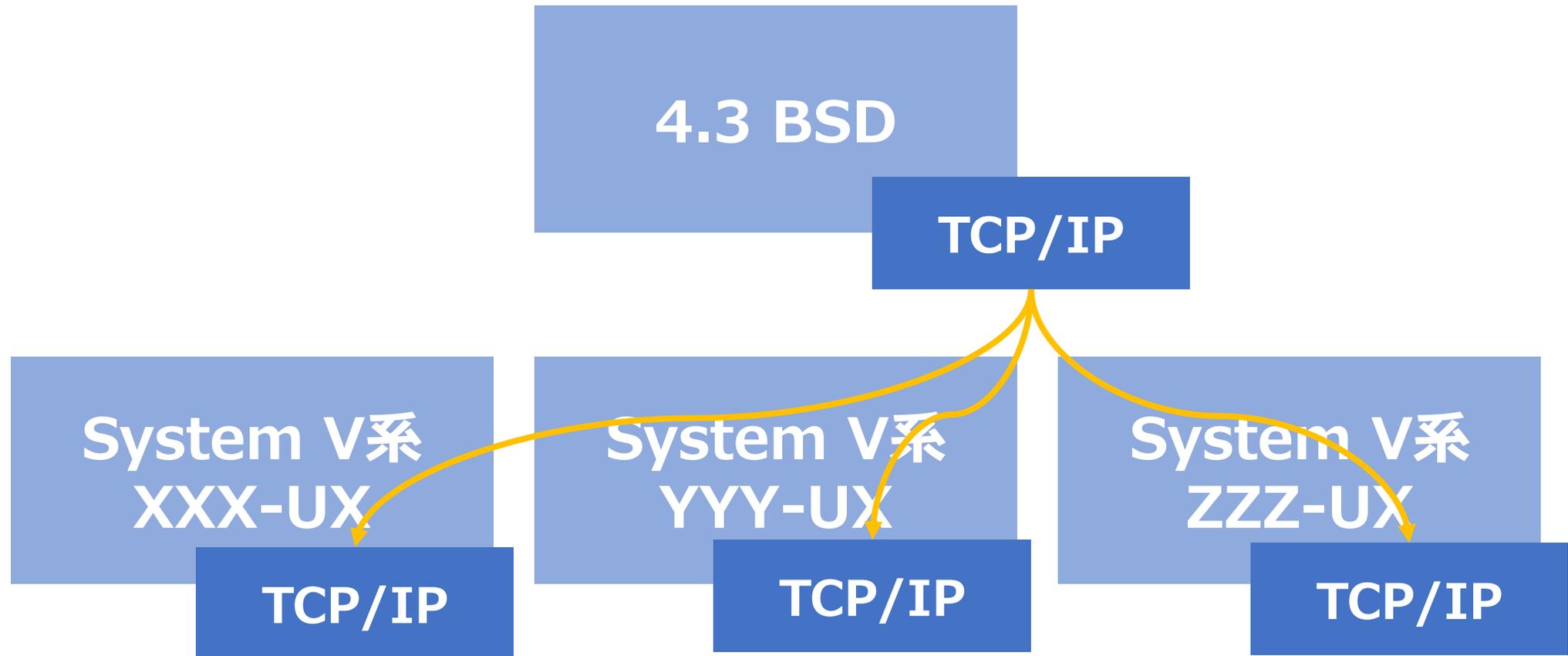
• JUNET (ジュネット)

- 慶応義塾大学の 村井純 先生がはじめられた日本最初のインターネット
- メールとネットニュース (fj.jokes等) の配信
- 学術目的のみで使用可能。商用利用は不可
- 商用メールは JUNET を経由しないよう、経路を指定して記述したりしてました

`keiro1!keiro2.example.com!yamada@uunet.uu.net`

TCP/IP 移植プロジェクト (1988年)

- ・ インターネットをサポートしていない UNIX 系システムに **TCP/IPモジュール**を移植

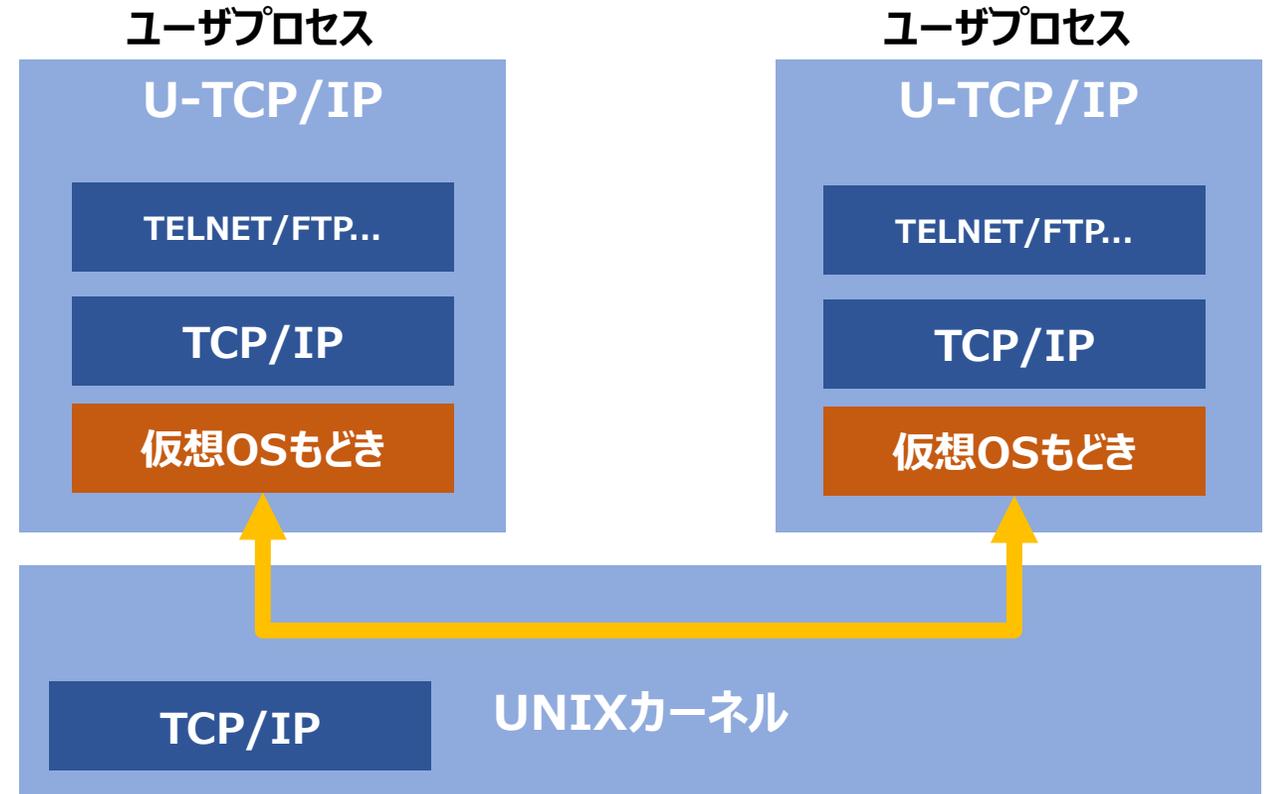


仕事：ユーザプロセスTCP/IP（1989年頃）

カーネルの再コンパイル： 2時間
ログファイル出力： 使えない
デバッガ： 使えない

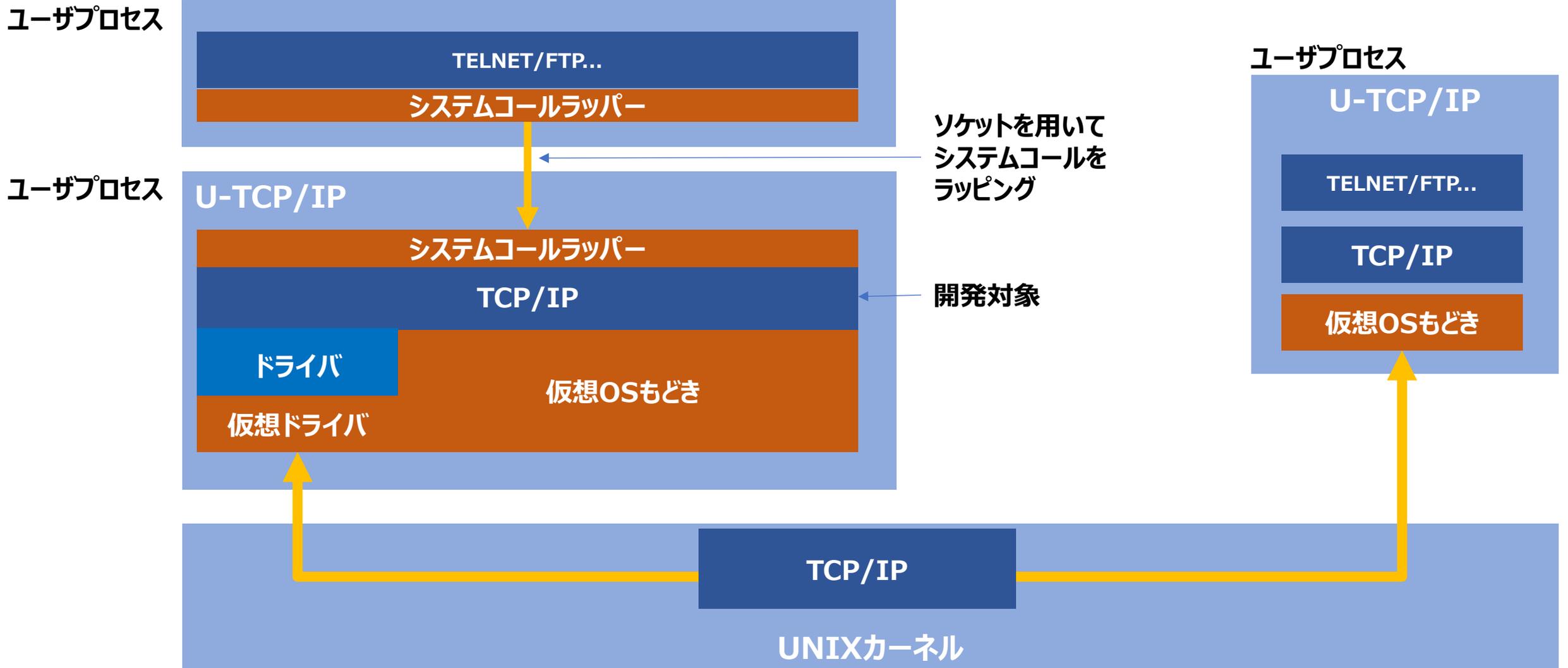
カーネル内モジュールをユーザ
プロセス空間で動かすための
簡単な **仮想OSもどき** を開発

再コンパイル： 2時間 → 数分
ログファイル出力： 不可 → 可能
デバッガ： 不可 → 可能



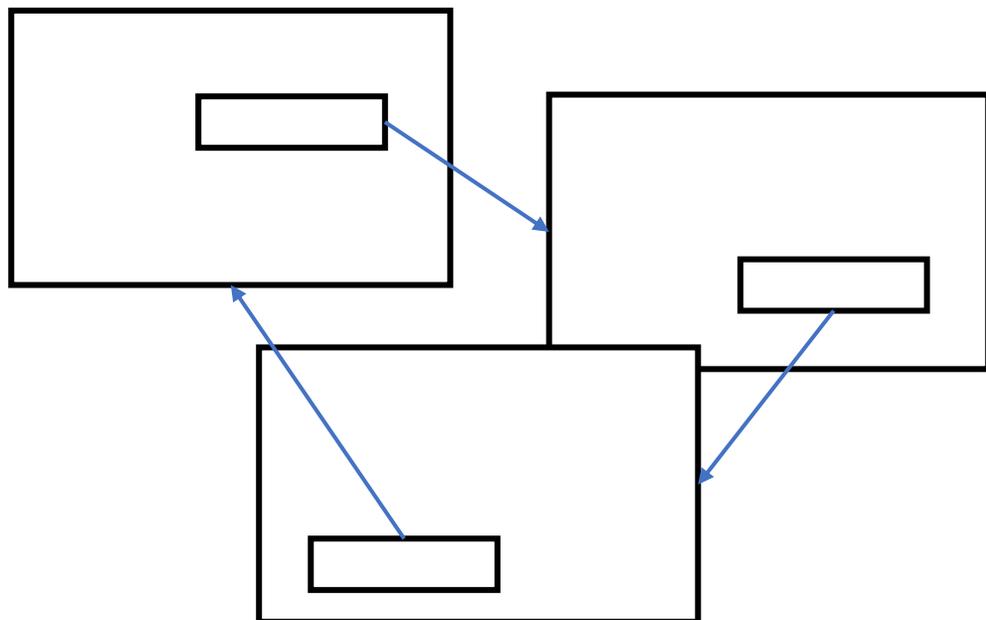
仕事が暇なときに遊びで作ってました...

仕事：ユーザプロセスTCP/IP（1989年頃）



Macintosh の HyperCard 好きでした（1988年）

- 1987年に Apple が開発
- カードとカードがメッシュにリンク
- **ハイパーテキスト** を最初に実現した最初の商用ソフトウェア

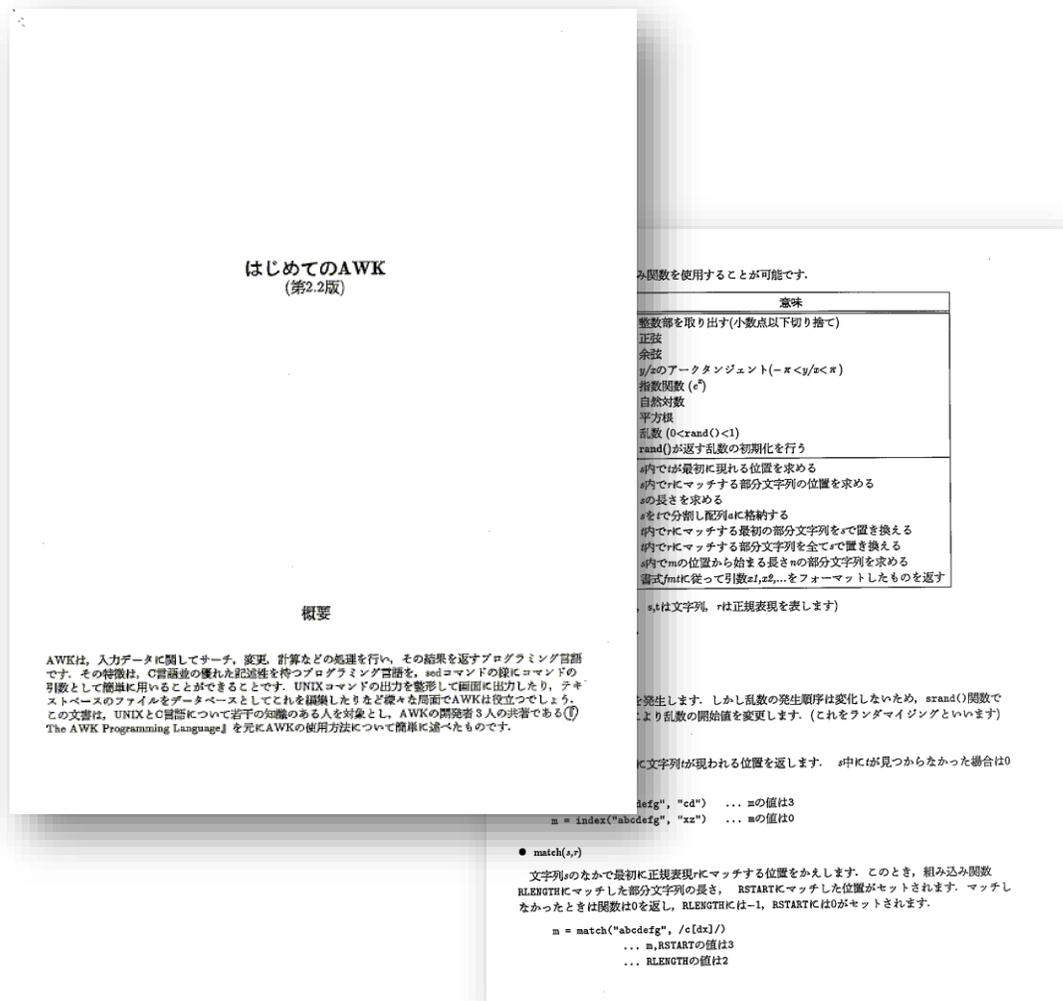


Sailko, CC BY 3.0
<<https://creativecommons.org/licenses/by/3.0/>>, via Wikimedia Commons

「とほほのWWW入門」の原点？（1989年）

- 会社で AWK の勉強会を実施
- 「**はじめてのAWK**」を執筆
- 今読み返してみると、今現在の自分の書き方スタイルとまったく同じ...
- これが原点だったのかもしれない

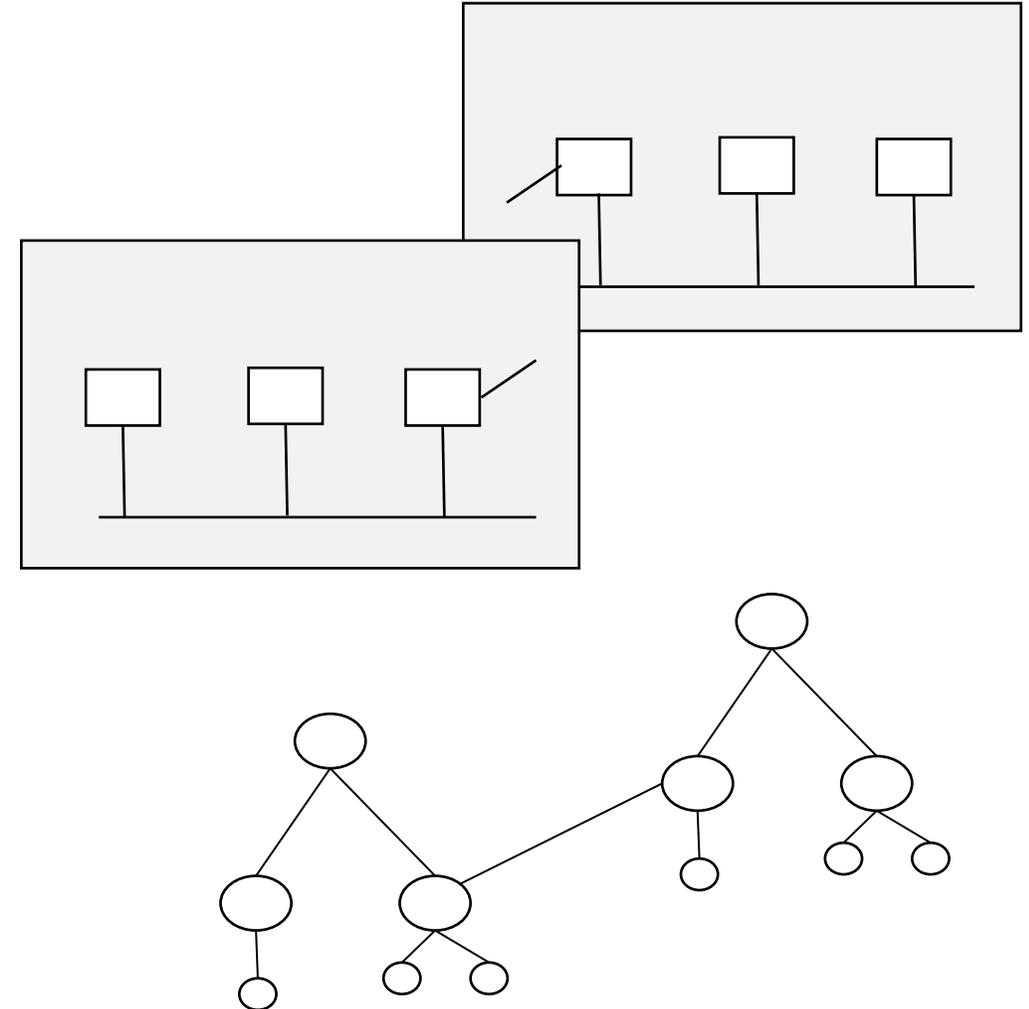
今でも AWK は使ってます
テキストの整形に無茶苦茶便利
是非使ってみて！



仕事：オブジェクトフレームワーク（1991年頃）

・ネットワーク管理システムの開発

- すべてのデータ（マップ、アイコン、管理対象機器、インタフェース、LAN...）をオブジェクトとして管理
- すべての操作を、**オブジェクトと関係操作、プロパティ操作**として実行
 - ・ マップオブジェクトとアイコンオブジェクトと管理対象オブジェクトを紐づけると、画面にアイコンが表示される。
 - ・ ロギングオブジェクトにパラメータを設定するとロギングオブジェクトを監視しているデーモンに通知されてロギングが始まる。
- オブジェクトに**クラスを動的に割り当てる**ことで、クラスメソッド(FTPログインなど)を呼び出せる。



まだ、オブジェクト指向という言葉も知らず、コンポーネントとかアビリティとか呼んでました

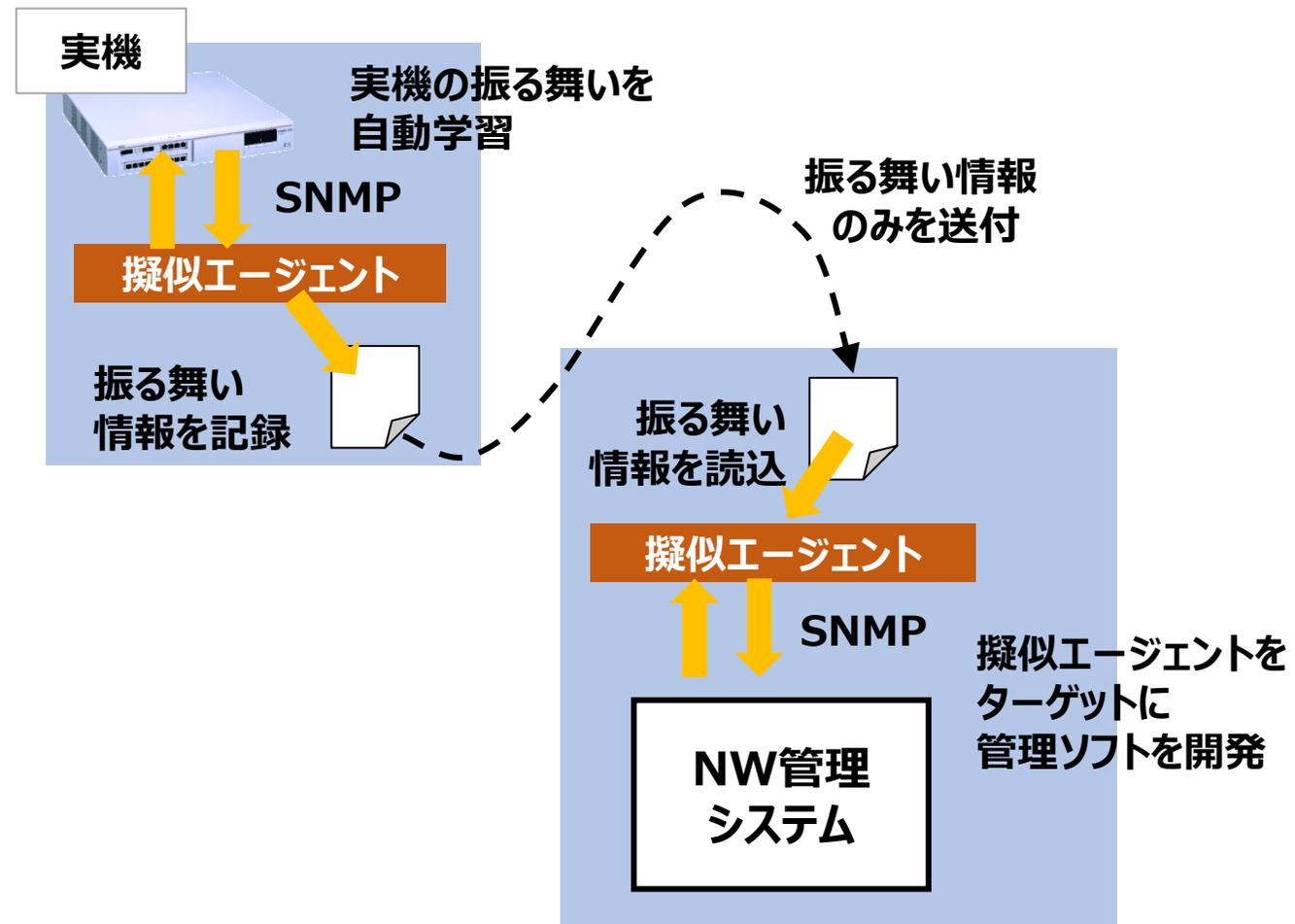
仕事：SNMP疑似エージェント（1995年頃?）

ネットワーク管理システムの開発。

管理対象の機器をなかなか貸してもらえない。
借りてもすぐに返せと言われる。
開発・評価期間がみじかい・・・。

機器のSNMP情報を自動学習して、
機器もどきとしてふるまう
疑似エージェントを開発。

これも暇なときに遊びで作ってたな...



