

「とほほのWWW入門」の 26年を振り返って

杜甫々（とほほ）

自己紹介

- 「とほほのWWW入門」管理人
 - 今年でもう26年になります
- 広島生まれ。広島在住。カープファン
 - 今年の観戦成績は5勝1敗。まずまず
- 本名は 岡山県玉野市 に多いみたい
 - 広島には10人程度。玉野市では 山田 より多い
 - 祖先は岡山出身だったのかも



個人的歷史

コンピューターとの出会い（1981年）

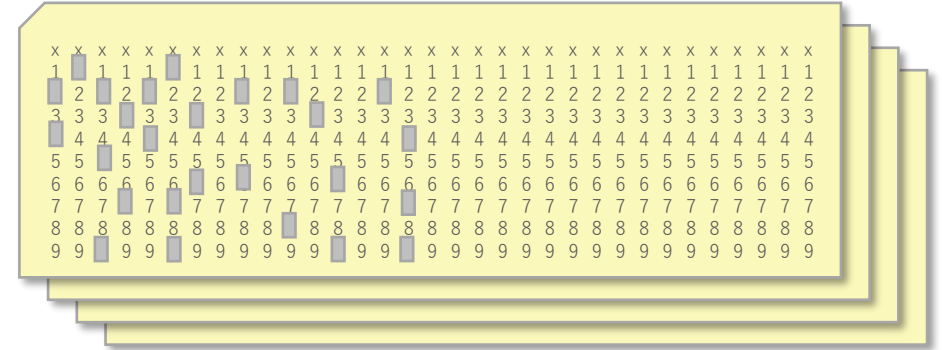
- **コモドール VIC-1001**
 - ブラウン管テレビに接続して使用
 - CPU：初代ファミコンと同じ 6502
 - メモリ：5KB
 - 外部媒体：カセットテープ
- 兄弟で金を出し合い、3KB の拡張メモリパックを追加



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

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)
 - 教授の意向で磁気ディスク使用禁止パンチカードを使用
- C言語 (1987年)
 - はじめて請け負ったプログラミングのバイト
 - 最初は BASIC の移植のはずだったが、いつの間にかC言語で新規開発することに
 - 最初に開発した化粧品販売管理システムのバイト収入は3万円でした。
- COBOL (1987年)
 - キーパンチャー (プログラムの入力のみ) のアルバイト



インターネットとの出会い（1988年）

- 入社。はじめてインターネットに触れる
 - 会社で研究所っぽいところに配属され、TCP/IP、インターネットに触れる
 - いきなり、UNIX カーネルコードを読まされ、いきなり、TCP/IP の講師をさせられる