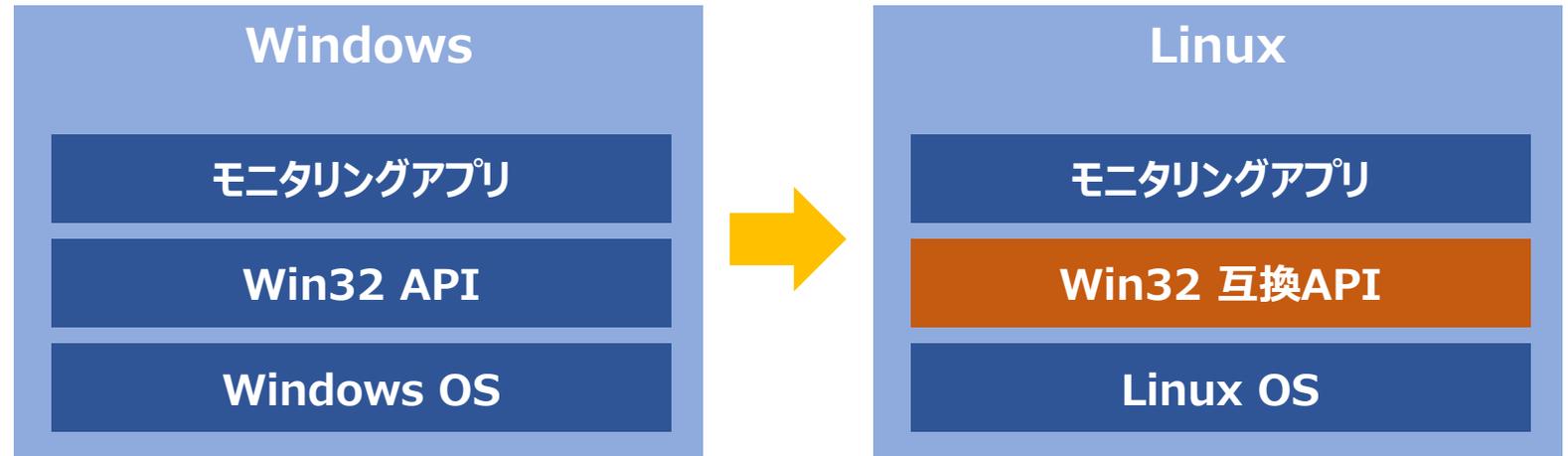
仕事：Win32 API 互換ライブラリ（2003年頃?）

セキュリティ管理システムの開発。

Windows で開発したモニタリングアプリをLinux に移植。

移植するのも面倒なので、必要最低限の **Win32 API互換ライブラリ** を開発。

パス名の ¥ を / に変更するくらいで、ほぼ動きました。



これは冬休みの遊びで...

OS・MW依存部と非依存部の分離



重要な業務ロジック部分を、OS、ミドルウェア、フレームワークに依存しないコードで開発する。

OS・MW・FWの変更やバージョンアップに追従しやすくなる。

例) 業務ロジックでは、myapp 以外のインポートを認めない。

```
import myapp.xxx
import myapp.yyy
import sys           ← NG
import Django       ← NG
```

仕事？遊び？

まだ残業規制とか厳しくない時代
仕事なんだか、趣味なんだか
とにかく **楽しく** やってました

先輩とケンカしたこともありました
このモジュールは
俺が作る！ いや俺が！

「とほほのWWW入門」 開設

「とほほのWWW入門」開始（1996年）

- 1996年8月16日
 - プロバイダ BIGLOBE に加入
- 1996年9月10日
 - 「とほほのHTML入門」開設
- 1996年9月24日
 - 「とほほのJavaScript入門」開設
- 1996年10月25日
 - 「とほほのWWW入門」に改名

HTML に関する情報って少ないなあ

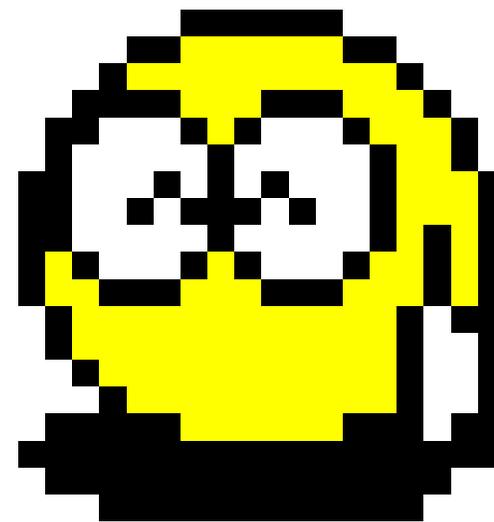
仕方無いので自分でまとめるかあ



The screenshot shows the homepage of the website 'とほほのWWW入門'. The title is written in red and blue text with a small yellow character icon. Below the title, there are two red stars and the text 'とほほのWWW入門' in blue. The main content area contains text in Japanese: 'HTML/JavaScript/CSS/CGIなど Webサイト作成に関する情報を満載した Web作成関連総合サイトです。' followed by a list of links: 'とほほの点字入門', 'とほほの所得税入門', 'とほほの家族紹介', 'むふふの素材集', 'とほほママの部屋', 'とほほママのおでかけスポット', 'とほほママのペイントブラシテクニック', and 'とほほママの映画館'. At the bottom, there is a copyright notice: 'Copyright (C) 1996-2022 杜甫々' and a small logo with the text 'とほほのWWW入門' and the yellow character icon.

アイコンとバナー

- なんとなく、アイコンを作成
 - 「とほほアイコン」と呼ばれますが、実は「ポップ君」という名前があります
 - 右は、汗をかいてるポップ君 (pop汗.gif)
- なんとなく、バナーを作成
 - 当時は 88×31 のバナーばかりでした
- 原色を使うのはデザインとしてけしからんという風潮のなか、絵心が無いので、ガンダムのような色使いで作成

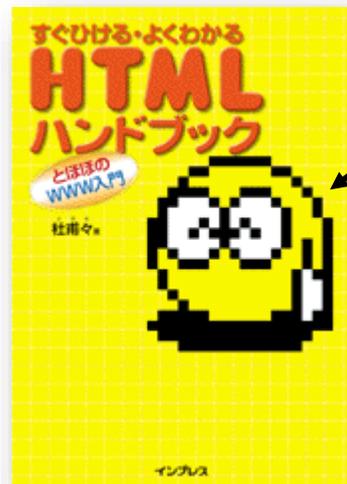


popase.gif



見た目のインパクトって重要

- ぱっと見ただけで覚えてもらえるってすごく重要なのかも



こいつが目立つ

サイトの内容をそのまま
書籍化しただけ

中古最高価格
¥225,100-



書籍用に新たに
書き下ろし

中古価格
¥539-

尊敬する二人の佐藤さんの一人目
佐藤可士和さんのデザイン



各社公式サイトから引用

もう一人は CMプロデューサーの 佐藤雅彦 さん
だんご3兄弟、ピタゴラスイッチ、サントリーモルツ、
湖池屋スコーン、バザールでゴザール、NEC文豪、
カロラーIIに乗って

Yahoo!のクールマーク（1999年）

- Yahoo! の「クールマーク」を獲得
- 当時はまだディレクトリ型検索が主流
- 管理者がカテゴリツリー上に各種サイトを手動登録
- 特にオススメサイトにはサングラスの「クールマーク」



<https://internet.watch.impress.co.jp/static/column/yyst/2004/12/03/>
より引用

ラウンジ（1997年～2000年）

- 1997年ラウンジ開設
 - Web作成に関わる質問 & 回答コーナー
 - Yahoo!知恵袋 のようなもの
- 良かった点
 - 初心者はこういう点で悩むんだと気づかせてもらいました
- 困っていた点
 - 初心者がついつい同じ質問をしてしまうことがどうしても許せない人達がありました

HTML改行の隙間について

[\[上に\]](#) [\[前に\]](#) [\[次に\]](#)

匿名希望 1997/06/16(月) 17:14:34

HTMLで改行すると、ブラウザで表示したときに「こん なふう」に隙間があいてしまうのですが、これを空かなくする方法はありませんか？

Wyas_System [\[E-Mail\]](#) [\[HomePage\]](#) 1997/06/16(月) 23:36:26

html で改行とは
のことでしょうか？

によって多少表示が変わってくるのでなんとも言えませんが、
ひとつだけ、応用として、<TABLE>タグを使って、
表の線を見えなくする方法などありますよ！

Wyas_System [\[E-Mail\]](#) [\[HomePage\]](#) 1997/06/16(月) 23:45:38

修正：表の線を見えなくするって、言ってることが違いました。
ごめんなさい。

Soma-papa [\[E-Mail\]](#) [\[HomePage\]](#) 1997/06/18(水) 19:54:55

「HTMLで改行」って、ちょっと長い文書を書くときに、「リターン」キーを押して入れるやつですね。仕様では無視されるはずなのに、「リターン」コードが入っていると、ちょっと空白が空く現象は、私も悩んでいます。解決策があれば、私も是非知りたいです。

Bunny4 [\[E-Mail\]](#) [\[HomePage\]](#) 1997/06/19(木) 12:17:10

改行しない事です。

ソースがどうあれ、表示されるHTMLに重点を置いてます、わたしは。

日本最古の「スレッドフロート型掲示板」 ???

ウィキペディア
フリー百科事典

スレッドフロート型掲示板

1の言語版

ページ ノート 閲覧 編集 履歴表示 ツール

出典: フリー百科事典『ウィキペディア (Wikipedia)』

スレッドフロート型掲示板（スレッドフロートがたけいじばん）とは、最新のコメント投稿がなされたスレッドがその掲示板のスレッド群の最上位に表示される機能を持つ**電子掲示板**を指す。**フローティングスレッド型掲示板**とも言う。

システム概要

掲示板そのものを主とする概念とは違い、掲示板の中におけるトピックや話題毎の**スレッド**と呼ばれるまとまりを主とする。スレッド一覧を表示する際の順序は、各スレッド内における最終投稿時間を用いる。投稿が行われた時点で、下に沈んでいった該当スレッドが一覧の最上位に浮上（float）していくように見えるためこう呼ばれる。

単純にスレッドが上がる形式の掲示板のみについてではなく、「あめぞう型掲示板」を指して「スレッドフロート型」、「マルチスレッドフロート型」と称する事が多い。

歴史

- 1998年2月 新規投稿されたスレッドが一番上に表示されるフロート型の掲示板スクリプト「multi2」が「terra（テルラ）」によって作られ本人のサイトTry the Homepage上で公開配布。日本初のフロート型掲示板と見られる。^[要出典]
- 1998年9月6日、「あめぞうリンク」の掲示板がterra作成で同じくフロート機能のある「res bbs」に変更される。
- 1999年1月頃、複数スレッドの題名がリスト表示されるヘッドライン機能をつけた「res bbs」改が「あめぞう」によって作成され、あめぞうリンクの掲示板に導入される。
- 1999年1月中旬から下旬頃（または1998年10月説もあり）、「レスキュー」作成である「MiniBBS（正式名は簡易BBS）」にスレッドフロート、複数スレッドの題名リスト表示に加えレスが裏に回る機能、つまりスレッドの1と最新レスのいくつかを表示して中間のレスを省略表示する機能を付け加えた「MiniBBS」改のスクリプトを「あめぞう」が開発し、あめぞうリンクの掲示板に使用される。

<https://ja.wikipedia.org/wiki/スレッドフロート型掲示板>

ひろゆき
@hirox246

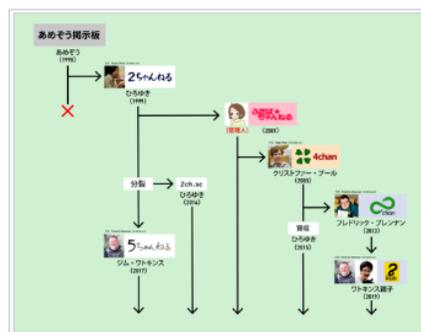
あめぞう掲示板のスクリプトが流出したのは、2chが出来た後だったかと。んで、勉強がてらに作った掲示板スクリプトを改造したので、本当の2chのベースがあるとしたら、「とほほのWWW入門」にあった掲示板スクリプトですね。

ひろゆきさん曰く：
「2チャンネルのベースはしいて言えば、
とほほの掲示板」らしい...

<https://twitter.com/hirox246/status/1579726433089642496>

■ 歴史

- とほほのラウンジ設置（1997年6月）
- とほほのラウンジソース公開（1997年9月）
- multi2（1998年2月）
- あめぞう掲示板（1998年6月）
- 2ちゃんねる（1999年）



あめぞうが発明したとされるスレッドフロート型掲示板（匿名掲示板）の系譜。あめぞうは2ちゃんねるの親サイトであり、4chanや8chanを含む、他すべてのchanサイトの源流、または雛型となった。

Webカウンター（1997年）

- カウンター用ライブラリ gifcat.pl を公開
- 当時は Unisys が GIF の圧縮に関する特許を持っていたため、フリーの画像連結ソフトが禁止されていた。
- Unisys の特許に抵触しないよう、GIF アニメ技術を用いて、疑似的に画像を連結

キリ番ゲットだぜ!!

あなたは **007566** 人目のお客様です。

それぞれの桁がアニメーションフレーム
画像データの圧縮・伸長を行わない

全文検索エンジン（grep型）（1997年）

- 毎回 Grep 型の検索エンジンを公開
 - 比較的小さなサイトに適用
 - Grep 型の中では当時では比較的高速
 - <https://cinema1987.org/data/howto.htm>

検索 [戻る]

[とほほのWWW入門](#)
セキュリティ (1) **CORS** (1) 同生成元ポリシー (1) JSONP (1)

[とほほのWWW入門 - とほほのWWW入門](#)
セキュリティ **CORS** 同生成元ポリシー JSONP

[CSSリファレンス - とほほのWWW入門](#)
フォントファイルは通常、同じドメインのサーバからしかダウンロードできません。他のドメインからダウンロードするには、**CORS**(Cross-Origin Resource Sharing)を用います。

[CSS - @font-face - とほほのWWW入門](#)
font-family で指定可能なWebフォントを定義します。フォントファイルは通常、同じドメインのサーバからしかダウンロードできません。他のドメインからダウンロードするには、**CORS** を指定してください。
CSS

[CSS - <image> - とほほのWWW入門](#)
仕様<https://drafts.css-houdini.org/css-paint-api/> 描画メソッドを持つクラスを記述した JavaScript ファイルを用意し、CSS.paintWorklet.addModule() で読み込む必要があります。**CORS** 制限によりローカルファイルを読み込むことはできません。JavaScript は Canvas 2Dリファレンス を参照してください。
example.html

とほほ死亡騒動（2009年）

- ネットで「とほほ死亡説」が流れる
- 亡くなられたのは、たまたま同じ広島で、たまたま同じハンドルで、別の活動されていた別の とほほ さん
- 杜甫々が亡くなると、非常に多くの方からお悔やみの言葉をいただきました
- 自分が生きている内に、自分が死んだときのまわりの言葉を聞くことができた貴重な機会だったかも

スタイルシート論争（2000年）

- 1996～2000年 W3C が HTML4.0/CSS/XHTML を発表。
- HTML4.0 の「目の見えない人にも意味のあるサイトを」という「アクセシビリティ」に強く共感する人達が多くあらわれる。

`HTML`

「太字」ってのは目の見えない人にとっては意味がないので使うべきではない。

`HTML`

「文字の色」ってのは目の見えない人にとっては意味がないので使うべきではない。

`
`

XHTML : HTMLをXMLに完全準拠させたもの

スタイルシート論争（2000年）

- W3C（WWW Consortium）が提唱する HTML 4.0 / XHTML と CSS が広まり始める中、「見栄えと意味は分離すべき」という W3C の理想の将来像に感化された方々との間に論争が・・・。
- （企業のWebサイトならともかく）（すくなくともブラウザの実装が安定するまでは）個人サイトにまで HTML 4.0 + CSS を強要すべきではないと言ったつもりが・・・。
 - 薄字部分の言葉が足りていなかったかも。反省。
- W3C、HTML 4.0、CSS、見栄えと意味の分離に対する批判であると誤解されて論争になってしまいました。
 - まあ、多少、批判してましたけど・・・。
- ラウンジも荒れてきて、2000年6月、ラウンジ廃止。

スタイルシート論争（2000年）

- HTML4.0/XHTML だけが正しいんだという主張に対して、当時、サイトで、こんな記事を投稿していました。

2000年3月

もし、FLEX-HTML1.0が存在したら...

もし、誰かが **W4C** という団体を設立し、FLEX-HTML1.0 (Flexible, Loose, Easy to learn and eXtensible HTML) という仕様を勧告したとします。

FLEX-HTMLでは、過去の**歴史的なタグをすべてサポート**します。~ を ~ で囲んではならないなどの、**難しい規則をできるだけ排除**します。**SGMLアプリケーションであることも放棄**します。**タグの拡張性**ももたせます。ユーザーエージェントは自分の解釈できるタグや属性のみを解釈します。最低限解釈できなくてはならないタグなどを3段階程度のレベルに分けます。

そんな時、なにが「正しいHTML」か？ということですが・・・

HTML4.0 → HTML5 → HTML ?

- HTML 4.0 は HTML5 で方向性を転換（2008～2014年）
- ちょっと無理をしすぎた XHTML は廃案（2009年）
- HTML5 の主管も W3C から WHATWG に移管（2019年）

WHATWG : Web Hypertext Application Technology Working Group

HTML5 はすでに終わってる？

- 現在の HTML の最新版は HTML5 ではありません
- 現在は HTML Living Standard が標準仕様として参照されています



バージョンレスで日々改定される

HTML5登場（2014年）

・スタイルシート論争時の私の主張が結構認められた気がしています...

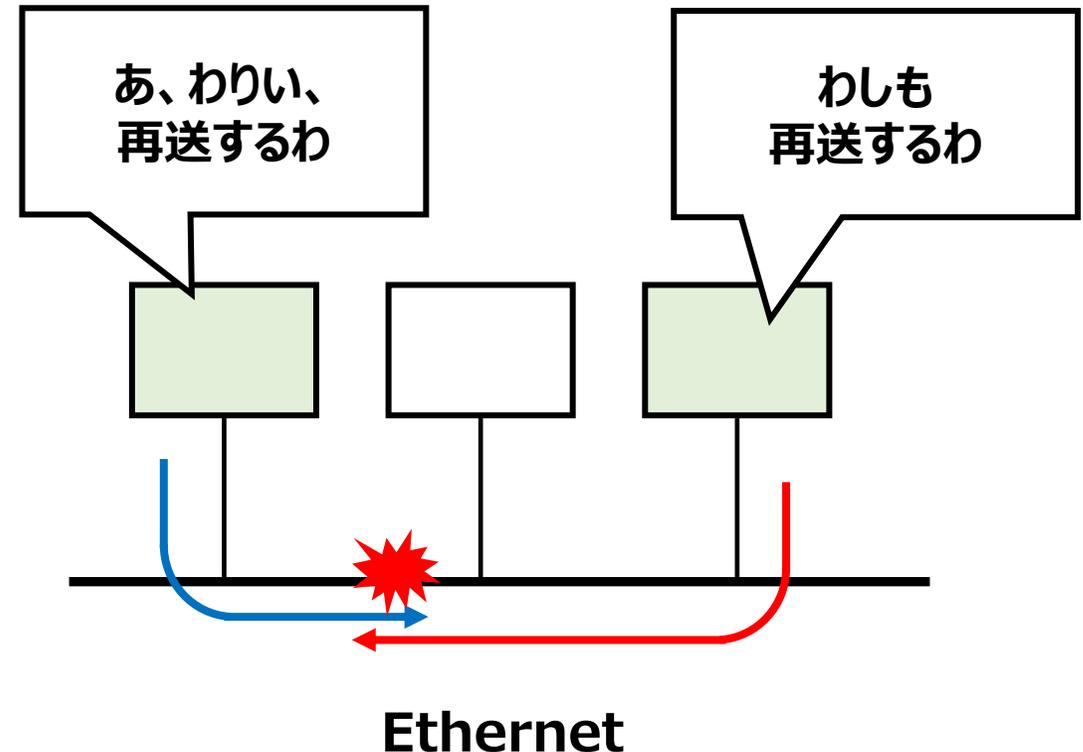
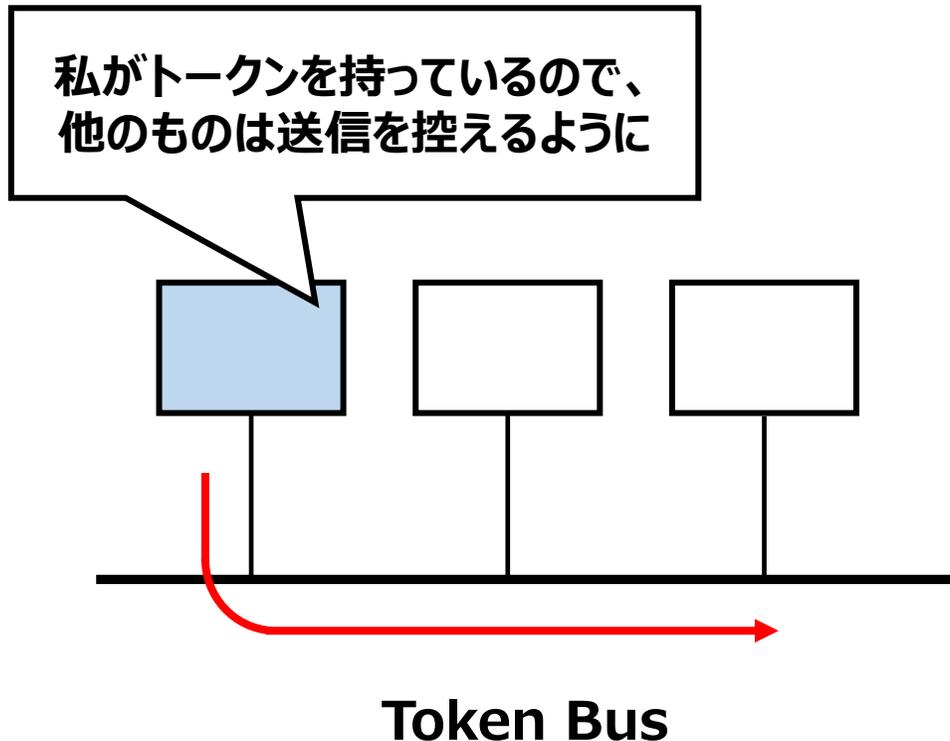
杜甫々の主張	HTML5の現在の状況
誰かが W4C という団体を...	Apple, Mozilla, Operaなどが WHATWG という団体をつくりました。W4C ではなく W5G でしたけど...
過去の歴史的なタグをすべてサポート	ブラウザは過去の機能もサポートすべきという方針に変わりました。などは少し意味を変えて残されました。
難しい規則をできるだけ排除	かなり排除されました。
SGMLアプリケーションであることも放棄	放棄されました。
タグの拡張性も持たせます	機能を拡張することができるようになりました。
仕様策定だけを先行するのではなく、業界の開発と歩調を合わせるべき	実装事例と歩調を合わせるようになりました

普及する仕様と、普及しづらい仕様

- **HTML 4.x / XHTML → HTML 5**
- **IPv6** 普及しつつあるけど時間かかりすぎた
- **XML/SOAP → JSON/REST**
- **Token Ring/Bus → Ethernet**
- **壊れないコンピュータ → 1部が多少壊れても代替機で稼働**

Token Ring/Bus → Ethernet

- LAN 上でパケットが衝突を起こさないように・・・
 - Token Ring/Bus : トークンを持つホストのみが送信を許される
 - Ethernet : 送ってみて、衝突したら再送すればいいや (ルーズさの勝利)



SOAP/XML → REST/JSON

- SOAP/XML

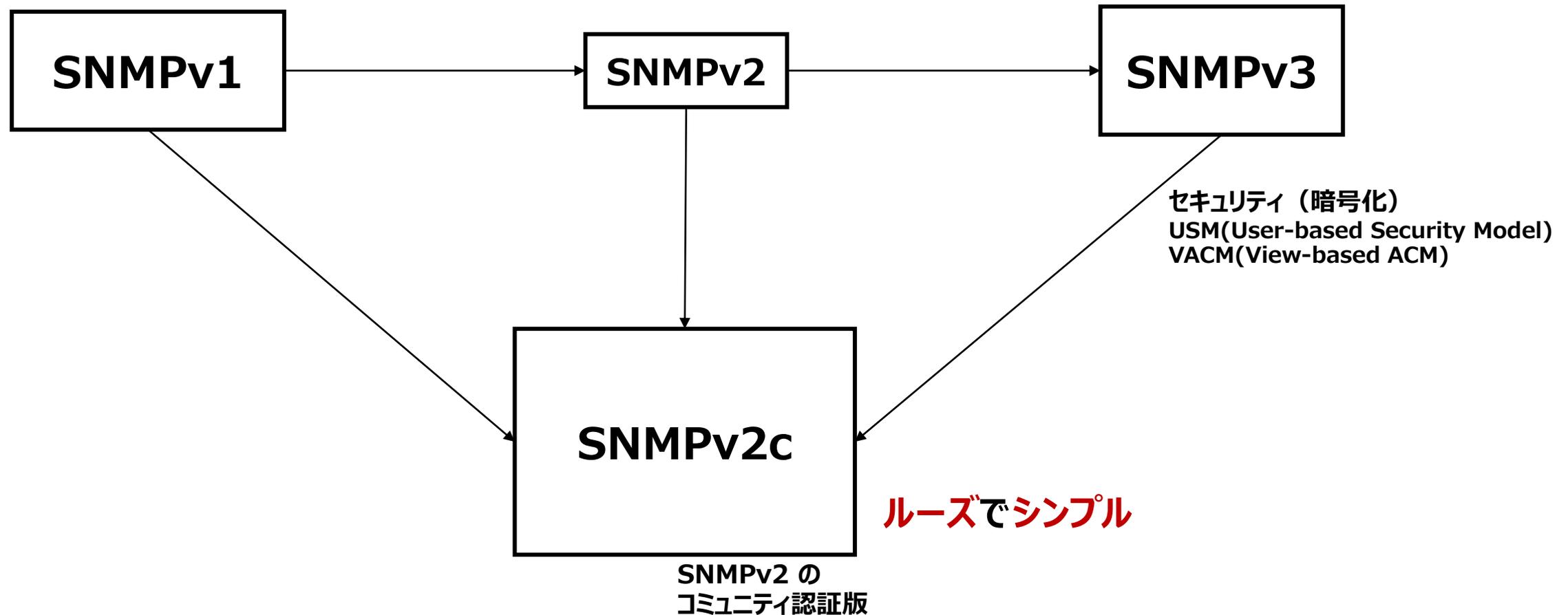
```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <getProductDetails xmlns="https://warehouse.example.com/ws">
      <productId>827635</productId>
    </getProductDetails>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- REST/JSON (シンプル)

```
curl https://warehouse.example.com/ws/products -d '{
  "productid": "827635"
}'
```

SNMPv3 → SNMPv2c

- 紆余曲折の末ようやく仕様確定した SNMPv3 だがあまり使用されていない



IPv6の苦悩

- 「良い仕様」を目指しすぎて、IPv4との互換性をないがしろにしてしまった？
 - 32ビットアドレスの拡張がメインであれば、IPv4 のオプションでアドレス拡張してもよかつたのでは？
 - そうすれば、IPv6未対応のルーターでも転送可能

133.x.y.z
192.168.2.34

IPヘッダ

IPオプション

IPデータ

IPv6アドレス：面倒

2001:0DB8:AC10:FE01::C0A8:0222

IPv4アドレスの拡張：シンプル

133.x.y.z:192.168.2.34

普及する仕様

多少**ルーズ**でも**シンプル**なものは普及しやすい
厳格で理想を求めすぎたものは普及までの敷居が高い

「広まるもの」も「よい仕様」の必須条件

広まらなければただのゴミ

プログラミング

今まで学んだプログラミング言語

• 学んだプログラミング言語

6502アセンブラ

Z80アセンブラ

N88-BASIC

FORTLAN

COBOL

Pascal

LISP

C言語

C++

C#

Java

Bash

PowerShell

AWK

Perl

PHP

Ruby

Python

JavaScript

Go

Rust

Scala

Haskell

Kotlin

• よく使う言語

- BASIC → C言語 → C++ → Perl → PHP → Python, JavaScript

- たまに AWK

この言語のこんなところが好き (Perl/AWK)

- Perl の正規表現を簡単に扱えるところ

```
if ($a =~ /(¥d+)-(¥d+)-(¥d+) (¥d+):(¥d+):(¥d+)/) {  
    $year = $1; $mon = $2; $mday = $3;  
    $hour = $4; $min = $5; $sec = $6;  
}
```

- AWK でコマンドの出力カラムを簡単に扱えるところ

```
$ ls -l / | awk '{ printf("%8d %s¥n", $5, $9) }'  
4096 etc/  
4096 home/  
4096 tmp/  
4096 usr/
```

この言語のこんなところが好き (JavaScript)

- JavaScriptの関数を引数にできるとこ

```
Window.setTimeout(() => {  
    console.log("Time!");  
}, 3000);
```

- すべての関数が非同期で実装されているとこ

```
Window.fetch(url)  
    .then((res) => {  
        console.log(res);  
    });
```

- 上位互換性がたもたれているとこ

- ブラウザで動作するということが、互換性を保つ要因のひとつになっている

この言語のこんなところが好き

- Python の {...} じゃなしにインデントでブロックするところは...
あまり好きじゃないけど、ようやく慣れてきた

```
if x > 10:  
    print("Big!")
```

以前、goto 文で戦ったことがあります

- try catch 構文の無いC言語での開発。
- goto 文は使用すべきではないという風潮・教育の中、一人だけ（下記の使用法に限って）goto 文使用すべきと主張。

```
int func(arg) {
    (...略...)
    if (error = getFilePath(arg, &fileName)) {
        goto done;
    }
    if (error = fileOpen(filename, &fp)) {
        goto done;
    }
    if (error = readFile(fp)) {
        goto done;
    }
done:
    if (fp) closeFile(fp);
    return error;
}
```

こんな構文欲しいな...

- 関数の途中で即時関数を簡単に呼び出す

```
func getContent(base, path) {  
    var protocol = getEnv("PROTOCOL");  
    var url = protocol + "://" + base + "/" + path;  
}
```

← 値を求めるための一時的な変数が本文中に出現してしまう



こんな風に書きたい

```
func getContent(base, path) {  
    var url = (base, path) {  
        var protocol = getEnv("PROTOCOL");  
        return protocol + "://" + base + "/" + path;  
    }  
}
```

← base と path から url を求めるんだということが明示される

AWKもなにかと便利なので使ってます

```
BEGIN {
    print "---- BEGIN ----";
}
/¥.log$/ {
    printf("%8d %s¥n", $5, $9);
}
END {
    print "---- END ----";
}
```

最初に ---- BEGIN ---- を表示

.log で終わる行であれば、5カラム目と9カラム目を表示

最後に ---- END ---- を表示

```
$ ls -l | awk -f test.awk
---- BEGIN ----
1962972 myapp_20240201.log
 925307 myapp_20240202.log
1184021 myapp_20240203.log
1587000 myapp_20240204.log
---- END ----
```

ファイル一覧のサイズとファイル名のみを表示

```
$ ls -l *.log | awk '{z+=$5};END{print z}'
5659300
```

*.log ファイルのサイズ合計を算出

PerlもAWKの様に使うことができた (-an オプション)

```
BEGIN { print "---- BEGIN ----\n"; }  
{  
    if (/\.log$/) {  
        printf("%8s %s\n", $F[4], $F[8]);  
    }  
}  
END { print "---- END ----\n"; }
```

```
$ ls -l | perl -an test.pl  
---- BEGIN ----  
1962972 myapp_20240201.log  
 925307 myapp_20240202.log  
1184021 myapp_20240203.log  
1587000 myapp_20240204.log  
---- END ----
```

```
$ s -l *.log | perl -ane '$z+=$F[4]; END{print "$z\n"}'  
5659300
```

フレームワーク

今まで学んだフレームワーク

• 学んだフレームワーク

Ruby on Rails

Drupal

Django

Flask

Laravel

CakePHP

AngularJS

Angular

Vue.js

React

Next.js

• よく使うフレームワーク

- CakePHP → Django/Flask → 自作

フレームワークに頼るのは実はあまり好きじゃない...

- なにを使ってもすぐに陳腐化...
- 依存パッケージが多くサポートが心配...
- バージョンアップへの追従にコストがかかる...

OSSのサポート期間って短すぎませんか？

- EndOfLife.date

- 様々なOSSのサポート期限を調べられる

- Node.js

- 2022年4月リリースのLTS版が2025年4月にはサポート切れ

LTS : Long Term Support

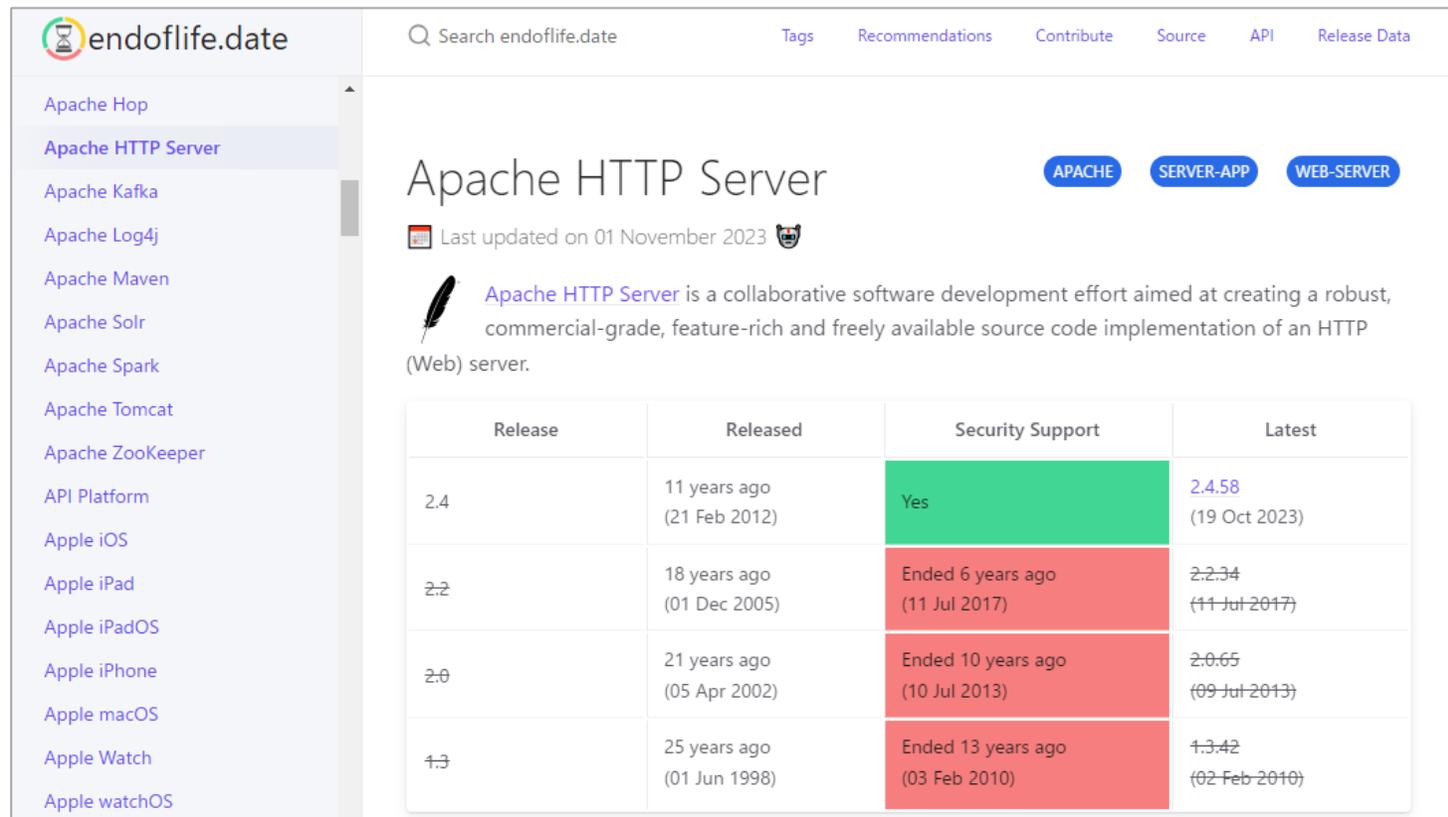
Release	Released	Active Support	Security Support	Latest
21	3 months and 2 weeks ago (17 Oct 2023)	Ends in 1 month and 4 weeks (01 Apr 2024)	Ends in 3 months and 4 weeks (01 Jun 2024)	21.6.1 (22 Jan 2024)
20 (LTS)	9 months ago (18 Apr 2023)	Ends in 8 months (22 Oct 2024)	Ends in 2 years and 2 months (30 Apr 2026)	20.11.0 (09 Jan 2024)
19	1 year and 3 months ago (18 Oct 2022)	Ended 10 months ago (01 Apr 2023)	Ended 8 months ago (01 Jun 2023)	19.9.0 (10 Apr 2023)
18 (LTS)	1 year and 9 months ago (19 Apr 2022)	Ended 3 months and 2 weeks ago (18 Oct 2023)	Ends in 1 year and 2 months (30 Apr 2025)	18.19.0 (29 Nov 2023)
17	2 years and 3 months ago (19 Oct 2021)	Ended 1 year and 10 months ago (01 Apr 2022)	Ended 1 year and 8 months ago (01 Jun 2022)	17.9.1 (01 Jun 2022)

Major Node.js versions enter Current release status for six months, which gives library authors to add support for them. After six months, odd-numbered releases (9, 11, etc.) become

<https://endoflife.date/>

LLTS版をサポートして欲しい (Long Long Term...)

- Apache 2.4 は11年以上サポートを継続。すばらしい



The screenshot shows the endoflife.date website for Apache HTTP Server. It includes a search bar, navigation links (Tags, Recommendations, Contribute, Source, API, Release Data), and a table of releases. The table has columns for Release, Released, Security Support, and Latest. The 2.4 release is highlighted in green, indicating active security support.

Release	Released	Security Support	Latest
2.4	11 years ago (21 Feb 2012)	Yes	2.4.58 (19 Oct 2023)
2.2	18 years ago (01 Dec 2005)	Ended 6 years ago (11 Jul 2017)	2.2.34 (11 Jul 2017)
2.0	21 years ago (05 Apr 2002)	Ended 10 years ago (10 Jul 2013)	2.0.65 (09 Jul 2013)
1.3	25 years ago (01 Jun 1998)	Ended 13 years ago (03 Feb 2010)	1.3.42 (02 Feb 2010)

Nginx 1年
Node.js 3年
React 7年以上
Angular 1年半
Vue.js 6年くらい?
Laravel 2年
Django 3年
Ruby on Rails 4年くらい?

フリーレンと言われそうだけど、
もう少し長くサポートしてほしい

<https://endoflife.date>

自作フレームワーク

- **使用するOSSを最小限にとどめる**
 - Nginx - Webサーバ
 - uWSGI - WebサーバとPythonを連携
 - Python - プログラム
 - SQLAlchemy - O/Rマッパー
 - Jinja 2 - テンプレートエンジン
- **主な機能**
 - 基本的にはDjangoの簡易版
 - URLルーティング
 - プラグイン機能

JSON

JSON は YAML の ほぼ サブセット

JSON は実は、YAML の ほぼ サブセットらしい。

JSON

```
{
  "users": [
    { "name": "Yamada", "age": 26 },
    { "name": "Tanaka", "age": 32 },
    { "name": "Suzuki", "age": 43 }
  ]
}
```

YAML (ブロックスタイル)

```
users:
- name: "Yamada"
  age: "26"
- name: "Tanaka"
  age: "32"
- name: "Suzuki"
  age: "43"
```

YAML (フロースタイル)

```
{
  users: [
    { name: "Yamada", age: "26" },
    { name: "Tanaka", age: "32" },
    { name: "Suzuki", age: "43" }
  ]
}
```

すべてのコマンドで `--json` オプションサポートして欲しい

```
$ ls --json
[
  {"permission": "-rw-r--r--", "link": 1, "user": "yamada", "group": "yamada",
   "size": 1234, "date": "2024/02/10 17:30:00", "name": "test.txt" },
  :
]
```

jq コマンドなどと組み合わせて正確なデータ処理ができそう。

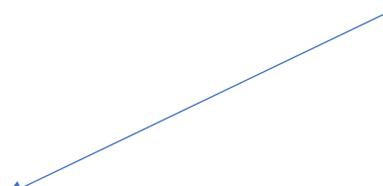
```
$ ls --json | jq 'map(select(.size > 1000))'
[
  {"permission": "-rw-r--r--", "link": 1, "user": "yamada", "group": "yamada",
   "size": 1234, "date": "2024/02/10 17:30:00", "name": "test.txt" },
  :
]
```

JSONで末尾のカンマ(,)を許してほしい...

末尾データのカンマ(,)を許してほしい...

```
{  
  "permission": "-rw-r--r--",  
  "link": 1,  
  "user": "yamada",  
  "group": "yamada",  
  "size": 1234,  
  "name": "sample.text",  
}
```

こいつ！



Index 使える IJSON が欲しい

パラメータ名が毎行出てくるのでデータ量が増える

```
[
  {"permission": "-rw-r--r--", "link": 1, "user": "yamada", "group": "yamada", "size": ...
  {"permission": "-rw-r--r--", "link": 1, "user": "yamada", "group": "yamada", "size": ...
  {"permission": "-rw-r--r--", "link": 1, "user": "yamada", "group": "yamada", "size": ...
]
```

パラメータ名をCSVのヘッダ行のように指定したい...