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？

その頃のインターネット・・・

- **JUNET（ジュネット）**

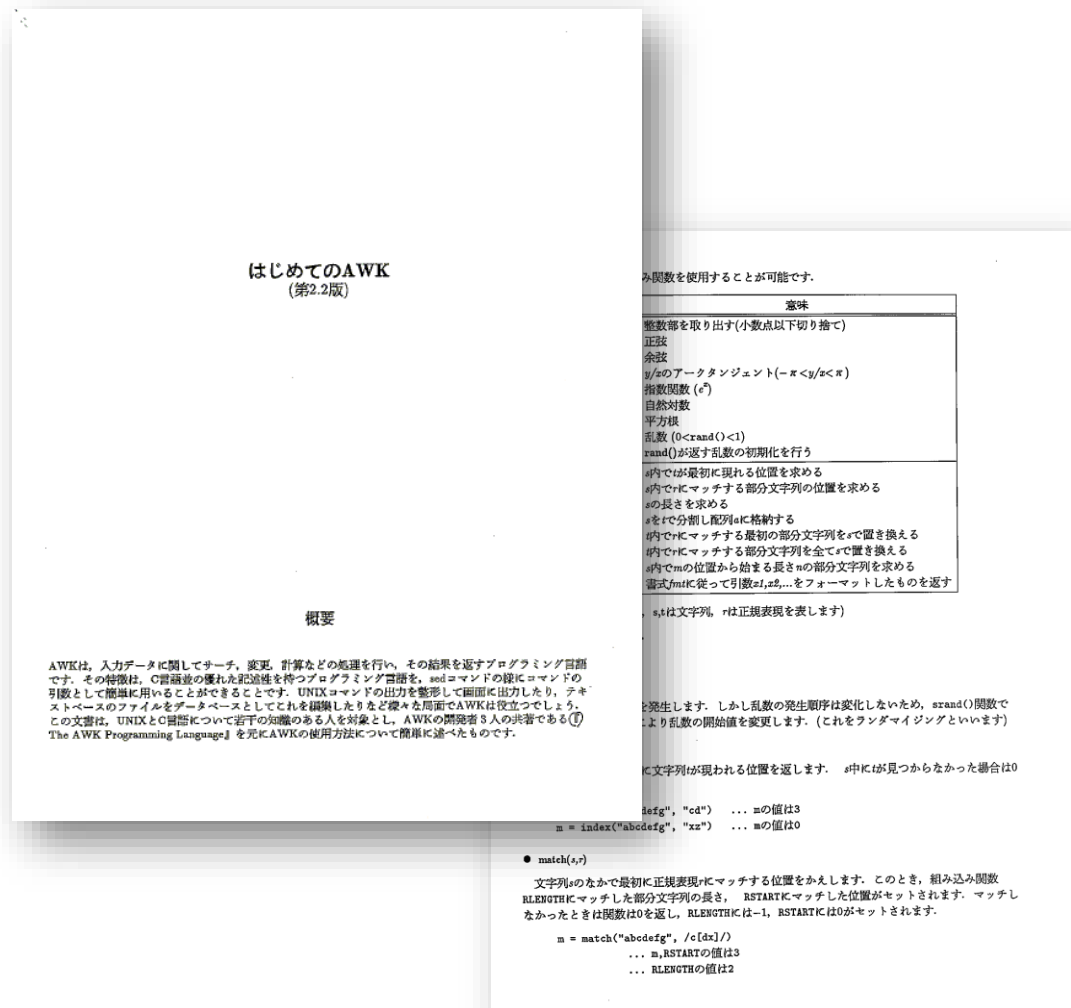
- 慶応義塾大学の 村井純 先生がはじめられた日本最初のインターネット
- メールとネットニュース（fj.jokes等）の配信
- 学術目的のみで使用可能。商用利用は不可
- 商用メールは JUNET を経由しないよう、経路を指定して記述したりしていました

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

「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！



「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



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

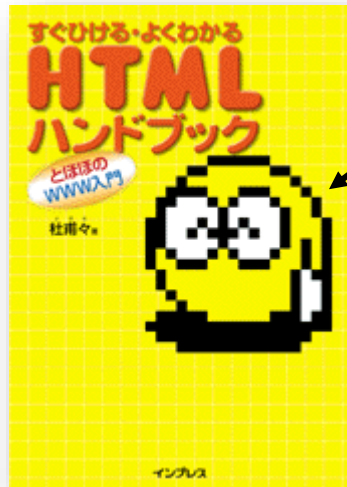


popase.gif



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

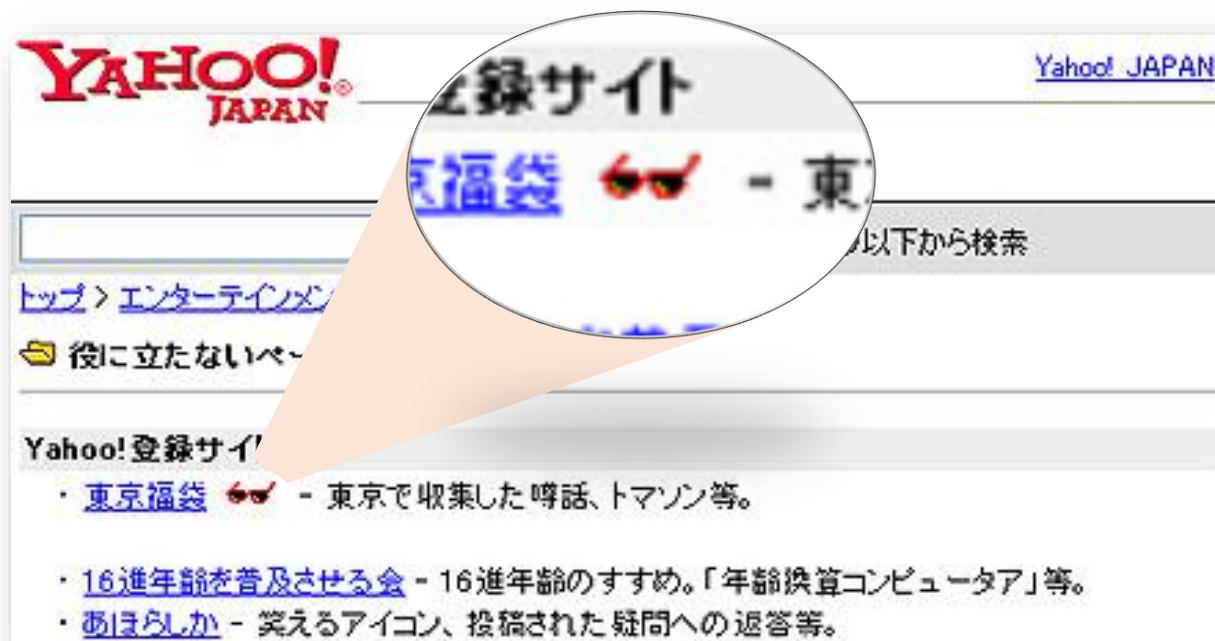


各社公式サイトから引用

もう一人は CMプロデューサーの 佐藤雅彦 さん

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/yst/2004/12/03/>
より引用

ラウンジ（1997年～2000年）

- 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ね

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<B R>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、<T A B L E>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。

B u n n y 4 [\[E-Mail\]](#) [\[HomePage\]](#) 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H T M Lに重点を置いてます、わたしは。

スタイルシート論争（2000年）

- W3C（WWW Consortium）が提唱する HTML 4.0 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・・・。

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ます...

杜甫々の主張	HTML5の現在の状況
W3C は所詮、正規の標準化機関ではない	現在は W3C ではなく WHATWG が標準化を行っています
SGMLアプリケーションであることをあきらめた方がよいのでは？	あきらめました
過去の歴史的な機能は歴史的理由によりサポートを継続すべきでは？	ブラウザは過去の機能もサポートすべきという方向に変わりました
機能（タグ・要素）の拡張性も持たせたい	機能を拡張することができるようになりました
仕様ばかりを先行策定するのではなく、実装事例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
<xx>～</xx> は <yy>～</yy> で囲めない等の規則をなるべく排除したい	一部ですが排除されました

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

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

「とほほの〇〇入門」

・ なんだかんだで、趣味の範囲で〇〇入門の執筆を継続

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）

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）

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...

「とほほのWWW入門」 執筆時に気をつけていること

○○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、自分としてはニガテ・・・

サンプルも極限まで短く

- サンプルコードもできるだけ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...



翻訳ただで安心しない

- 英語の仕様書を日本語に翻訳ただけでは不十分ことがある
 - 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
- 何を書いてあるか（書いてないか）がわかる
- 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

とほほのOpenSSL入門

[トップ](#) > とほほのOpenSSL入門

目次

- [OpenSSLとは](#)
- [OpenSSLのインストール](#)
- [OpenSSLのバージョンとサポート終了日](#)
- [OpenSSLの主なオプション](#)
- [OpenSSLのコンフィグファイル](#)
- [よく使われる拡張子](#)
- [PEM形式](#)
- [SSLとTLSについて](#)
- [OpenSSLのヘルプを表示する](#)
- [ランダムなデータを生成する](#)
- [ハッシュ値を生成する](#)
- [BASE64エンコード・デコードする](#)
- [秘密鍵を作成する\(パスフレーズ無し\)](#)
- [秘密鍵を作成する\(パスフレーズ有り\)](#)
- [公開鍵を作成する](#)
- [証明書署名要求\(CSR\)を作成する](#)
- [証明書\(CRT\)を作成する](#)
- [共通鍵を用いて暗号化・復号化する](#)
- [公開鍵を用いて暗号化・復号化する](#)
- [サーバと接続テストを行う](#)
- [自己署名証明書\(オレオレ証明書\)](#)
- [オレオレ認証局\(CA\)](#)
- [エラーメッセージ](#)
- [リンク](#)