```
[
  ("permission", "link", "user", "group", "size", ...)
  {"-rw-r--r--", 1, "yamada", "yamada", 1234, ... },
  {"-rw-r--r--", 1, "yamada", "yamada", 1234, ... },
  {"-rw-r--r--", 1, "yamada", "yamada", 1234, ... }
]
```

JSON5 っていうのはあるみたい

```
{
  /* Comment ... */
  // Comment ...
  'attr1': 'value1',
  attr2: "value2",
  attr3: "value¥
value",
  attr4: 0xfffe,
  attr5: Infinity,
  attr6: NaN,
  attr7: "value7",
}
```

コメントが書ける
シングルクオート(')も使用可能
変数名は "... " で囲まなくてもよい
バックslashで文字列を複数行記述できる
16進数も使用できる
Infinityも使用できる
NaNも使用できる
ケツカンマOK

<https://json5.org/>

Unicode

もうすこしシンプルにできなかったか・・・

- **文字コード：**
 - 文字を 群 × 面 × 区 × 点 で表す
 - $128 \times 256 \times 256 \times 256 = 2,147,483,648$ 文字
- **すべての文字を 2バイト (65,536文字) で表すのは無理があった**
 - 通常使う文字の範囲であれば足りたかもしれないが、自分の名前は正確に表現したい
- **サロゲートペア**
 - U+10000~U+10FFFF までの文字を
U+D800~U+DBFF / U+DC00~U+DFFF の2文字にマッピング
 - UTF-8 で表現すると 6バイト使用して1文字を表す

結構面倒な Unicode の仕様

結合文字

「ガ」1文字だったり「カ」+「゛」だったり

ガ U+30AC

ガ U+30AB U+3099

異体文字

後ろにつけると前の文字が微妙に変わる

禰 U+79B0

禰 U+79B0 U+E0100



絵文字修飾子

後ろにつけると前の絵文字の色が変わる

 U+1F466

 U+1F466 U+1F3FD

 U+1F466 U+1F3FF

結合文字 (Zero Width Joiner)

文字と文字を連結すると別の文字になる

  U+1F635 U+1F4AB

 U+1F635 U+200D U+1F4AB

何バイトで1文字なのか、計算できない・・・

もうすこしシンプルにできなかったか・・・

- **最上位ビットが立っていないバイトが文字の終わりだと楽なのに・・・**

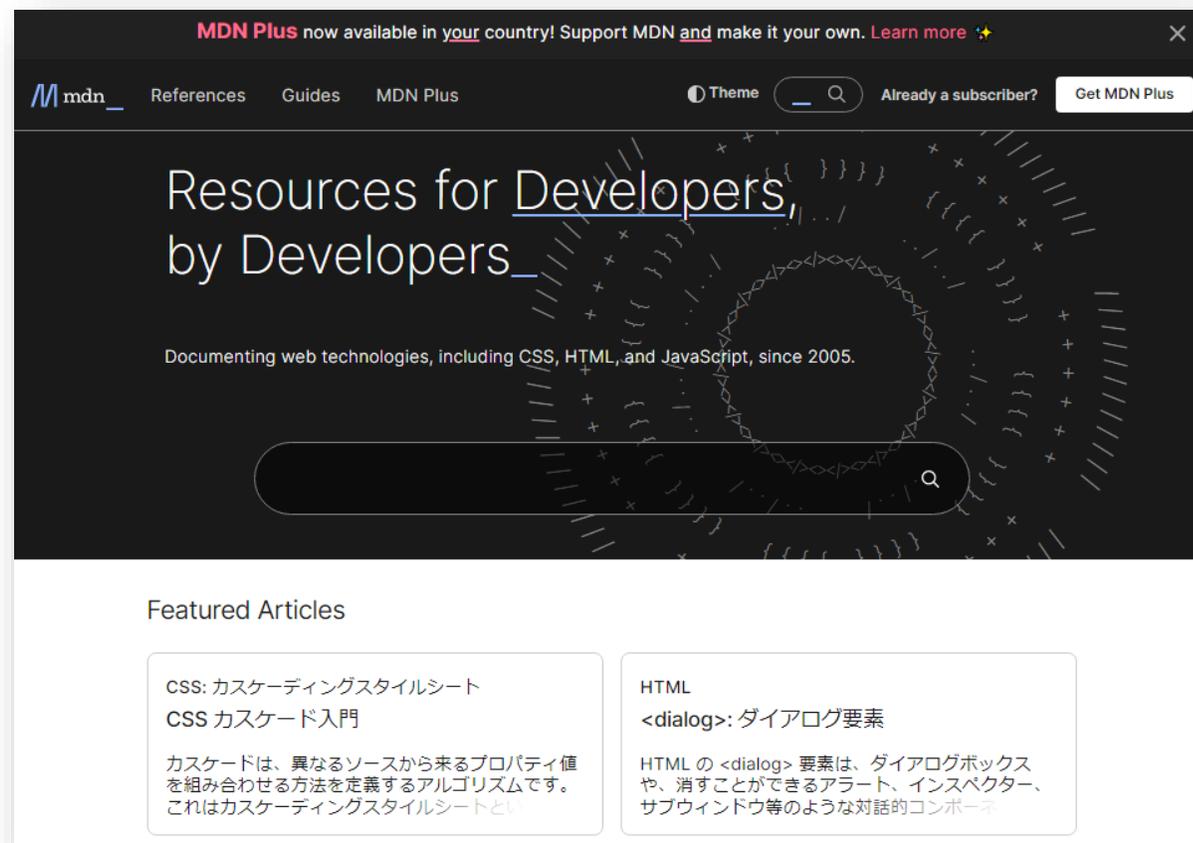
- **1バイト文字 (128文字) 0群 0面 0区 *点**
 - 0x00~0x7F
- **2バイト文字 (16,384文字) 0群 0面 *区 *点**
 - 0x80~0xFF / 0x00~0x7F
- **3バイト文字 (2,097,152文字) 0群 *面 *区 *点**
 - 0x80~0xFF / 0x80~0xFF / 0x00~0x7F
- **4バイト文字 (268,435,456文字) *群 *面 *区 *点**
 - 0x80~0xFF / 0x80~0xFF / 0x80~0xFF / 0x00~0x7F

**3バイトあれば、6バイトで表現している
U+0000~U+10FFFF のサロゲート領域を含めた
1,112,064文字を収容できちゃう**

参考にしているサイトや 情報源

HTML/CSS/JSの技術文書 (MDN)

- <https://developer.mozilla.org/>
- Mozilla Developer Network
- HTML/CSS/JS 等に係る技術文書を数多く掲載
- Mozilla (Firefox) を中心とされていたが、最近は Microsoft や Google も参画



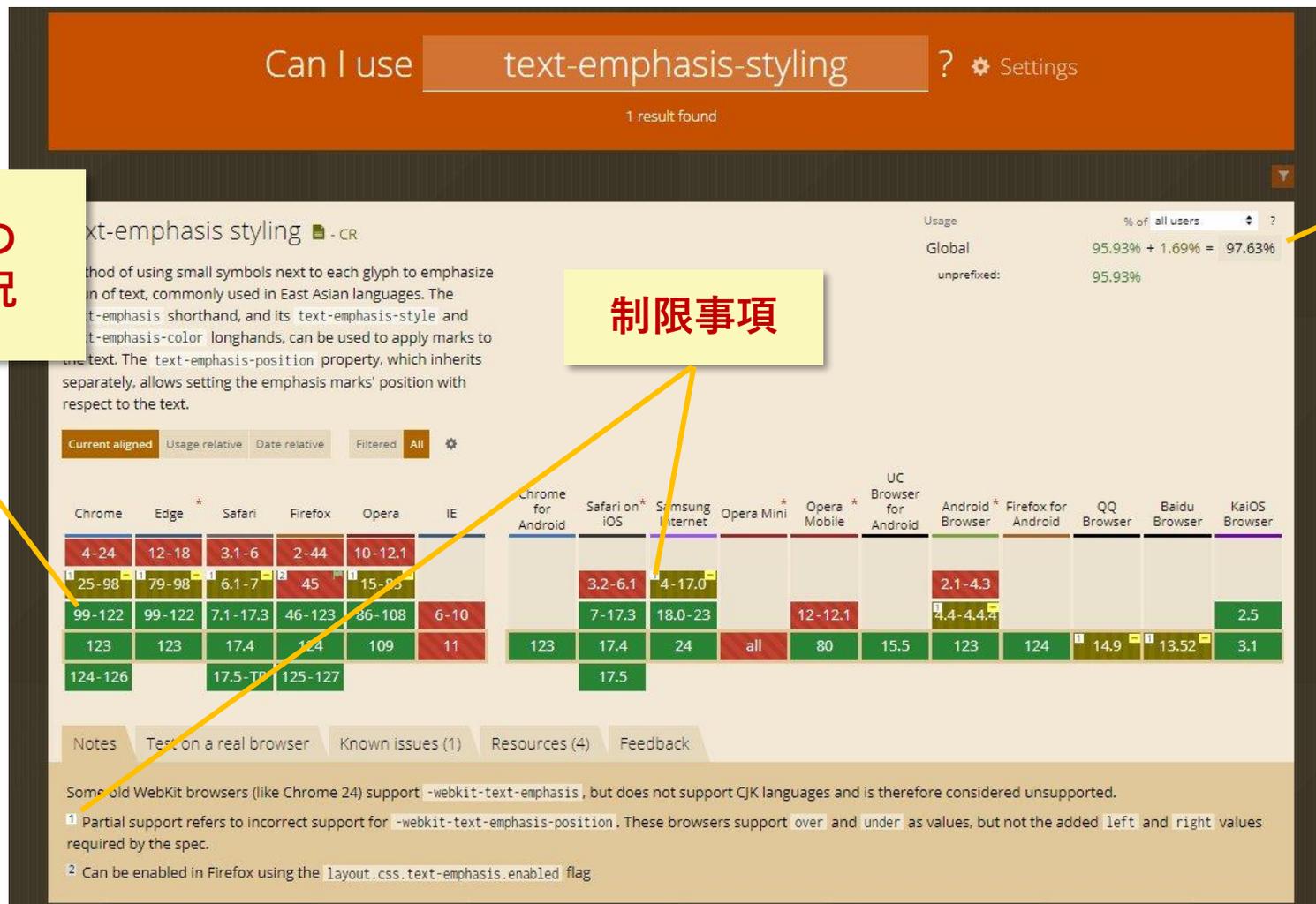
HTML/CSS/JSのブラウザサポート状況 (Can I Use)

- <https://caniuse.com/> が便利

トータルのサポート率

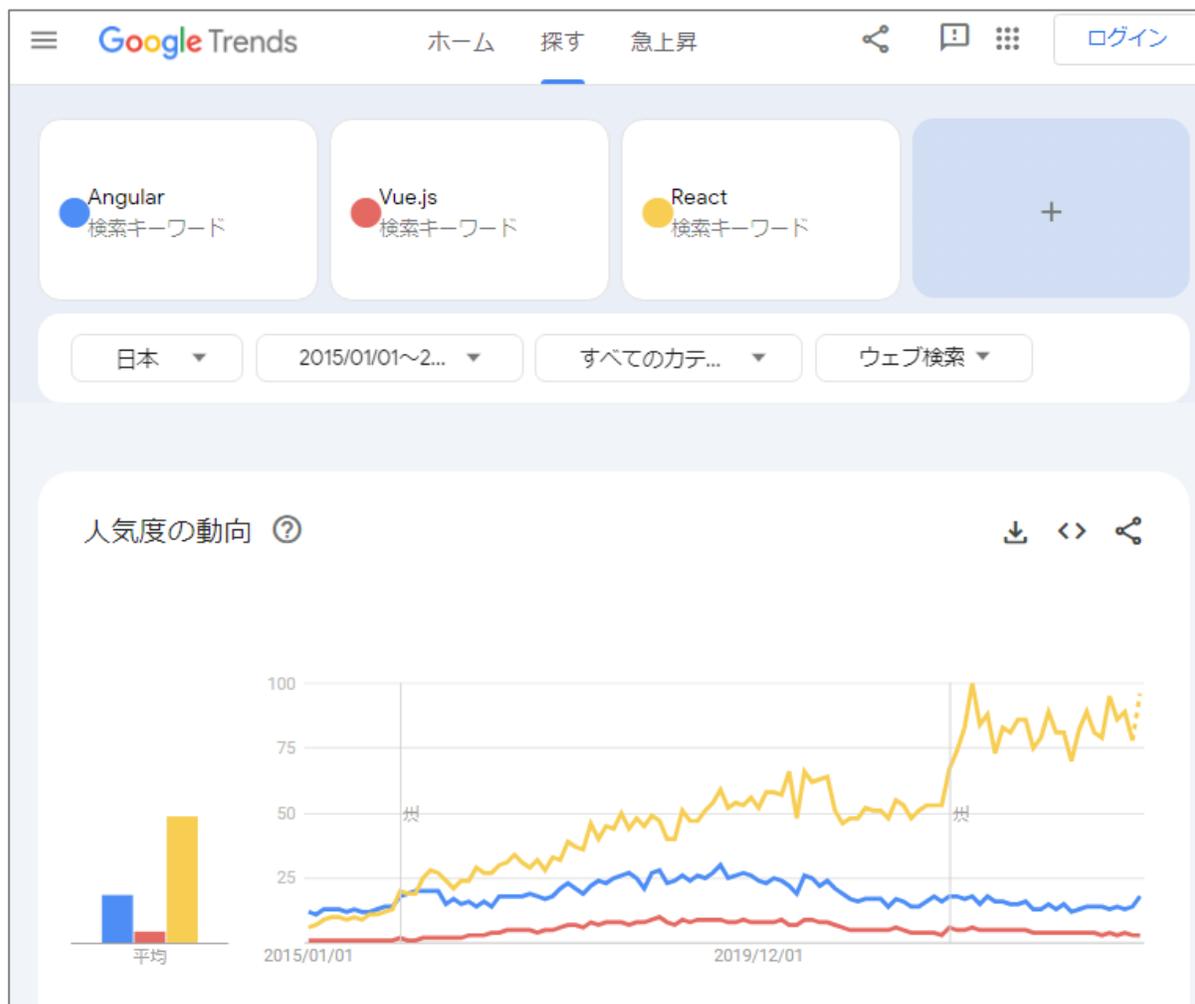
各ブラウザのサポート状況

制限事項

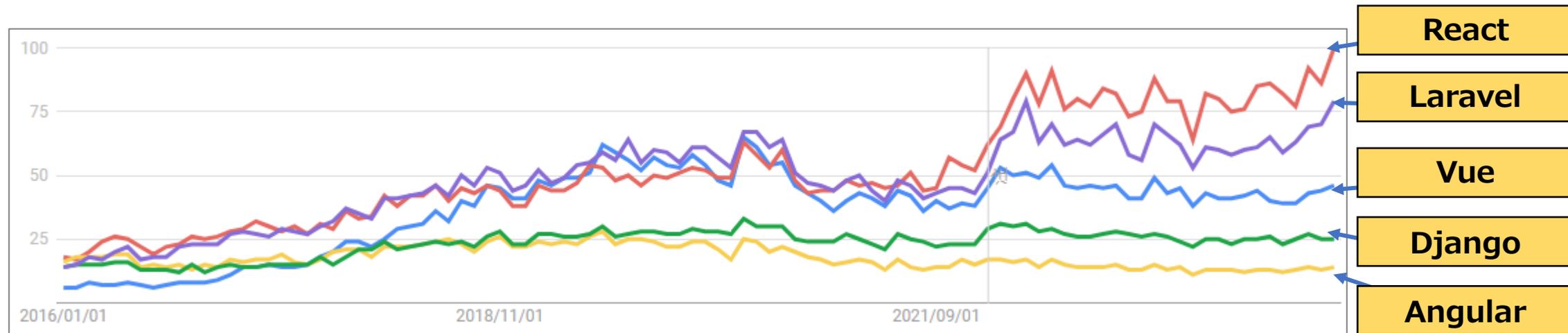
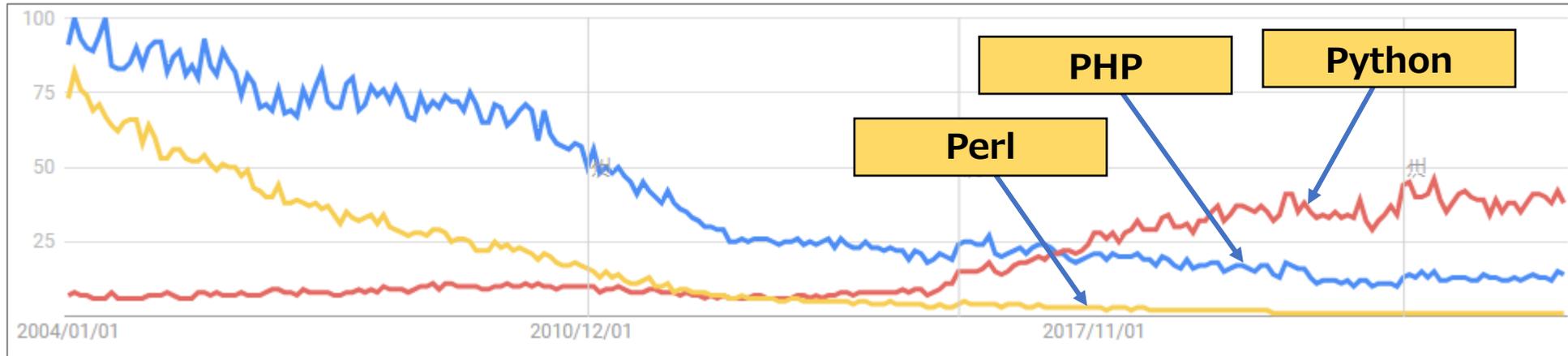


普及度の調査 (Googleトレンド)

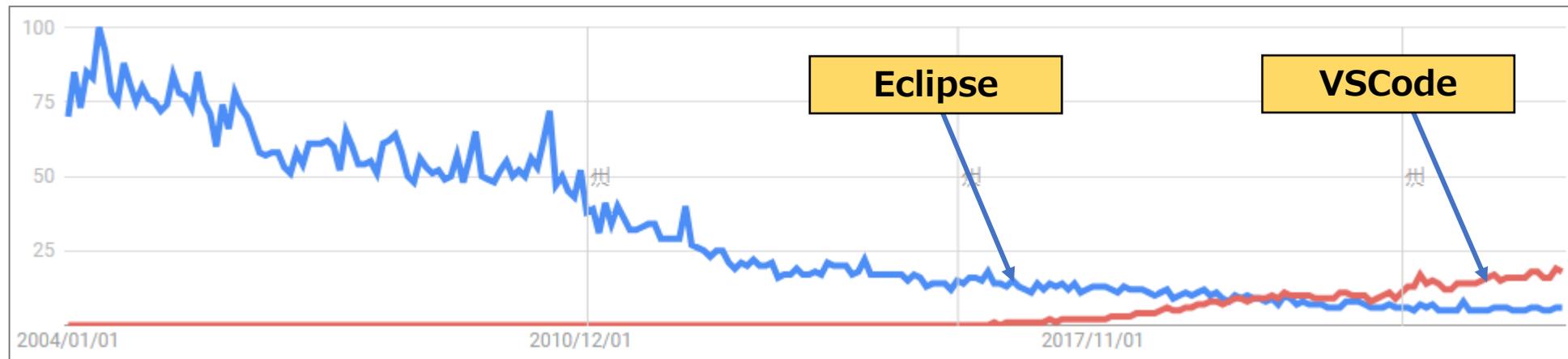
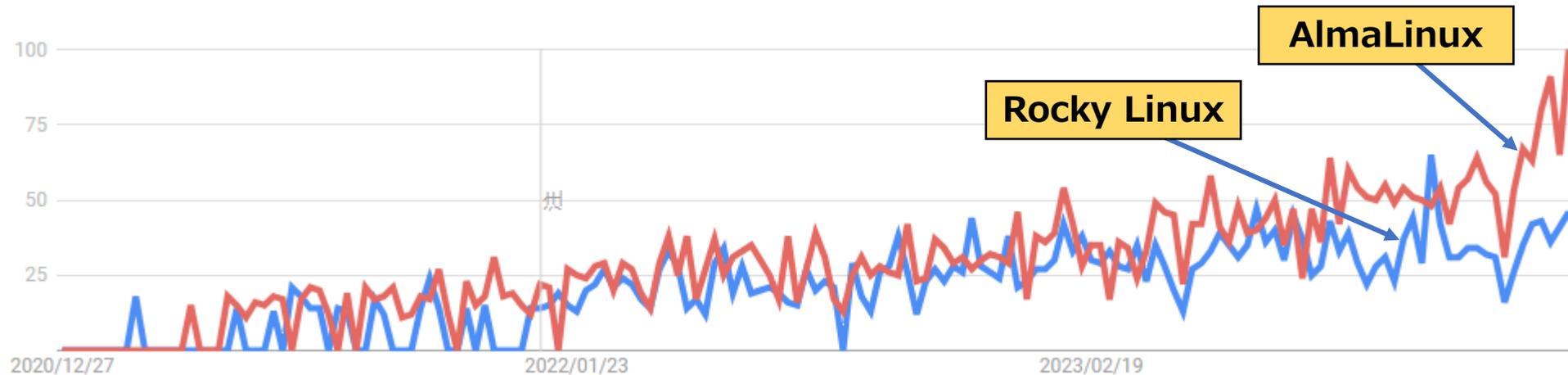
- <https://trends.google.co.jp/> が便利



普及度の調査 (Googleトレンド)



普及度の調査 (Googleトレンド)



何を参考に執筆ネタを集めているか

・ IT関連ニュース

- ITmedia
- Yahoo!ニュース
- 日経クロステック
- インプレス
- GIGAZINE
- などなど



Qiita

- Qiita のトレンド情報
- Qiita 人気記事ランキング

Qiita

記事、質問を検索

ホーム タイムライン **トレンド** Qiita Conference 質問 公式イベント 公式コラム Organization

タグランキング 週間 月間

Python 117 users

初心者 89 users

AWS AWS

すべて

最近人気の記事を毎日5時/17時に更新

@ucan-lab in 株式会社ミライトデザイン
2024年04月09日
GitHub 返信テンプレート を設定して、レビュー意図を明確に伝えよう！
Git GitHub
♡ 100

@wayama_ryousuke (ryousuke wayama)
2024年04月10日
日本語LLM 9種を量子化して回答内容を比較調査してみた
量子化 LLM
♡ 44

@qwer123123
2024年04月10日
Pythonで理解する電気回路
Python 電気回路
♡ 59

@sergicalsix (sergicalsix) in 株式会社 ARISE analytics
2024年04月08日
OpenAIのライバル: Cohereが最高にイケている件
生成AI LLM Cohere
♡ 102

Google Qiita ランキング

すべて 画像 ニュース 動画 ショッピング もっと見る ツール

約 355,000 件 (0.23 秒)

Qiita
https://qiita.com / Qiita
Qiita 週間いいね数ランキング【自動更新】
2020/12/23 — いいね数ランキング: 39 位: 恋がしたい系エンジニア (24)、ブログを作る #恋
愛: 38 位: Manimを使ったアニメーション表現: 37 位: [CSS] mask-image ...

Qiita
https://qiita.com / Qiita
Qiitaのいろいろランキング2022
2024/01/19 — ユーザーごとの、2022年の間で増加したフォロワー数のランキングです。な
お、Qiita公式アカウントは除外しています。ランク、ユーザー名、フォロワー...

Qiita
https://qiita.com / Qiita
Qiitaのいろいろランキング2023
2024/01/19 — 2023年のユーザーの皆さまの活動を、色々なカテゴリーでランキングにしまし
た！今年も公式から提供していきます。ぜひ過去のランキングもご覧ください。

Qiita
https://qiita.com / JavaScript
【保存版】Qiita歴代いいね数ランキング100【自動更新】
2023/03/01 — 【保存版】Qiita歴代いいね数ランキング100【自動更新】 - [1位] Markdown記
法 チートシート (15119いいね) - [2位] プログラミングでよく使う英単語の ...

Qiita
https://qiita.com / Qiita
Qiita週間ストック数ランキング【自動更新】
2018/10/14 — Qiita週間ストック数ランキング【自動更新】 - 1位 たった2つのステップを意識
するだけで書けない単体テストがほぼなくなる - 2位 RAGの実装戦略まとめ

GitHubスター数

- GitHubのスター数で人気記事をチェック

The screenshot shows a browser window displaying GitHub search results. The search query is `stars:>1&s=stars&type=Repositories`. The results are sorted by 'Most stars' and show 8M results. The top results are:

- EbookFoundation/free-programming-books**: Freely available programming books. 316k stars, updated 18 hours ago. Topics: education, list, books, resource, hacktoberfest.
- sindresorhus/awesome**: Awesome lists about all kinds of interesting topics. 295k stars, updated 4 days ago. Topics: lists, awesome, unicorns, resources, awesome-list.
- public-apis/public-apis**: A collective list of free APIs. 287k stars, updated 3 days ago. Topics: api, lists, open-source, list, development.
- public-apis/public-apis**: A collective list of free APIs. 280k stars, updated 9 days ago. Topics: computer-science, algorithm, algorithms, interview, data-structures.

The left sidebar shows filters for 'Filter by' (Code: 1.5k, Repositories: 8M, Issues: 0, Pull requests: 0, Discussions: 0, Users: 0) and 'Languages' (JavaScript, Python, Java, C++, HTML, TypeScript, C, Jupyter Notebook).

<https://github.com/search?q=stars:%3E1&s=stars&type=Repositories>

colissさんの記事

- CSSなどに関する最新技術情報を発信

<https://coliss.com/>



The screenshot shows the coliss.com website with a navigation bar at the top containing 'Top', 'Category', 'Contact', and 'About'. A search bar is located on the right. The main content area features a list of articles, each with a title, a post date, a brief description, and a link to read more.

coliss

Top Category Contact About

サイト構築 -CSS

知っておくと便利なCSSの単位: コンテナクエリに基づく相対単位 (cqw, cqh, cqi, cqb, cqmin, cqmaxなど)の便利な使い方を解説

Post on:2024年4月4日

CSSでよく使用する単位といえば、px, r(em)あたりでしょうか。CSSはそれら以外にもたくさんの単位が使用できます。第3弾となる今回はコンテナクエリに基づく単位 (cqw, cqh, cqi, cqb, cqmin)

[...記事の続きを読む](#)

そろそろSassとはさよなら、Sassの便利な機能がネイティブのCSSでどんどん使用できるようになっています

Post on:2024年4月2日

2024年になり、CSSは急速な進化を遂げています。中でもワークフローにも影響を大きく与えるのが、Sassに代わる新機能の数々です。ネイティブのCSSで変数を扱えるようになり、ネストや:is()疑似クラスや:has()

[...記事の続きを読む](#)

知っておくと便利なCSSの単位: ビューポートに基づく相対単位 (vw, vh, lvh, svh, dvh, vb, vi など)の便利な使い方を解説

Post on:2024年3月28日

CSSでよく使用する単位といえば、px, r(em)あたりでしょうか。CSSはそれら以外にもたくさんの単位が使用できます。前はフォントに基づく相対単位でしたが、今回はビューポートに基づく相対単位 (vw, vh, lv

[...記事の続きを読む](#)

CSSの知っておくと便利な実装テクニック! 相対カラー構文を使用して、1つのカラーに対して濃く

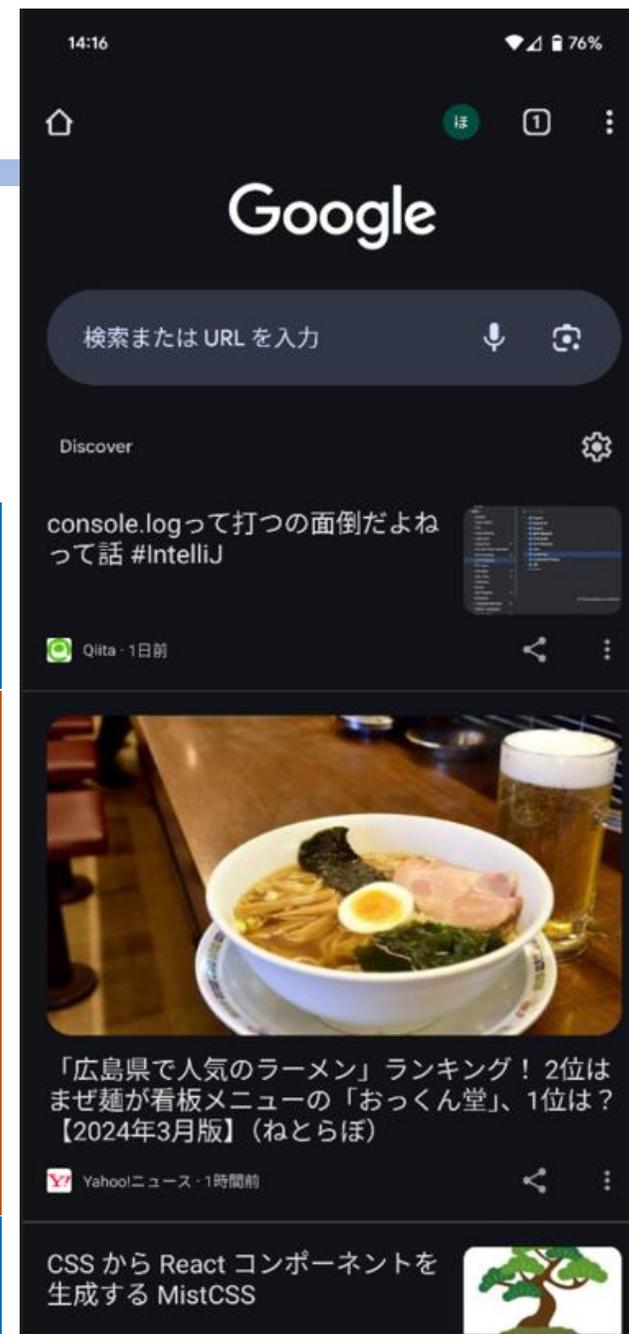
Google Discover

- スマホ版Chromeが表示する「おすすめの記事」
- 閲覧者の指向に合わせて記事をチョイス
- 悔しいけど、読みたくなる記事が多い

JavaScript関連

食べ歩き関連

React関連



人・コミュニケーション

- 人と話してて、興味をもったもの
- 最近ネタに困っているなので、書いてほしいものがあればメールください。

執筆環境

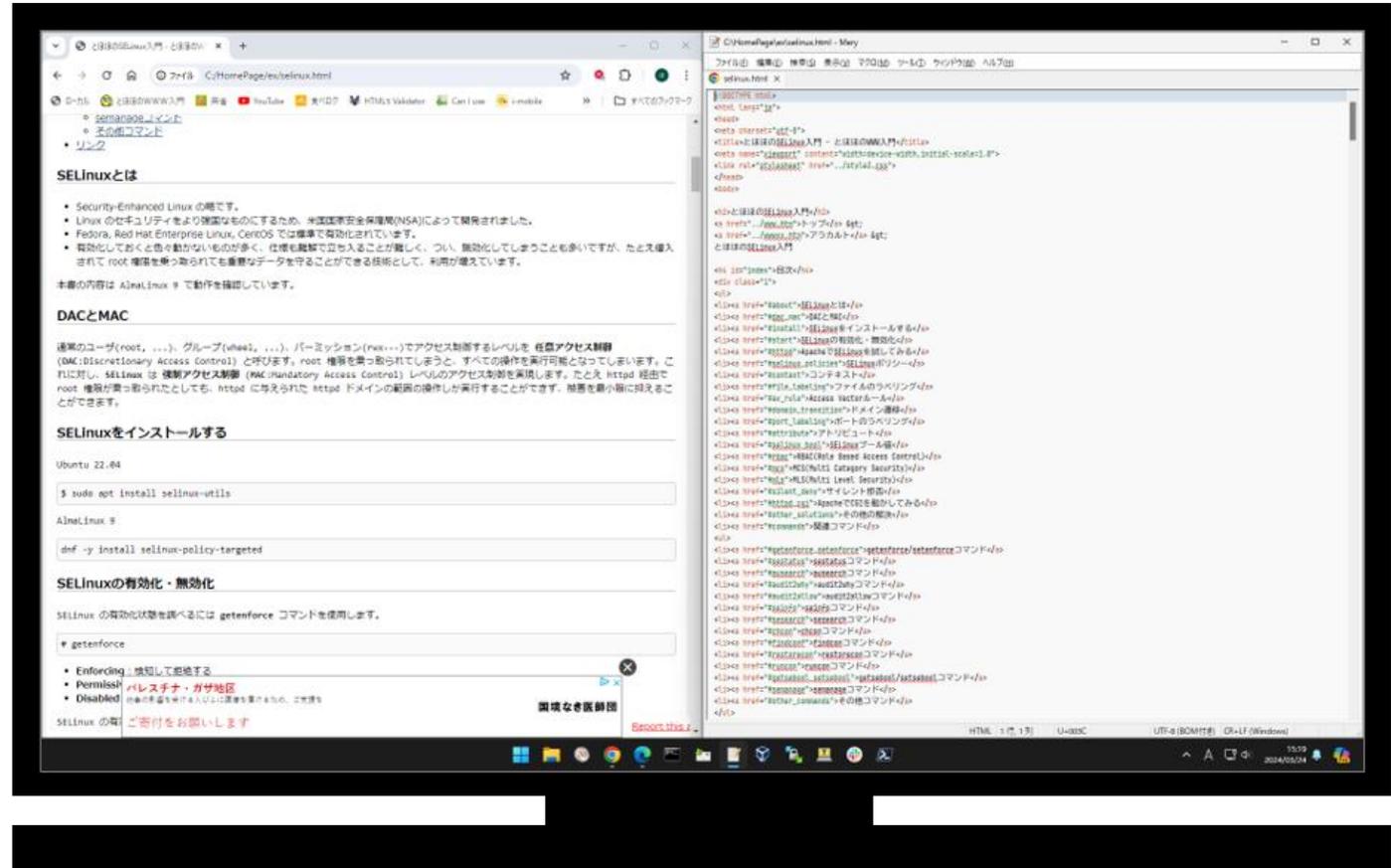
執筆環境

- PC
 - ドスパラ THIRDWAVE
 - Core i5-10400 2.90GHz
 - メモリ16GB
 - Windows 11 Home
 - VirtualBox or WSL + Linux
- ディスプレイ
 - SyncMaster S23B300



ディスプレイの中

- 基本的に全画面か左右分割
- 左にブラウザ（Chrome）
- 右にテキストエディタ
 - Mery（まれに Vim）
 - JavaScript でマクロを組めるのが便利



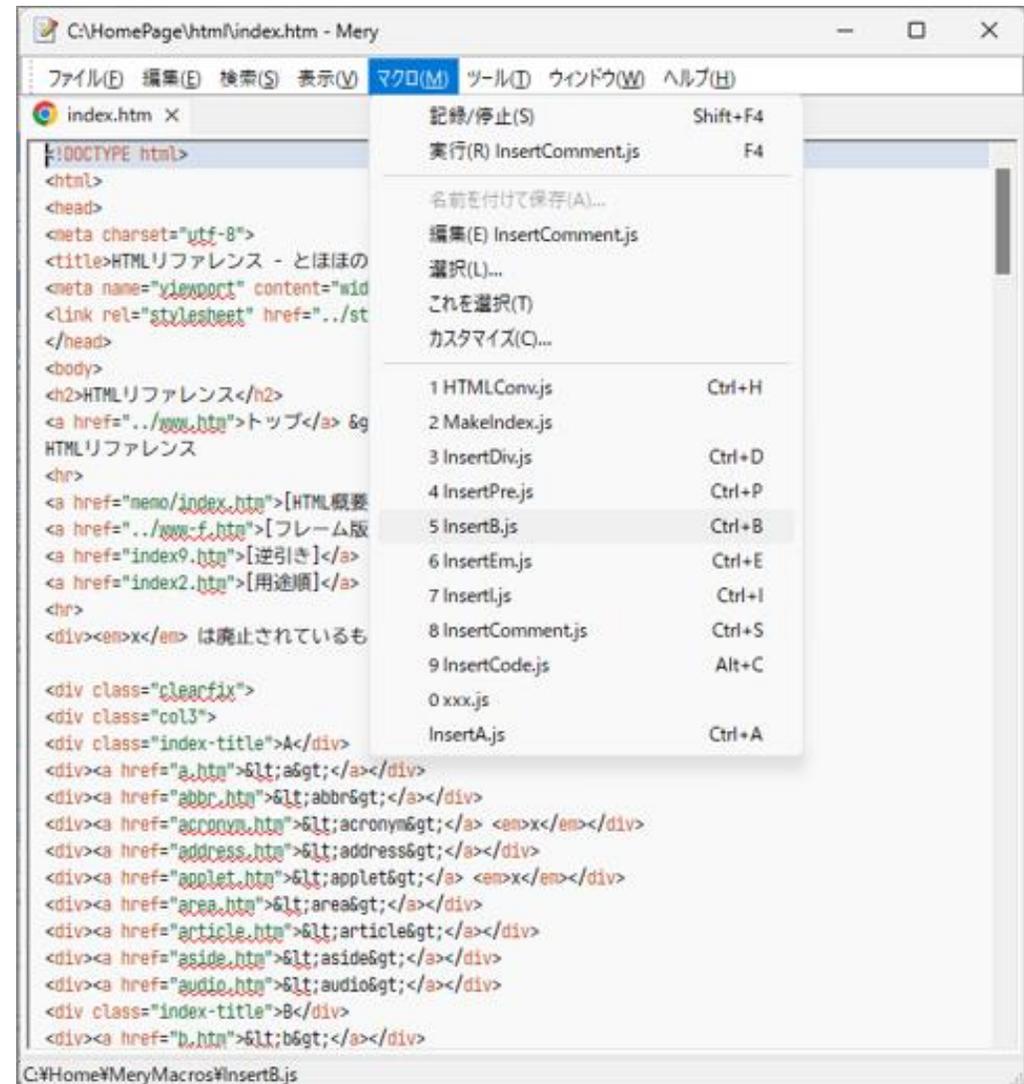
テキストエディタ (Mery)

- JSでマクロ作ってショートカットに登録
- 選択範囲をで囲むマクロ