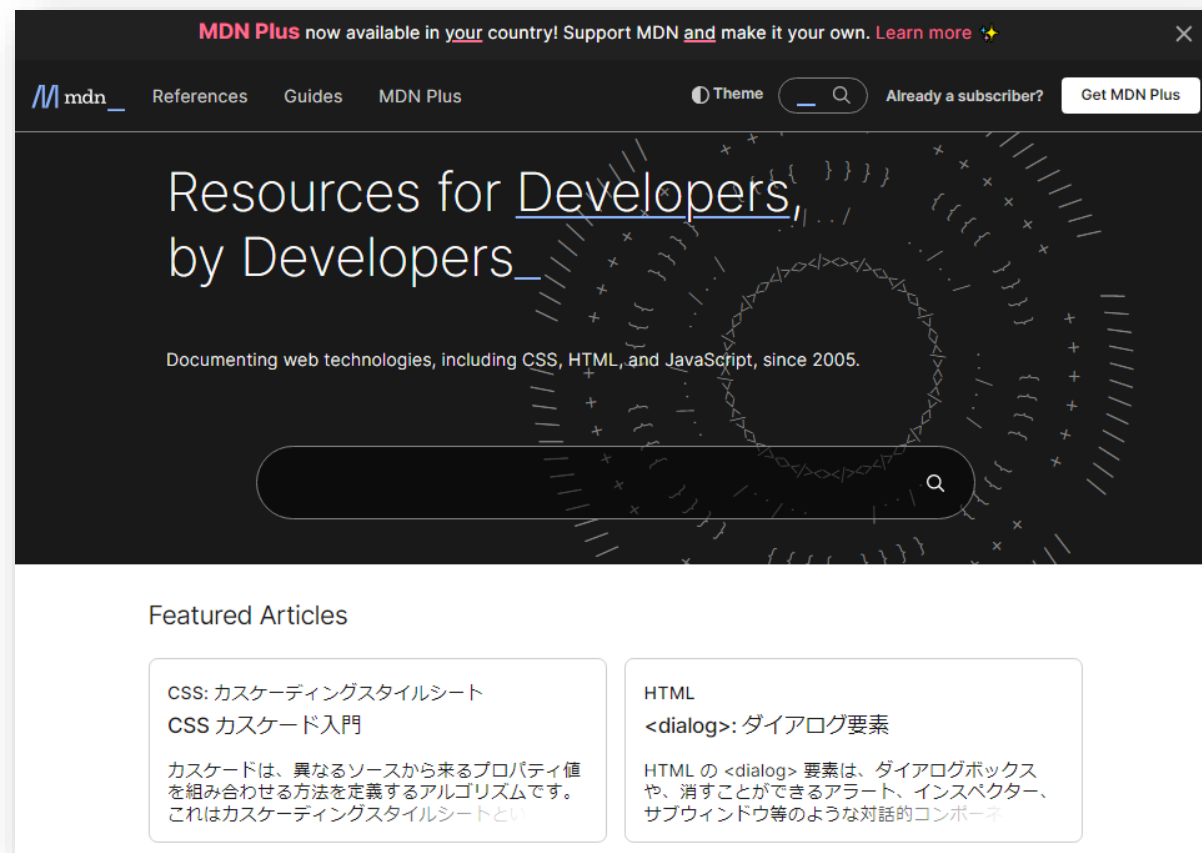
索引も大事

- 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	
表記について	@charset	hyphenate-character	
CSS3関連規約	:checked	hyphens	
記号・その他	I		R
*{...}	image-orientation		:read-only
E{...}	image-rendering		:read-write
ns E{...}	ime-mode		:required
E.F{...}	@import		resize
E.class	!important		<resolution>
E#id	:in-range		richness
E.F{...}	:indeterminate		right
E > F{...}	inherit		:root
E + F{...}	initial		row-gap
E ~ F{...}	inset		ruby-align
E .F{...}	invalid		ruby-overhang
[attr]	:is()		ruby-position
[ns attr]	isolation		
[attr="val"]			S
[attr~="val"]			:scope
[attr "val"]			scrollbar-3dlight-color
[attr^="val"]			scrollbar-arrow-color
[attr\$="val"]			scrollbar-base-color
[attr*="val"]			scrollbar-darkshadow-color
A	J		scrollbar-face-color
accent-color	justify-content		scrollbar-highlight-color
:active	justify-items		scrollbar-shadow-color
::after (:after)	justify-self		scrollbar-track-color
	K		::selection
	@keyframes		size
	L		speak
	:lang(c)		speak-header
	:last-child		speak-numeral
	D		
	cue		
	cue-after		
	cue-before		
	cursor		

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
していたが、最近は Microsoft や
Google も参画



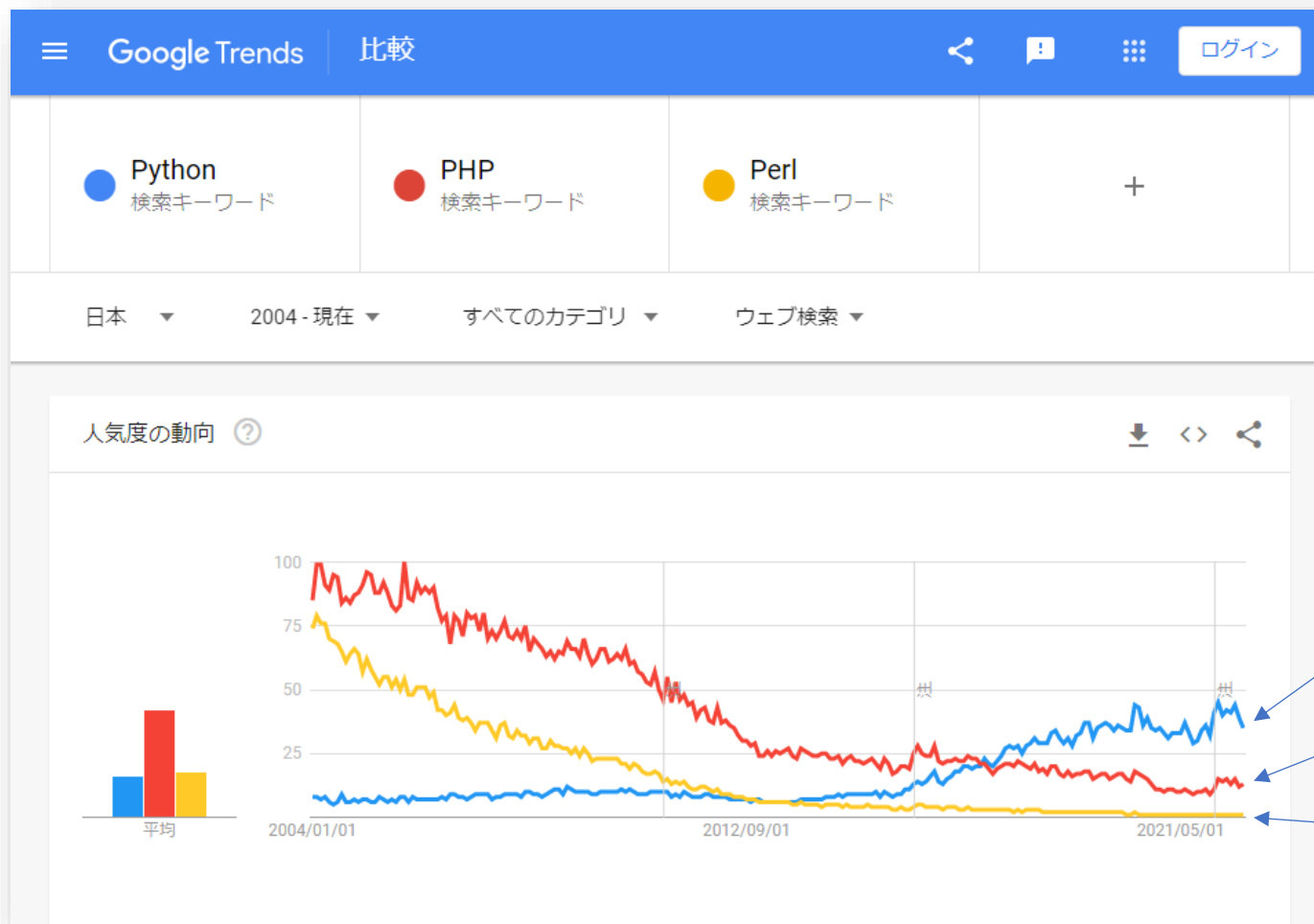
HTML/CSS/JSサポート状況の調査

- <https://caniuse.com/> がとても便利



普及度の調査（Googleトレンド）

- <https://trends.google.co.jp/> が便利



指定キーワードの
過去検索回数を
グラフ化してくれる

Python

PHP

Perl