```
document.selection.Text =  
'<b>' + document.selection.Text + '</b>'
```

- 選択範囲をHTMLエスケープするマクロ

```
document.selection.Text = document.selection.Text  
.replace(/&/g, "&amp;")  
.replace(/</g, "&lt;")  
.replace(/>/g, "&gt;");
```



テキストエディタ (Vim)

- 慣れるのは面倒だけど、慣れたら早い
- カーソルキーに手を伸ばさなくてよい
 - 3行下に移動

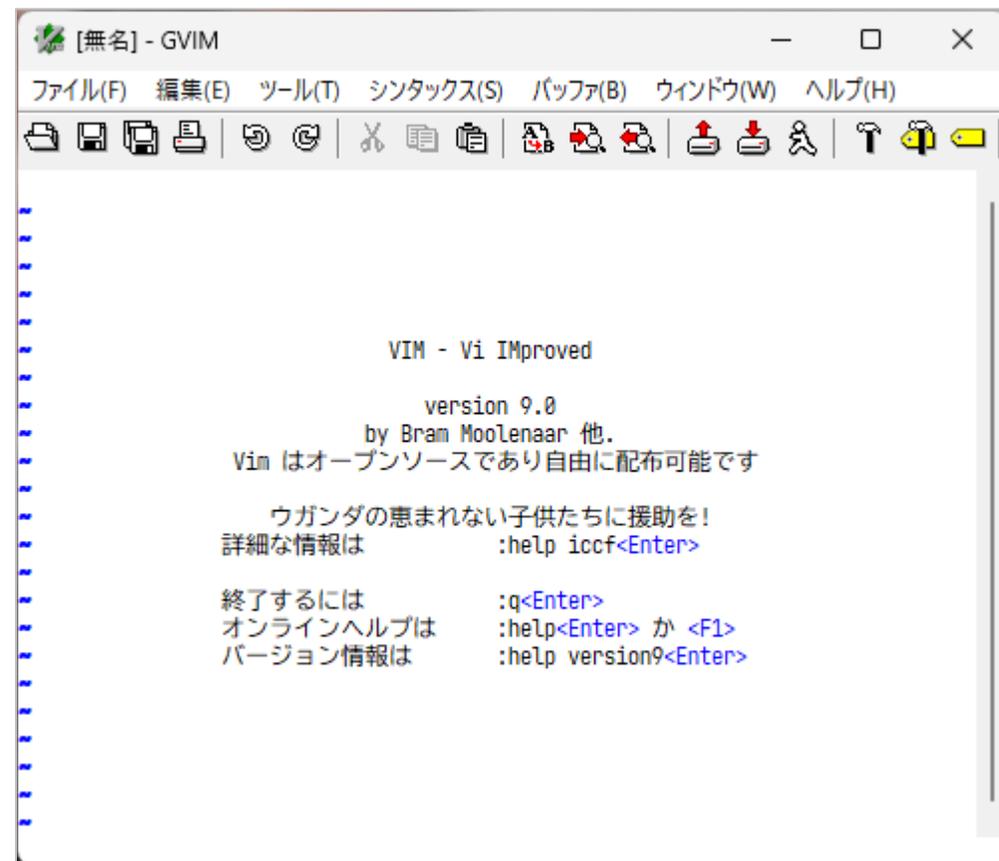
```
jjj
```

- 3行分コピペ

```
3yykp
```

- 3行の先頭にを挿入

```
I<li>ESCj.j.
```



HTMLのインデントはちょっと特殊

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title>とほほのXXX入門 - とほほのWWW入門</title>
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<link rel="stylesheet" href="../style3.css">
</head>
<body>

<h2>とほほのXXX入門</h2>

<h4 id="about">XXXとは</h4>
<ul>
<li>AAA
  <ul>
    <li>AAA
  </ul>
</li>
</ul>
```

あまりインデントしない

あまり階層は深くしない

インデントをつけるのはリストとテーブル程度

** と は同じインデント**

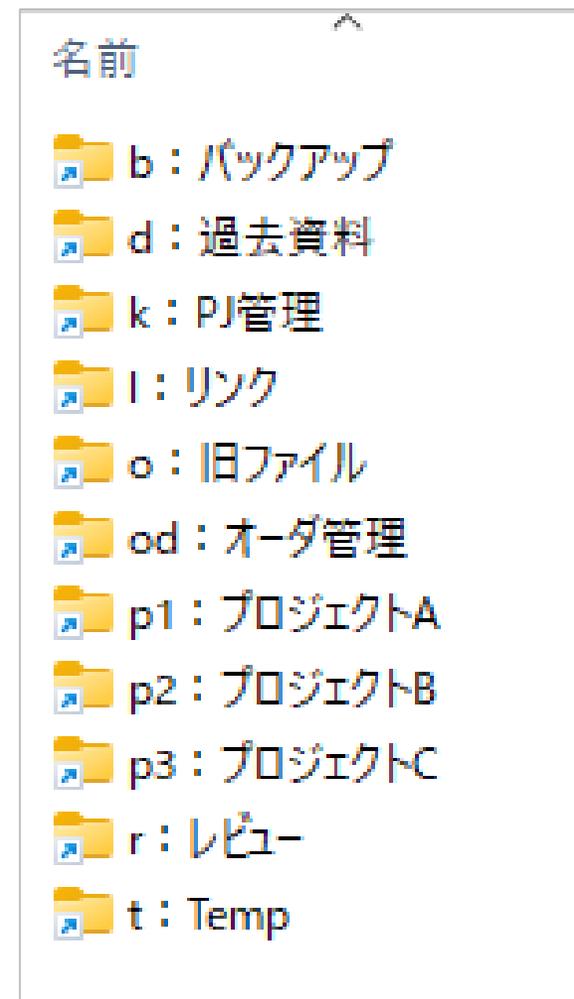
ショートカット無茶苦茶利用します

• 編集作業開始

- [Alt]+[Ctrl]+[C] Chrome起動
- [Alt]+[Ctrl]+[L] エディタ起動
- [Win]+[→] エディタを右寄せ
- [Ctrl]+[O] ファイルを開く...

• フォルダにアクセス

- [Alt]+[Ctrl]+[P] エクスプローラ起動
- [P] [1] [Enter] で プロジェクトA に移動



マウスは極力使わない

- ウィンドウ切り替えは **Alt+Tab**
- ショートカットを多用
 - 「とほほのショートカット入門」参照
 - <https://www.tohoho-web.com/ex/short-cut.html>
- ショートカットの例
 - Chrome起動 : **Alt+Ctrl+C**
 - Mery起動 : **Alt+Ctrl+L**
 - Windowを閉じる : **Alt+F4**
- 昔 (X-Windowの頃) は、新人さんには、マウスをディスプレイの上に置くよう指導してました
 - 基本的にショートカットを使用。どうしても使いたいときだけマウス



執筆時の参考サイト

- 比較的本家サイトを見ることが多い
- 本家以外のサイトは...
 - 概略を知るにはよい
 - 記事の内容が古かったりする
 - 間違いがあつたりする

英語は...

- 英語はそれほど得意ではない
- 最近では自動翻訳の精度もあがってきた
- 単語にマウスカーソル合わせるとその単語だけ翻訳してくれるプラグインも便利



理解するときは目次をコピーして塗りつぶしていく

- 自分がどこまで理解できたかを **見える化** する
- まずは、仕様書の目次をコピーする
- 先頭に ○ をつける
- 理解できたら ● にする
- 不明点が残れば○行を増やす
- すべて ● になったら完成

- 言語リファレンス
 - 基本的な構文
 - 型
 - 変数
 - 定数
 - 式
 - 演算子
- 制御文
- ○yieldステートメント
- 関数
- クラスとオブジェクト
- 名前空間
- 列挙型(Enum)
- エラー
- 例外(exceptions)
- ファイバー
- ジェネレータ
- アトリビュート
- リファレンスの説明
- 定義済みの変数
- 定義済みの例外
- 定義済みのインターフェイスとクラス
- 定義済みのアトリビュート

○○入門ページの基本的スタイル

もし~ならば(if, else, elif)

if は「もし」を意味します。式が真であれば、インデントされたブロックを実行します。下記の例では、num の値が 10より大きければ BIG を3回表示します。

Python

```
if num > 10:  
    print "BIG"  
    print "BIG"  
    print "BIG"
```

else は「さもなくば」を意味します。下記の例では、num が 10より大きければ BIG を、さもなくば SMALL を表示します。

Python

```
if num > 10:  
    print "BIG"  
else:  
    print "SMALL"
```

elif は「さもなくばもし」を意味します。下記の例では、num が 10より大きければ BIG を、10と等しければ NORMAL を、さもなくば SMALL を表示します。

Python

```
if num > 10:  
    print "BIG"  
elif num == 10:  
    print "NORMAL"  
else:  
    print "SMALL"
```

← 2~3行の説明と

← サンプル

← 2~3行の説明と

← サンプル

← 2~3行の説明と

← サンプル

…の繰り返し

これが一番理解しやすいんじゃないかなと思ってる

文章は極限まで短く

- 文章はできるだけ **2~3行** に収まるように心がけてます
- ただし、情報量は落さず、情報密度を上げる
- 長い文章は書きたくない、読みたくない
- 博士と女の子が会話するタイプは、ニガテ...

霊夢と魔理沙はすきだけど...

サンプルも極限まで短く

- サンプルコードも極限まで短くなるように心がけています
- シンタックスハイライトは使用しない
- 説明上重要な個所のみ **赤太字** で表示

JavaScript

```
function aFunc2(data) {  
  return new Promise(function(callback) {  
    setTimeout(function() {  
      callback(data * 2);  
    }, Math.random() * 1000);  
  });  
}
```

とにかく、短いけど情報量は落さずコンパクトに

- 最近だとプログラミング言語の説明も1言語がほぼ1ページに収まるようになってきました

スクロールバー
このくらいになっちゃいますが...



翻訳しただけで安心しない

- **英語の仕様書を日本語に翻訳しただけでは不十分なことがある**
 - 目標は、仕様の定義を正確に日本語化することではない
 - 仕様や現在の実装や注意点を **理解** してもらうことが大事

説明サイトとリファレンスサイトの両立をめざす

- **説明サイト** と **リファレンスサイト** を両立できたらな。と考えてます
 - 説明サイト：はじめて勉強する人が入門書として読む
 - リファレンスサイト：一度勉強した人が思い出すために参照する
- 難しいですが...

目次も大事

- 各ページにも、ページ内容の **目次** をつけてます
- 何が書いてあるか（書いてないか）がわかる
- 章の親子関係がわかるように**章番号**つけました
- リファレンスとして参照する場合も便利
- 目次の自動生成マクロがとても便利
- **アクセシビリティ**としても重要

👤 とほほのCSSトランジション入門

[トップ](#) / [CSSトランジション](#)

目次

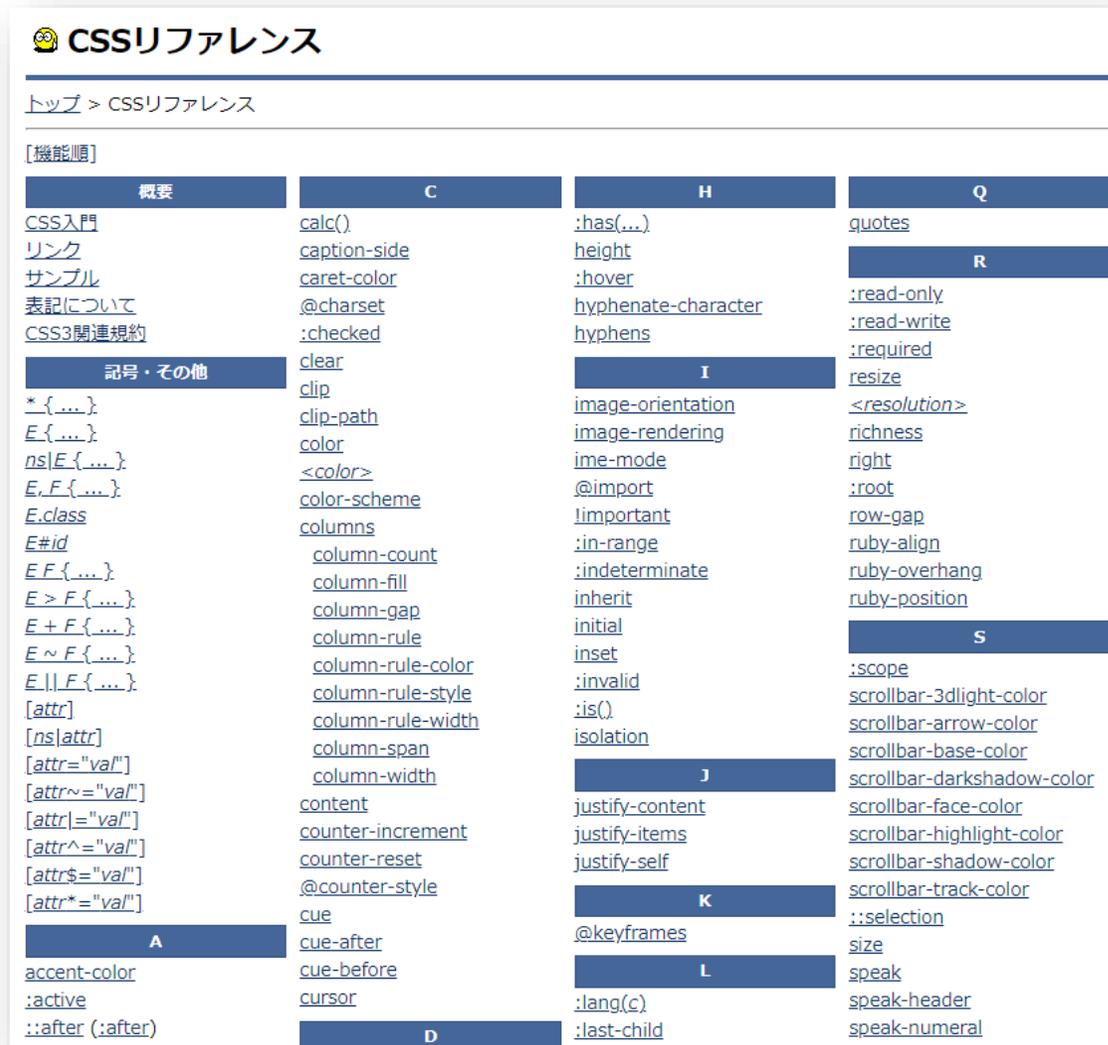
1. CSSトランジションとは
 - 1.1 トランジションとアニメーションの違い
2. 簡単な使用例(transition)
3. トランジション可能なプロパティ
4. トランジションのトリガー
 - 4.1 マウスを乗せた時(:hover)
 - 4.2 フォーカスが当たった時(:focus)
 - 4.3 ラジオボタンがチェックされた時(:checked)
 - 4.4 クラスが追加・削除された時
 - 4.5 JavaScript不要の任意タイミング
5. 詳細設定
 - 5.1 時間(transition-duration)
 - 5.2 開始遅延時間(transition-delay)
 - 5.3 プロパティ指定(transition-property)
 - 5.4 タイミング関数(transition-timing-function)
6. 離散値に対するトランジション(transition-behavior, @starting-style, overlay)
 - 6.1 background-image などの離散値に適用する例
 - 6.2 visibility: hidden に適用する例
 - 6.3 display: none に適用する例
 - 6.4 popover に適用する例
 - 6.5 <dialog>に適用する例
7. auto に対するトランジション
8. ページ間トランジション(@view-transition, view-transition-name)

1. CSSトランジションとは

- 英語の transition は 遷移 という意味を持ちます。
- マウスを乗せた(:hover)、フォーカスが当たった(:focus) などの状態変化が発生した際にアニメーション

索引も大事

- ページ数の多いものは極力 **索引** もつけるようにしています
- いまはまだ自動化できていないので、メンテは結構面倒
- 暇見つけて自動化します...



🤖 CSSリファレンス

トップ > CSSリファレンス

[機能順]

概要	C	H	Q
CSS入門	calc()	:has(...)	quotes
リンク	caption-side	height	
サンプル	caret-color	:hover	R
表記について	@charset	hyphenate-character	:read-only
CSS3関連規約	:checked	hyphens	:read-write
	clear		:required
記号・その他	clip	I	resize
*{...}	clip-path	image-orientation	<resolution>
E{...}	color	image-rendering	richness
ns E{...}	<color>	ime-mode	right
E,F{...}	color-scheme	@import	:root
E.class	columns	!important	row-gap
E#id	column-count	:in-range	ruby-align
E F{...}	column-fill	:indeterminate	ruby-overhang
E > F{...}	column-gap	inherit	ruby-position
E + F{...}	column-rule	initial	S
E ~ F{...}	column-rule-color	inset	:scope
E F{...}	column-rule-style	:invalid	scrollbar-3dlight-color
[attr]	column-rule-width	:is()	scrollbar-arrow-color
[ns attr]	column-span	isolation	scrollbar-base-color
[attr="va"]	column-width	J	scrollbar-darkshadow-color
[attr~="va"]	content	justify-content	scrollbar-face-color
[attr = "va"]	counter-increment	justify-items	scrollbar-highlight-color
[attr^="va"]	counter-reset	justify-self	scrollbar-shadow-color
[attr\$="va"]	@counter-style	K	scrollbar-track-color
[attr*="va"]	cue	@keyframes	::selection
A	cue-after	L	size
accent-color	cue-before		speak
:active	cursor	:lang(c)	speak-header
::after (:after)	D	:last-child	speak-numeral

理解してから記事を書くのではなく・・・

- 覚えたことをかたっぱしから記事にしていく
- 覚えるだけではなく、説明できるってことが深い理解につながる
- 「習うより慣れろ」よりも「習うより教えろ」

インターネットとの出会い（1988年）

- インターネット（TCP/IP）グループに配属される...

今度、社内講習があるから
大丈夫だよ

という訳で、その講師よろしくね

講師やるとなったら、全力で覚えるっしょ
よろしくー

インターネット（TCP/IP）なんて、
全然わからないんですが...

ほっ

えw？

えw？ えw？

校正は娘が...

- 誤記やタイポも多いのですが、記事をアップして2～3日後くらいに、娘が「とうちゃん、ここ、間違っとるよ～」とか教えてくれます。

毎週日曜日に何かは更新

- サボってた時期もありますが、最近は **毎週日曜日** にはなんらかの記事を更新

👤 主な更新履歴

[トップ](#) > 主な更新履歴

過去の履歴

[2023年](#) / [2022年](#) / [2021年](#) / [2020年](#) / [2019年](#) / [2018年](#) / [2017年](#) / [2016年](#) / [2015年](#) / [2014年](#) / [2013年](#) / [2012年](#) / [2011年](#) / [2008年](#) / [2005年](#) / [2004年](#) / [2003年](#) / [2002年](#) / [2001年](#) / [2000年](#) / [1999年](#) / [1998年](#) / [1997年](#) / [1996年](#)

更新履歴

2024年8月4日(日)

- 「[とほほの韓国語入門](#)」を少し改訂しました。

2024年7月28日(日)

- 「[とほほのDevTools入門](#)」を追加しました。

2024年7月21日(日)

- [Python](#) の [フォーマット文字列](#) を加筆しました。
- [暗号化入門](#) を少し加筆・整理しました。

2024年7月14日(日)

- 「[HTMLリファレンス](#)」で下記を追加しました。
 - [body](#) 要素の [onpageswap](#), [onpagereveal](#) 属性
 - [details](#) 要素の [name](#) 属性
 - [script](#) 要素の [fetchpriority](#) 属性
 - [template](#) 要素の [shadowrootmode](#), [shadowrootdelegatesfocus](#), [shadowrootclonable](#), [shadowrootserializable](#) 属性
- 「[Shadow DOM](#)」を大幅加筆しました。

2024年7月7日(日)

- 「[とほほのカーブ入門](#)」を追加しました。

よく読まれている記事

■ カテゴリ別

	記事	割合
1	HTML	10.68%
2	CSS	7.09%
3	JavaScript	6.61%
4	珍しい苗字	4.88%
5	Python	4.34%
6	トップページ	3.37%
7	Java	3.30%
8	Bootstrap	2.46%
9	Perl	2.33%
10	バッチ(BAT)	2.06%
11	PowerShell	1.68%
12	Bootstrap 5	1.51%
13	正規表現	1.14%
14	拡張子	1.07%
15	Perl5	0.98%

■ ページ別

	記事	割合
1	珍しい苗字	4.88%
2	トップページ	3.37%
3	バッチ(BAT)	2.06%
4	PowerShell	1.68%
5	Python/演算子	1.32%
6	正規表現	1.14%
7	拡張子	1.07%
8	文字コード	0.97%
9	AWK	0.95%
10	jq	0.85%
11	Next.js	0.82%
12	HTML/トップページ	0.78%
13	OpenSSL	0.77%
14	JSON	0.76%
15	シェル(Bash)	0.74%

「とほほの…」で一番アクセスの多いページ

■ 番外編

[点字 \(1\)](#)

[映画\(2021年\) \(1\)](#)

[珍しい苗字 \(1\)](#)

[陶磁器 \(1\)](#)

[仏教 \(1\)](#)

[韓国語 \(1\)](#)

[中国語 \(1\)](#)

[洋楽 \(1\)](#)

[所得税 \(1\)](#)

トップページを
抑えて堂々の1位

🤪 珍しい苗字

[トップ](#) > [珍しい苗字](#)

珍しい苗字

Web関連技術もネタが尽きてきたので、今日は珍しい苗字についてちょっと調べてみました。読み方がある場合があります。人数や情報は「[苗字由来net](#)」さんを参考にさせていただきました。

風情のある名前

小鳥遊	たかなし	鷹が居ないと小鳥が遊べるので「たかなし」さん。全国で約30人ほど。
月見里	やまなし	山が無いと月がよく見えるので「やまなし」さん。全国で約270人ほど。
春夏秋冬	あきなし	秋が無いので「あきなし」さん。全国で20人ほど。
臥龍岡	ながおか	龍が臥せている状態が長い丘に見えるので「ながおか」さん。全国で約...

季節感のある名前

四月一日	わたぬき	春になると着物の綿を抜くので「わたぬき」さん。
五月七日	つゆり	梅雨にはいるので「つゆり」さん。
栗落花	つゆり	こちらも「つゆり」さん。梅雨にはいるところに栗の花が落ちるから。
六月一日	うりわり	瓜が割れる頃なので「うりわり」さん。実在するかどうかは調査中。
八月一日	はずみ	八月になると稲の穂を摘むので「はずみ」さん。
八月朔日	はずみ	「朔日」は「一日」と同意。「はっさく」という読み方もあるそうです。

数字にまつわる名前

一	にのまえ	数字の一は二の前なので「にのまえ」さん。「SPEC」のドラマにも出て
二	したなが	下の線が長いので「したなが」さん。実在するかどうかは不明。
四	あずま	「東屋」のことを中国伝来語で「四阿」と書くことから「四」を「あずま
九	いちじく	ひとつの字で九なので「いちじく」さん。

長く続けてこられた 秘訣とは

長く続けられてきた秘訣

- ・「とほほのWWW入門」を29年間も続けることができた秘訣とは・・・？

好きだから

ゴルフ・パチンコ・釣りなどと一緒

「継続は力なり」 < 「好きこそものの上手なれ」

好きにはなれないときは、一度、自分を騙して好きになってみる。

挫折しそうなときは...

- 2年ぐらい更新していない時もありました。

休む時は休むと割り切ってみる

再開のきっかけ：

新しい情報 **期待してます** などの言葉

周りからの言葉によってモチベーションが復活

部下に仕事を好きになってもらったり、やる気を復活させたりも
上司の大事な仕事

**これから何が
ひろまっていくのだろうか
(未来)**

これからなにが広まっていく？

• Web関連

- 1990年代 : HTML/CGI
- 2000年代 : Server-Side Script (CakePHP, Django, Laravel, ...)
- 2010年代 : Single-Page Application (React, Vue, Angular)
- 2020年代 : その他 (SSR, HTMX, ???)
- :

サーバーサイドからクライアントサイドに移行して、
また、すこし、サーバーサイドに戻ってきた・・・？

最近ちょっと面白いなと思っているもの

- Next.js

- React なので Client-Side メインなのかと思っていたら Server-Side メイン...
- JSの初期ロードが遅いなどの問題を解決

- HTMX

- サーバ機能を HTTP で呼び出して得た HTML を張り付け

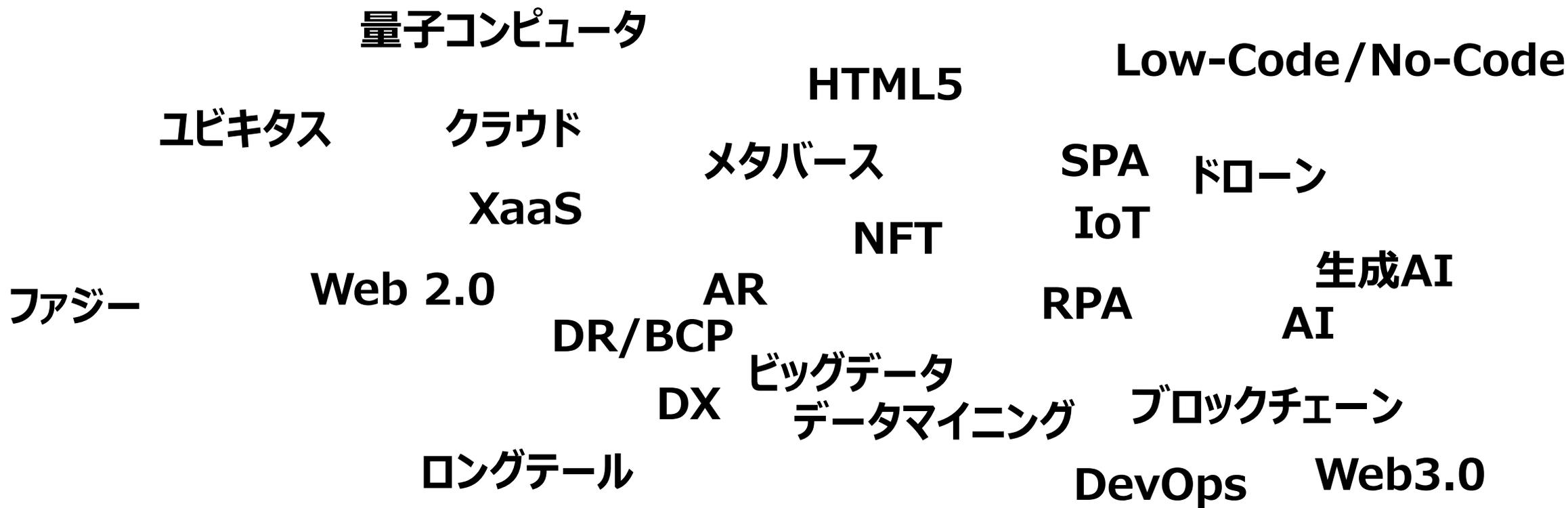
```
<div id="out">OUT</div>
```

```
<button hx-get="/hello" hx-target="#out">Click me</button>
```

- 他にも変わった視点のものが出てくるかも

良い技術と金になる技術は違ったりもする??

- ファッション業界とIT業界は実は似ていたりする?
 - その年々の **流行** があり、流行に乗ると売りやすい、金にしやすい?

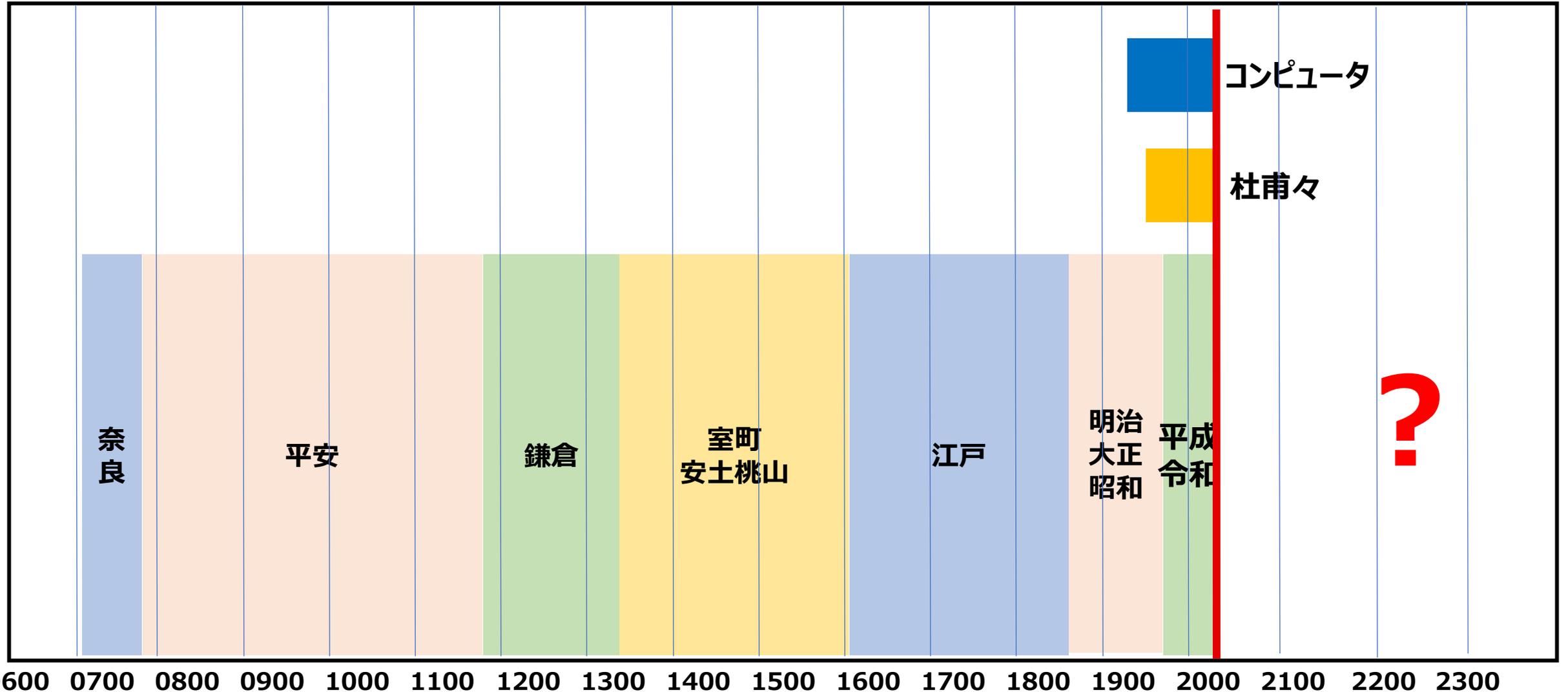


今後注目のキーワード？

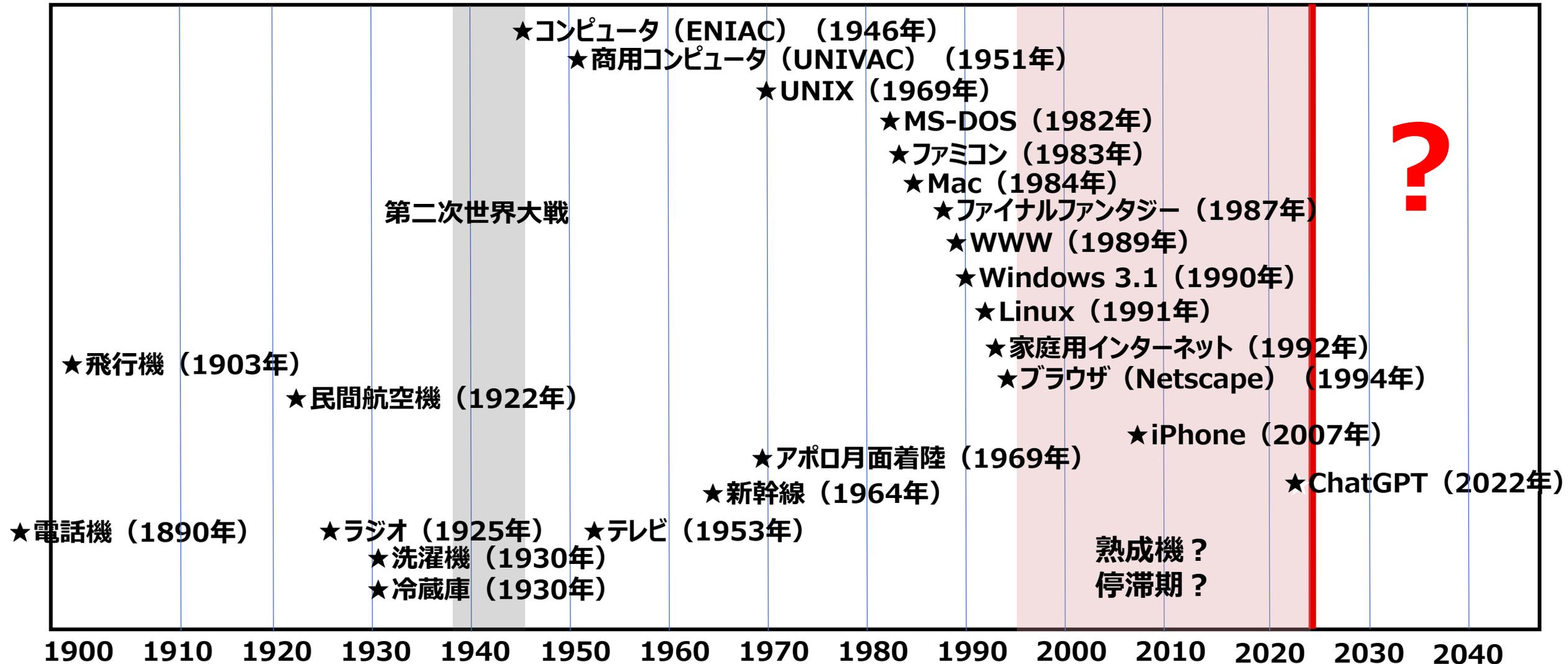
・ガートナーさんの言うには

- **AI TRiSM** ... AIの信頼性、リスク、セキュリティの管理
- **CTEM（持続的脅威露出管理）** ... 持続的なリスク管理
- **持続可能テクノロジー** ... 長期生態系バランスを見据えたテクノロジー
- **プラットフォームエンジニアリング** ... 開発環境の開発・維持
- **AI拡張型開発** ... AIを用いた開発
- **インダストリ・クラウド・プラットフォーム** ... 業界別クラウドプラットフォーム
- **インテリジェント・アプリケーション** ... アプリにAIを組み込んで
- **生成AIの民主化** ... 生成AIを誰でも使える様に
- **拡張コネクテッド・ワークフォース** ... 従業員から得られる価値の最適化
- **マシン・カスタマー** ... AIによる売買契約

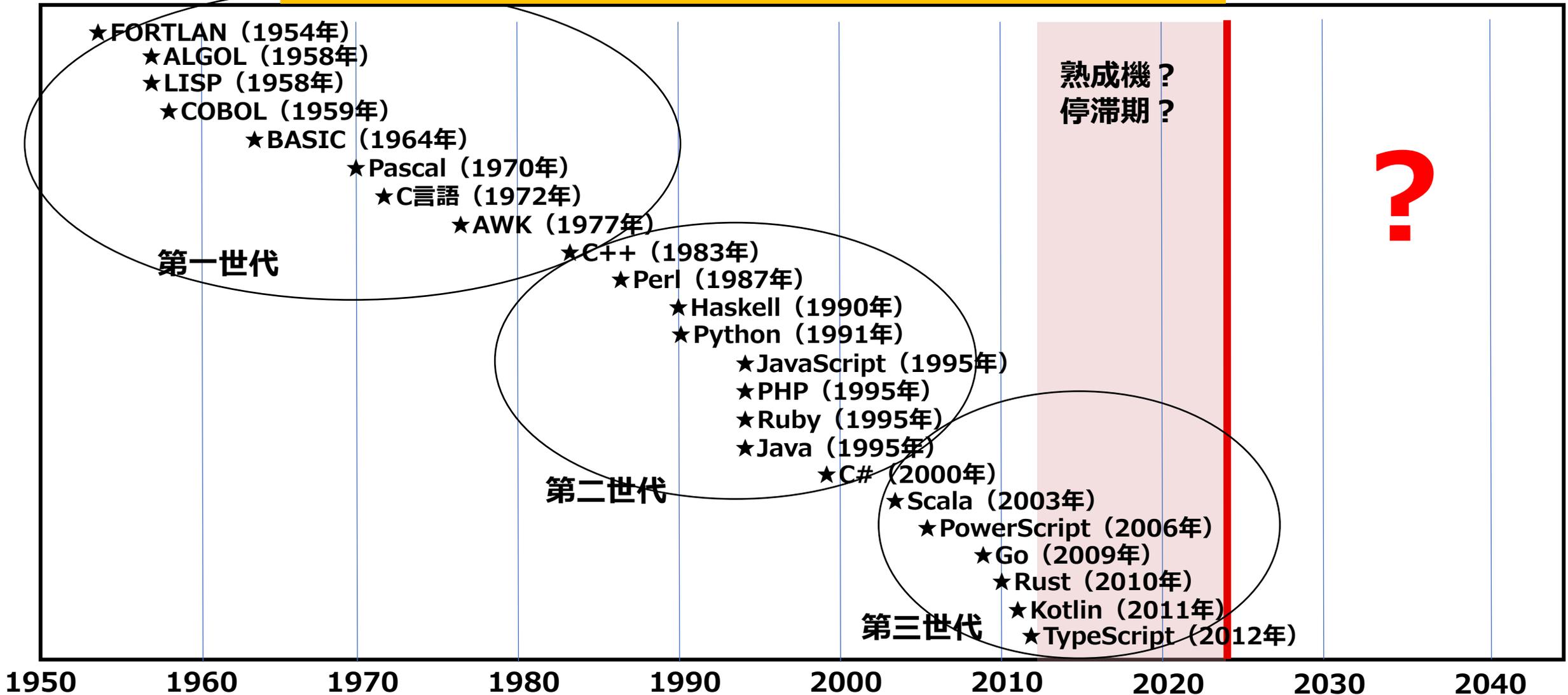
今後、どんな発展があるのだろうか



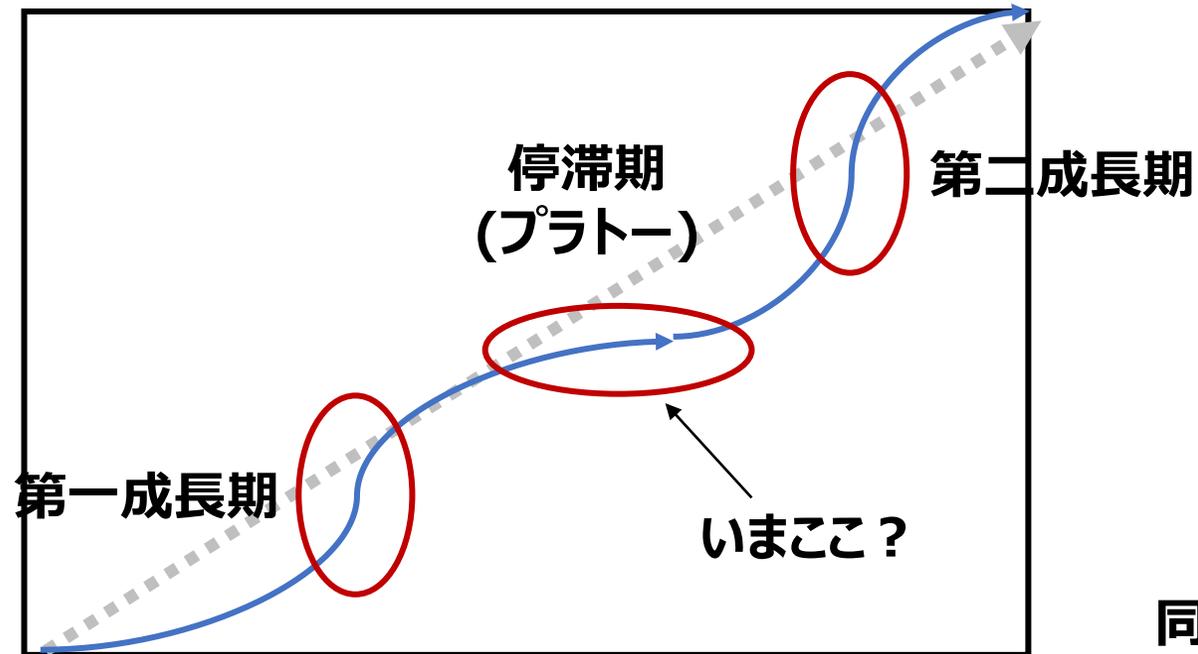
コンピュータの歴史



プログラミング言語の歴史



成長曲線・プラトー



学習の成長も
文明の成長も
同じような曲線になるのかも

ITルネッサンス?
IT復興?

プラトールを超えて

そろそろ、停滞期を終えて、
新しい進化を**見る**ことができる
...のかもしれない

もしかすると、停滞期を終えて、
新しい進化を**作る**ことができる
...のかもしれない

趣味のコーナー

番外編

・息抜きコーナー

点字、陶磁器、洋楽、仏教、韓国語、所得税、資産運用などなど



The screenshot shows a website menu with two main sections: 'アラカルト' (Arakato) and '番外編' (Banpaiben). The '番外編' section is highlighted with a red box. Below the menu, there is a copyright notice and a URL.

■ アラカルト	
アラカルト (42)	厳選リンク集 (1)
How To (39)	杜甫々の本 (1)
厳選ツール集 (10)	スペシャルサンクス (1)

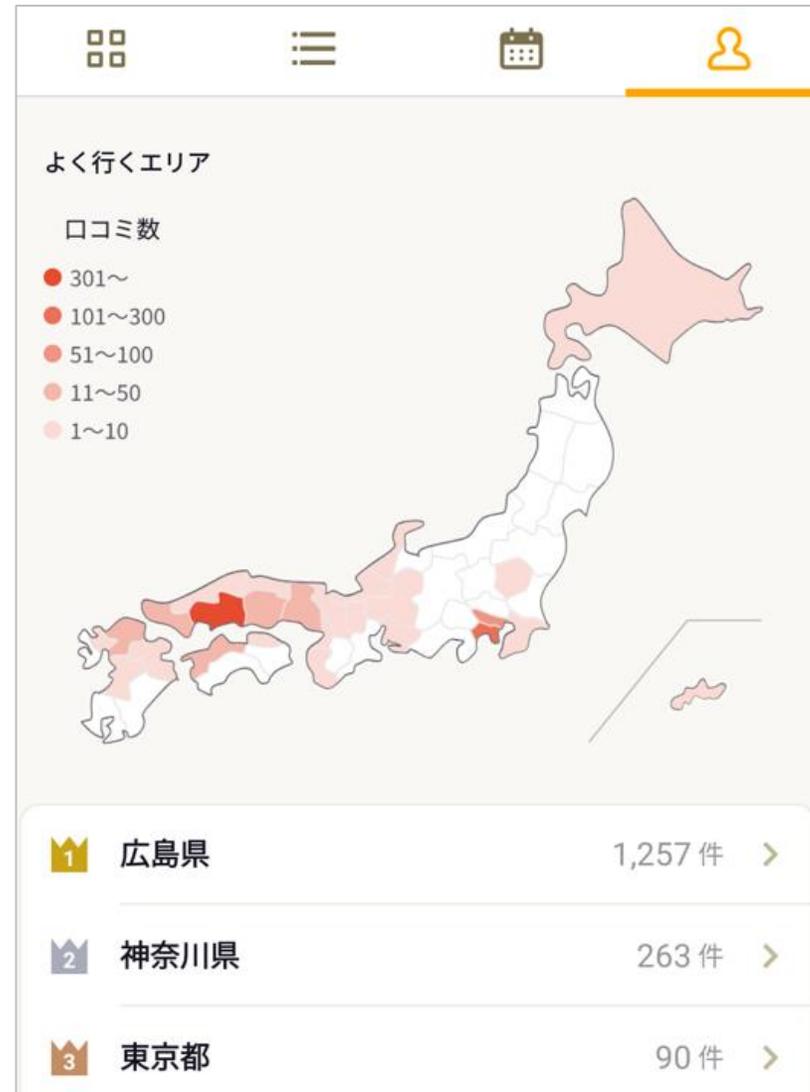
■ 番外編	
点字 (1)	洋楽 (1)
陶磁器 (1)	所得税 (1)
仏教 (1)	資産運用 (1)
韓国語 (1)	とほほの雑記帳 (5)

Copyright (C) 1996-2022 杜甫々 
<https://www.tohoho-web.com/www.htm> 

こちらへんも
息抜きに
見てみてください

食べ歩き（食べログ）

- 食べログ片手に食べ歩き中
- 訪問店舗数：**1,911店舗**
- 食べログ VIP会員



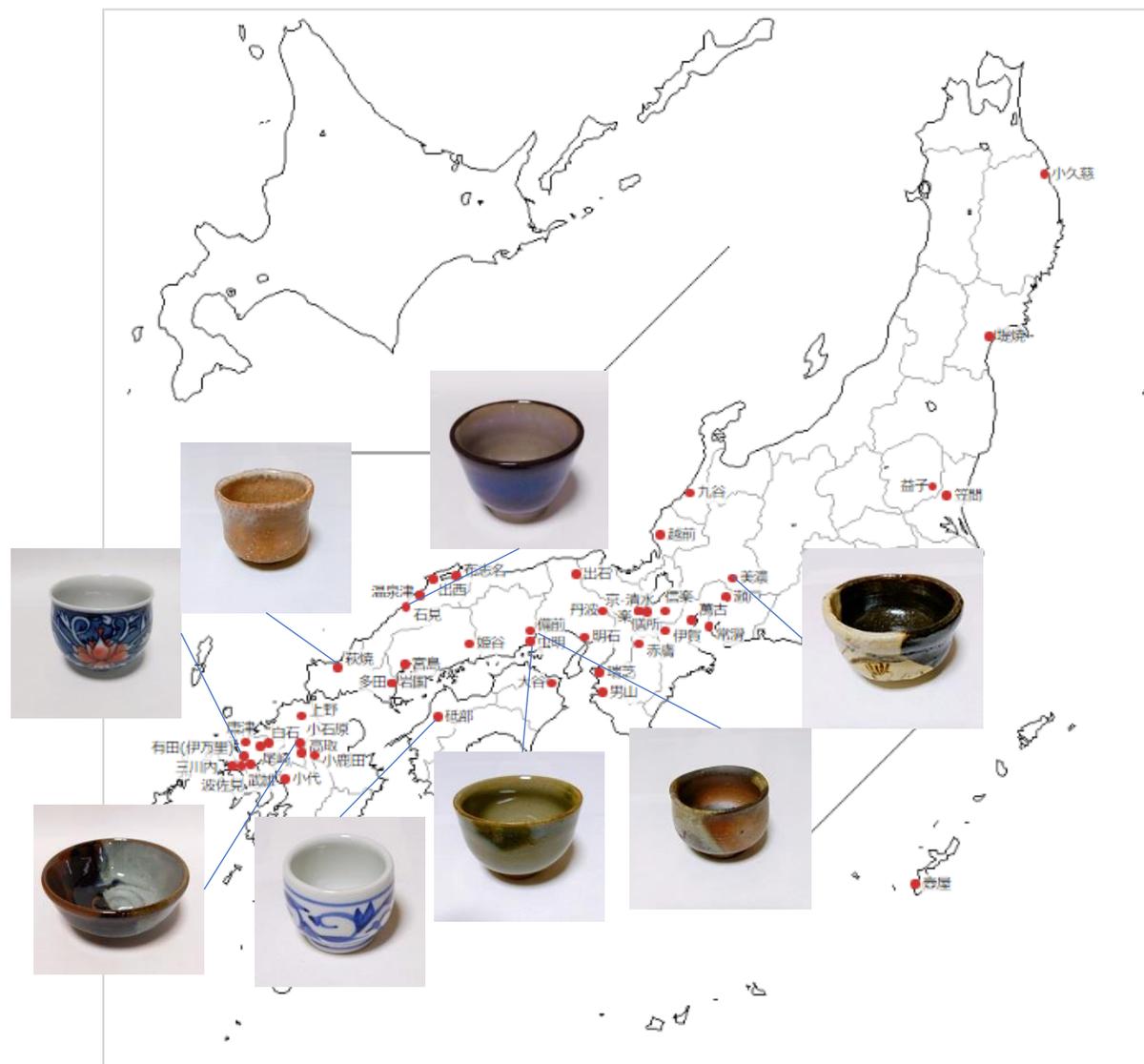
御湯印めぐり

- スーパー銭湯好きです
- 中四国のスーパー銭湯めぐり
- 「御湯印」をコンプリートするとタオルがもらえる
- 今年もやっています

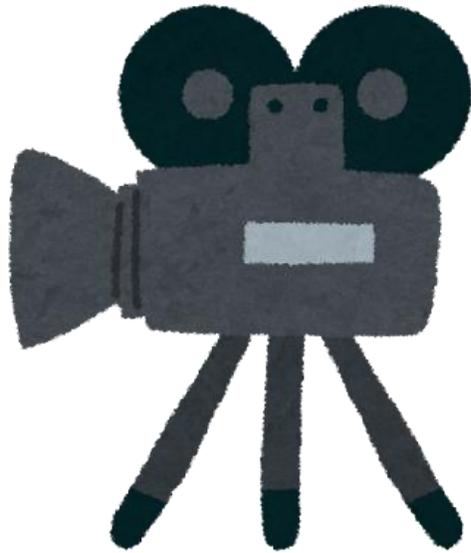


陶磁器

- 最近は陶磁器にはまってました。
- 産地を回って 猪口 を集めてます。



その他の趣味



映画

多い年は
80本/年くらい



カラオケ

娘と11時間...



カープ

6回/年くらい
観戦



チョコザップ
ちょこっと...

とほほの中国語/韓国語/点字入門

Google で
「○○入門」で検索すると
結構トップに表示

とほほの中国語入門

トップ > 中国語

中国語とは

- 中国の言語です。
- 中国は広いので場所が変わると方言も強く中国人同士でも通じません。
- 中国国内の標準語として、北京語に近い「普通話(ブートンボ)

四声(しせい)

中国語を発音する際は「四声(しせい)」が重要です。同じ「まー」変化します。四声を誤ると中国の人には通じません。

第1声	第2声	第3声	第4声
mā 妈 マー お母さん	má 麻 マァ? 麻(アサ)	mǎ 马 マァア? 馬	mà 骂 マァ 罵る
→	↗	↘↗	↘
高い音のまま	低音から高音に上がる	一度下がって上がる	高音から低音に下がる

ピンイン

中国語の発音はピンイン(拼音)で表します。例えば「我爱你(アイニー)」と読みます。ピンインには a á à ã など四音を示すがあります。

とほほの韓国語入門

トップ > とほほの韓国語入門

韓国語とハングル

韓国(한국/ハングッ)で使用される言語が、韓国語(한국말/ハングマル)で、韓国語で使われるハングルには「偉大な」、グルには「文字」という意味があります。

子音・母音・パッチム

子音	母音	子音	母音	子音	母音
パッチム	パッチム	母音			

한 국 요 리

- ハングルの1文字は「子音」、「母音」、「パッチム」から構成されます。
- 子音は、日本の「あかさたな」に相当します。
- 母音は、日本の「あいうえお」に相当します。
- パッチムは、子音や、子音の組み合わせで構成されます。無いこともあります。
- 子音は、パッチムとして使用される時は読み方が異なるものもあります。
- 下の表は日本人が覚えやすいように、韓国で一般的に使用される並びとは変えています。

子音					
ㅇ	ㄱ	ㅋ	ㆁ	ㄴ	ㅇ
	g/k	s	d/t	n	h
ㄷ	ㅌ	ㄷ	ㅌ	ㄷ	ㅌ
r	j	ch	b/p	k	t
ㄱ	ㅋ	ㆁ	ㄴ	ㅇ	
kk	ss	tt	jj	pp	

とほほの点字入門

トップ > とほほの点字入門

ある日ビールを飲んでいると、ビールの缶に下図のような突起がありました。点字です。何て書いてあるのかわかりたくなり、www入門をちょっとお休みして、番外編「点字入門」を作成してみました。このページは出典さえ明記していただければ、コピー、印刷、ばらまき自由ですのでご利用ください。



マス(6つの点)

点字はすべて6個の点で表わされます。それぞれの点を1の点~6の点と呼びます。1の点~6の点を合わせてマスと呼びます。

1 4
2 5
3 6

母音(アイウエオ)

あいうえおは、左上側の3つの点のみを用います。これは丸覚えしましょう。

ア	イ	ウ	エ	オ
●○	●○	●●	●●	○●
○○	●○	○○	●○	●○
○○	○○	○○	○○	○○

子音(アカサタナ)

か、さ、た、な、は、ま、ら行は、右下側の3つの点で子音(K, S, T, N, H, M, R)を表わし、母音(A, I, U, E, O)と組み合わせで使用します。結構簡単ですね。

とほほのグルメ系入門

🍛 とほほのタイ料理入門

トップ > とほほのタイ料理入門

代表的なタイ料理

トムヤムクン



OpenCage, CC BY-SA 2.5, via Wikimedia Commons

トム(茹でる)+ヤム(混ぜる)+クン(海老)。一番有名ですね。海老がメインのスープです。世界三大スープのひとつとされます。

カオ・マン・ガイ



Takeaway, CC BY-SA 3.0, via Wikimedia Commons

カオ(ご飯)+マン(油)+ガイ(鶏)。鶏肉を茹で、その茹で汁で炊いたご飯を添えた料理。本場は軍鶏(シャモ)を使用するそうです。

パット・ガバオ(ガバオ)



Takeaway, CC BY-SA 3.0, via Wikimedia Commons

パット(炒める)+ガバオ(ガバオ)

パットタイ

🍜 とほほの広島ラーメン入門

トップ > とほほの広島ラーメン入門

広島ラーメンとは

広島西部を中心とするラーメンです。最近少し全国的知名度が上がった広島東部の尾道ラーメンとは異なります。茶褐色の醤油とごま油、豚骨、鶏ガラ、野菜などを白濁するまで煮出していますが、意外にあっさりしています。中細ストレートのやわらか麺。具材はチャーシュー、ネギ、モヤシ、シナチクが基本。ラーメンではなく中華そばと呼ぶ店も多いようです。

広島ラーメンの歴史

広島ラーメンの老舗と言えば「しまい」「すずめ」「陽気」の3軒ですが、親戚関係があります。「段原食堂」を経営していた沖裕さん、次男が「しまい」を、奥さんの姉妹の娘、つまりは姪っ子姉妹の妹が「すずめ」を、姉が「陽気」を創業します。「しまい」と「乙丸」も姉妹。「うくいず(流川)」と「うくいず(新天地)」も姉妹。姉妹店というか、奥姉妹が関わる老舗が多い気がします。

上海・段原食堂 (現店)

戦後の混乱の中、満州からの引揚者や中国人が屋台で提供する中華そばに感動して沖裕(おきみのる)さんが屋台「上海」を開業しました。昭和25年(1950年)に屋台が禁止されたため店舗「段原食堂」を開業しました [1]。次男の誠治さんは「しまい」を、妻シゲノさんの姪の関上タツコさんは「すずめ」、タツコさんの姉の津留田マサエさんは「陽気」を開店させるなど **広島ラーメンのガイア** とも言えます。

しまい(舟入→大町) (現店)

Wikipedia では沖裕さんが店主となっています [1] が、沖裕さんの次男沖誠治さんが父(沖裕さん)に習い昭和30年初頭舟入で開業したのが「しまい」です [2]。妻沖ヒデコさんと共に営まれていました。開店時の名前は「姉妹」。姉妹の由来は2説あるようです。 **広島ラーメンの元祖** として続けられました2017年4月16日閉店。

[1] 段原食堂の沖シゲノさんと、「すずめ」の関上タツコさんの母が姉妹だったからという説 [2]

[2] 「しまい(大町)」の沖ヒデコさんと、「しまい(牛田區)」の沖栄子さんが姉妹で開いたからという説 [12]

すずめ(寿寿女)

沖シゲノさんの姉の娘である関上(旧姓:沖)タツコさんと夫の関上卓爾さんが「しまい」の沖誠治さんに習って開業したのが「すずめ」です。開業時は「寿寿女」でした。長男の関上道弘さん、娘婿の山本誠二さんも加わり、「めじろ」、「つぼめ」など小鳥の名前を冠する **小鳥系広島ラーメンの元祖** となりました。2015年4月に閉店しましたが、舟入で山本さんの「寿々女」が後継店としてオープンされています。

タイ料理

🍢 広島のお好み焼き「八昌」

トップ > 広島のお好み焼き「八昌」

目次

広島のお好み焼き屋さんで美味しいと評判なのが「八昌」。でも、青八昌系、赤八昌系、だるま八昌系などいろいろあるようですので、整理してみました。

- 元祖八昌系 (赤八昌系) 元祖八昌(竹屋町)
- 青八昌系 (第一世代) 八昌(美研橋) / 八昌(銀山)
- 青八昌系 (第二世代) たかのぼし八昌 / ロベズ
- 青八昌系 (第三世代) Masaru / 元祖八昌(竹屋町)

🍷 とほほのお酒入門

トップ > とほほのお酒入門

お酒とは

- アルコールを含む飲料です。
- 日本の酒税法では「アルコール分1度以上の飲料」と定められています。
- 製法によって「醸造酒」「蒸留酒」「混成酒」の3種に大別されます。
- 日本の法律では上記3種

🍷 とほほの簡単クッキング

トップ / 簡単クッキング

はじめに

料理も嫌いではないので朝飯、昼飯、酒のつまみとかチャチャと作ったりしてます。短期間ですが妻が働いてた時期もあるのでテレワークの自分が晩飯作ってた時もありました。素人でも簡単に作れる簡単クッキングを少し紹介...

蒸留酒

醸造酒を蒸留(一度気化)して、ウイスキー、ブランデー

発泡性酒類

日本の酒税法による分類

ワイン(wine)

ブドウを発酵させてつくられる

ワインの製法

ワインは製法によって下記に

スティルワイン

スティル(still)は静か、動かない

赤ワイン、白ワイン、ロゼ



カリカリチーズせんべい。クッキングシートにスライスチーズを乗せて1分10秒レンジするだけ。チーズがカリッカリ、サクサクになって美味しいです。ビールのあてにちょっと困ったらこれ。



玉子サラダサンド。最近の朝の定番がこれ。糖質を控えよう朝は10枚切りパンにしています。パンをトーストしている間に玉子1個をフライパンに落として箸で黄身と白身を混ぜる程度に少しだけ混ぜます。弱火で焼いて、固まり切らないように巻いて、火を止めて、あとは余熱でじんわり。パンに乗せてあり物の野菜のつけて、マヨネーズとかケチャップとかお好みで。



玉子マヨトースト。本当は食パンにマヨネーズで防壁を作って卵を割り入れてオープンでしっかり焼くんだけど、玉子が焼けるまで待つと結核時間がかかるしパンも焦げやすくなるので、パンにマヨネーズ塗ってトーストし、40秒間だけレンジした玉子を後乗せします。最後に塩コショウをひとふり。玉子をレンジする時は黄身に2~3箇所穴をあけておくのを忘れないように。



ずばろフレンチトースト。朝は10枚切りパンなので普通のフレンチトーストを作ろうとすると玉子1個はちょっと多いので、玉子の代わりにマヨネーズを使ってみました。パンがひたひたになるくらいに牛乳につけて、片面にマヨネーズ。フライパンでオリーブオイルで両面焼くだけ。



ずばろオムライス。材料は/スタ用のミートソースとご飯と玉子のみ。ミートソースは軽くレンジしておきます。ご飯を少し炒めて、ミートソースを半分加えてもう少し炒めます。玉子2個をぐるぐる回しながらドレス・ド・オムライスっぽく焼いてのつけて、残ったミートソースをぶっかけます。市販ミートソースを使うので無茶苦茶簡単。

広島お好み焼き

お酒

簡単クッキング

とほほの投資・法令・税金系入門

資産運用

とほほの資産運用入門

トップ / とほほの資産運用入門

目次

- 1. はじめに
 - 1.1 用語
 - 1.2 資産運用と投資
 - 1.3 リスクとリターン

とほほの相続入門

トップ / 相続

目次

- 1. はじめに
- 2. 法定相続人
 - 2.1 相続配分の例

とほほの法令入門

トップ / とほほの法令入門

目次

- 1. 法令の分類と優先度
- 2. 憲法
- 3. 条約
- 4. 法律
- 5. 勅令
- 6. 政令・施行令
- 7. 府令・省令・庁令・規則
- 8. 条例・規則・要綱・例規
- 9. 告示・訓令・通達・通知
- 10. 民法と刑法
- 11. 親告罪と非親告罪
- 12. 懲役と禁錮
- 13. 実刑と執行猶予
- 14. 身近な法律違反
- 15. 主な法令一覧
- 16. リンク

資産運用

相続税

相続税

退職

とほほの退職入門

トップ / 退職

定年退職も近づいてきたことから、退職金、年金、雇用保険、健康保険等について整理してみました。

目次

- 1. 退職給付制度
- 2. 退職金制度
 - 2.1 退職金を一時金
 - 2.2 退職金を年金で
 - 2.3 5年ルール・19
- 3. 年金
 - 3.1 年金の3階建て
 - 3.2 公的年金
 - 3.2.1 国民年金(旧版：「とほほの所得税入門」)
 - 3.2.2 厚生年金(旧版：「とほほの所得税入門」)
 - 3.2.3 繰上げ受給企業年金
 - 3.3 確定給付企業年金
 - 3.3.1 確定給付企業年金
 - 3.3.2 企業型確定拠出年金
 - 3.3.3 厚生年金基金
 - 3.4 個人型確定拠出年金
 - 3.4.1 個人型確定拠出年金
 - 3.4.2 国民年金基金
 - 3.5 加給年金
 - 3.6 特別支給の老齢厚生年金
 - 3.7 在職老齢年金

とほほの確定申告入門

トップ / とほほの確定申告入門

確定申告における所得税の計算は年々変わります。ここに記載しているのは執筆時(令和5年(2023年)分)の計算方法のみです。このまま計算するんだという勇気のみ参照し、実際の計算は国税庁のサイトを利用するか、免許を持った税理士さんに相談してください。

目次

- 1. 確定申告とは
- 2. 所得税の計算
 - 2.1 給与所得
 - 2.2 収入
 - 2.3 控除
- 3. 所得税の計算
 - 3.1 収入
 - 3.2 必要経費
 - 3.3 所得
 - 3.4 所得割
 - 3.5 課税
 - 3.6 税額
 - 3.7 税額控除

確定申告

個人事業主

とほほの個人事業主入門

トップ / 個人事業主

目次

- 1. はじめに
- 2. 働き方の呼称
- 3. 開業届
- 4. 青色申告
- 5. 消費税
 - 5.1 軽減税率制度
 - 5.2 消費税の端数計算
 - 5.3 課税事業者(納税義務者)
 - 5.4 免税事業者
 - 5.5 消費税に関する売上高・税額の名称
 - 5.5.1 課税売上高
 - 5.5.2 売上税額
 - 5.5.3 仕入税額
 - 5.5.4 仕入税額控除
 - 5.6 消費税に関わる期間の名称
 - 5.6.1 基準期間
 - 5.6.2 特定期間
 - 5.6.3 課税期間

とほほのその他入門

仏教

とほほの仏教入門

トップ / とほほの仏教入門

目次

- はじめに
- 仏教とは
- 本分派
 - 大乗仏教(だいじょうぶつぽうまう)
 - 上座部仏教(じょうざぶぶつぽうまう)
 - 小乗仏教(しょうじょうぶつぽうまう)
 - 浄土(じんと)
 - 密教(みつまう)
- 仏教の階層
 - 加害(かがい)
 - 菩薩(ぼさつ)
 - 明王(めいおう)
 - 空宗(くうしゅう)
- 如来(にょらい)
 - 娑婆世界(しゃかにょらい)
 - 娑婆世界(あみだにょらい)
 - 無量寿如来(むりやうじゆにょらい)
 - 大日如来(だいにくわにょらい)
 - 毘盧遮那如来(ひるしやなにょらい)
- 菩薩(ぼさつ)
 - 弥勒菩薩(みろくぼさつ)
 - 観音菩薩(かんのんぼさつ)
- 明王(めいおう)
 - 不動明王(ふどうめいおう)
- 志願(しごん)
 - 七福神(しちふくじん)

洋楽

とほほの洋楽入門

トップ / とほほの洋楽入門

コロナの影響でゴールデンウィークにドライブもできないので、今日はちと、洋楽についてまとめてみました。発表年順にしています。年は、そのアーティストが最初にヒットしたあたりの年でまとめています。ロックの歴史的なものから、ロックじゃないもの、個人的な好みも入り混じっています。というか、個人的好みが大半です。

1800年代

船乗り達

Johnny Boker (ジョニー・ボーカー) (19世紀初頭) [YouTube](#) [Wiki](#)
最初の Rock & Roll ってどんな曲だろうと思って探したらこんな曲にたどり着きました。19世紀初頭の船乗り歌です。Rock & Roll は元々船の縦方向の揺れ(rocking)と横方向の揺れ(rolling)の意味だったそうですが、この曲では「Do my Johnny Boker, Come rock and roll me over (ジョニー・ボーカー、私のところに来て、私を揺らして)」と、ちょっとセクシャルな意味で使用されています。その後、Rock & Roll という言葉は、黒人音楽では魂が揺れ動くの意味で使用されたり、揺れる、ダンスする、騒ぐ、性交するなど、様々な意味で使用されるようになりました。

1934年

The Boswell Sisters (ボスウェル・シスターズ) [Wiki](#)
Rock And Roll (ロックン・ロール) (1934年) [YouTube](#) [Wiki](#)
アメリカのミュージカル&コメディドラマ映画「大西洋横断メモリー・ゴーラウンド」の中で歌われた曲です。この曲はまだ船の揺れの意味合いが強いですが、でも「Rock & Roll」が「楽しいこと」を示す言葉として使われ始めたような気がします。

1947年

Roy Brown (ロイ・ブラウン) [Wiki](#)
Good Rockin' Tonight (グッド・ロッキン・トゥナイト) (1947年) [YouTube](#) [Wiki](#)
ナイトクラブで歌っていたロイ・ブラウンのオリジナル曲です。この曲はまだジャンプ・ブルースと呼ばれていました。彼がファンだったフレイジー・ハリス が曲にやって来た時、この歌をハリスに歌ってほしいと申し出ましたが受けませんでした。後日、別のブルース歌手セシル・カントにこの曲を聴かせたところ、カントは思いなりレコード会社の社長に電話、そのまま朝の4時に録音スタジオに連れていかれ、

点字

とほほの点字入門

トップ / とほほの点字入門

ある日ビールを飲んでいたら、ビールの缶に下図のような突起がありました。点字です。何で書いてあるの知りたくなり、WWW入門をちょっとお休みして、番外編「点字入門」を作成してみました。このページは出典さえ明記していただければ、コピー、印刷、ばらまき自由ですのでご利用ください。



目次

- マス(6つの点)
- 母音(アイウエオ)
- 子音(アカサタナ)
- 濁音(ガギクゴ)
- 半濁音(ハピプペホ)
- 拗音(マイユエヨ)
- 拗濁音(ギャギョ)
- 拗半濁音(ビャビュビョ)
- 外来発音(ヴァヴィヴォ)
- 「ン」,促音(っ),長音(ー),読点(。),句点(。)
- 数字(1 2 3 4 5)
- アルファベット(A B C)
- 規則
- 最後に・・・

カーブ

とほほのカーブ入門

トップ / とほほのカーブ入門

目次

- 球団名称
- 経営
- 本拠地・球場
- 今年の成績
- 歴代順位
- 歴代優勝
- ユニフォーム
- 歴代背番号
- 永久欠番
- 歴代監督
- 歴代エース(先発最多勝)
- 歴代最多勝投手
- 歴代最優秀投手
- 歴代ノーヒット・ノーラン
- 歴代首位打者
- 歴代本塁打王
- 歴代盗塁王
- 殿堂入り
- 他球団への移籍選手
- 他球団からの移籍選手

映画

2025年に見た映画

トップ / 映画 / 2025年

パリタクシニ★

フランス映画。免許寸前で金欠のタクシニ運転手シャルル。92歳の老女をパリの反対側の老人ホームまで送る仕事を引き受けます。途中で彼女と話したり寄り道しているうちに...

オッペンハイマー

ノーラン監督の話題作。原子爆弾を開発したオッペンハイマーを描く作品。アカデミー賞作品賞、監督賞、主演男優賞、助演男優賞、編集賞、撮影賞、作曲賞など7部門を総なめ。感想としては...うーん、長かったな...

すずめの戸締り

新海誠監督。九州の女子高生すずめ。災いの扉を代々閉めてまわる草太。椅子にされてしまった草太と戸締りの旅に...。新海誠って新しい作品になるごとに女の子が活発になっているような気がします。

シェフ! ミツ星レストランの舞台裏へようこそ

前回はたつけどこの続かない料理人ジャッキー。ジャッキーが尊敬するミツ星レストランのラガルドを「LEON」のジャン・レノが演じてます。ドタバタコメディ映画。

ボテ子★

伊坂幸太郎の同名中編小説を映画化。同じ日に同じ病院で生まれながら、空き巣を生業とする今村(濱田岳)とプロ野球選手の尾崎。今村の恋人役を木村文乃が演じてます。大森南朋もよかったな。

ウイ、シェフ!

前回は自信あるけどオーナーと喧嘩して店をやめた女性シェフカティ。移民の少年たちが生活する施設で働き始めます。前半は登場者の大半がなかと怒りまくっていつつまらなかつたのですが、後半はまあまあ。

そんなことを考えながら続けてます & 続けていきます。

とほほのWWW入門 検索 (By Google)

とほほの娘がLINEスタンプはじめました



メニュー	■ はじめに	
HOME	ご使用上の注意 (1)	管理者へのメール (1)
フレーム版	主な更新履歴 (24)	自己紹介 (1)
ダウンロード	■ 基本編	
ラウンジ	用語集 (82)	逆引きリファレンス (32)
URLの広場	Webページ作成入門 (7)	
	■ フォーマット	
	HTML (400)	SVG (1)
	HTML5 (7)	VML (1)
	XHTML (1)	GIF (1)
	MathML (1)	CSV (1)
	DTD (1)	セマンティック・ウェブ (1)
	JSON (1)	
	■ CSS	
	CSS (ABC順)(719)	Sass (1)
	Bootstrap (61)	CSSフレームワーク (1)
	Less (1)	リセットCSS (1)
	■ プログラミング言語	
	JavaScript (39)	Scala (1)
	TypeScript (1)	Haskell (1)

ご清聴ありがとうございました

以下、過去記事

「とほほのPerl入門」更新しました

- Perl入門がかなり陳腐化していたので
約23年ぶりに更新しました

長期にわたり
放ってしまって
すみませんでした....

Java も1.4時代のものなので...
更新していきます...

🐘 とほほのperl入門

[トップ](#) > とほほのperl入門

概要編

- [Perlとは](#)
- [インストール](#)
- [実行方法](#)
 - [引数指定](#)
 - [ファイル指定](#)
 - [自己完結型](#)
 - [標準入力読み込み型](#)
- [基礎知識](#)
 - [簡単な実行例](#)
 - [これだけは覚えて](#)
 - [strictモードとwarningモード](#)
 - [関数の括弧](#)
- [定数](#)
 - [数値](#)
 - [文字列](#)
 - [エスケープシーケンス\(¥\)](#)
 - [数値と文字列](#)
 - [真\(true\)と偽\(false\)](#)
 - [バイナリデータ](#)
- [変数](#)
 - [変数宣言\(my\)](#)
 - [スカラー\(\\$XXX\)](#)
 - [配列\(@XXX\)](#)
 - [ハッシュ\(%XXX\)](#)
 - [型グロブ\(*XXX\)](#)
 - [変数名に使える文字](#)
 - [定義済変数みと未定義変数](#)
 - [省略時の変数\(\\$_\)](#)
 - [環境変数\(\\$ENV{'XXX'}\)](#)

Perlのここんとこが好き

```
$a = "2024-02-04 13:24:56";  
if ($a =~ /(\d+)-(\d+)-(\d+) (\d+):(\d+):(\d+)/)  
{  
    $year = $1;  
    $mon = $2;  
    $mday = $3;  
    $hour = $4;  
    $min = $5;  
    $sec = $6;  
}
```

**正規表現を演算子で記述できて、
マッチ部を簡単に取り出せるところ。**

Perlでtry catchがサポートされました (実験的)

- Perl 5.34 (2021/5/20)

```
use feature qw(try);
no warnings "experimental::try";
try {
    my $err = func();
    if ($err) { die "ERROR!!!" }
} catch($e) {
    print "catch: $e";
}
```

Perlでクラス(class)がサポートされました (実験的)

• Perl 5.38 (2023/7/2)

```
use v5.38;
use feature 'class';

class MyClass {
    field $param1 = "";

    ADJUST {
        print "MyClass object created.¥n";
    }
    method hello() {
        print "Hello world!¥n";
    }
}

my $obj = MyClass->new;
$obj->hello();
```

クラス宣言

クラスフィールド

コンストラクタ

クラスメソッド

その他番外編

他にも、ちょっと気になったこと、いろいろ紹介しているので、見に来てください。
今回のテーマである、「**お好み**」の紹介ページもあります。

■ アラカルト

[アラカルト](#) (42)

[How To](#) (39)

[トラブルシュート](#) (2)

[ショートカット](#) (1)

[厳選ツール集](#) (10)

[厳選リンク集](#) (1)

[杜甫々の本](#) (1)

[スペシャルサンクス](#) (1)

■ 番外編

[点字](#) (1)

[映画\(2021年\)](#) (1)

[珍しい苗字](#) (1)

[陶磁器](#) (1)

[仏教](#) (1)

[韓国語](#) (1)

[中国語](#) (1)

[洋楽](#) (1)

[所得税](#) (1)

[相続](#) (1)

[資産運用](#) (1)

[法令](#) (1)

[広島ラーメン](#) (1)

[広島お好み焼き「八昌」](#) (1)

[タイ料理](#) (1)

[お酒](#) (1)

[三大〇〇](#) (1)

[とほほの雑記帳](#) (1)

お好み焼き

陶磁器

- 去年は
出石焼、越前焼、
九谷焼、珠洲焼、
をまわってきました。
- よい旅だったのですが、
ですが...元旦から...



まとめ

- 「継続は力なり」 < 「好きこそものの上手なれ」
- 説明サイト と リファレンスサイト の両立を目指す
- 可能な限りコンパクトに。情報量は落さず情報密度を上げる
- 目次と検索は重要
- 着目して欲しい箇所だけを目立たせよう
- 翻訳しただけで安心しない
- 初心者には優しく。初心者目線を保てることも大事

自分の中での座右の銘

いちぎょうにゆうこん
一行入魂

昔の雑誌に掲載されていたものですが、

「覆水ディスククラッシュ」や

「一寸(ちょっと)の虫(バグ)にも五分(ごふん)のデバッグ」ってのも好き