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

よく聞かれること

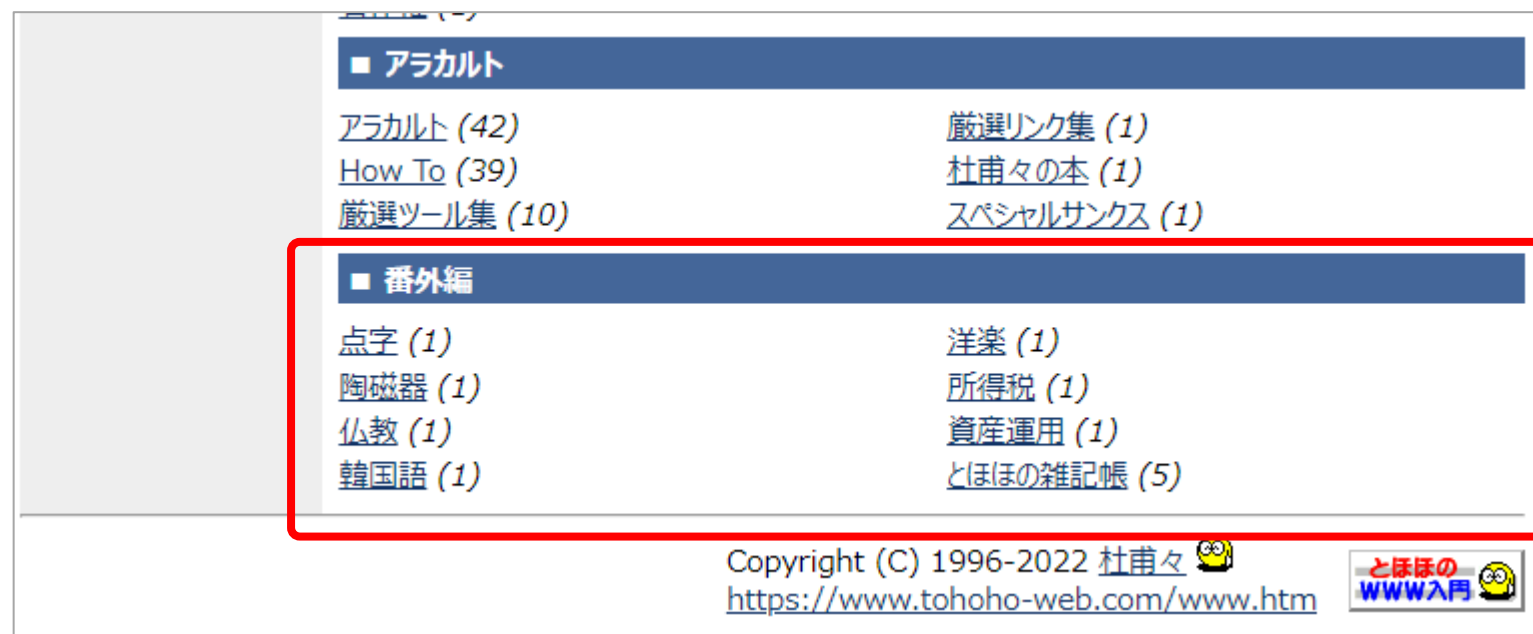
- 継続の秘訣はなにですか？
 - とくに秘訣はなくて...趣味だから...

番外編

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

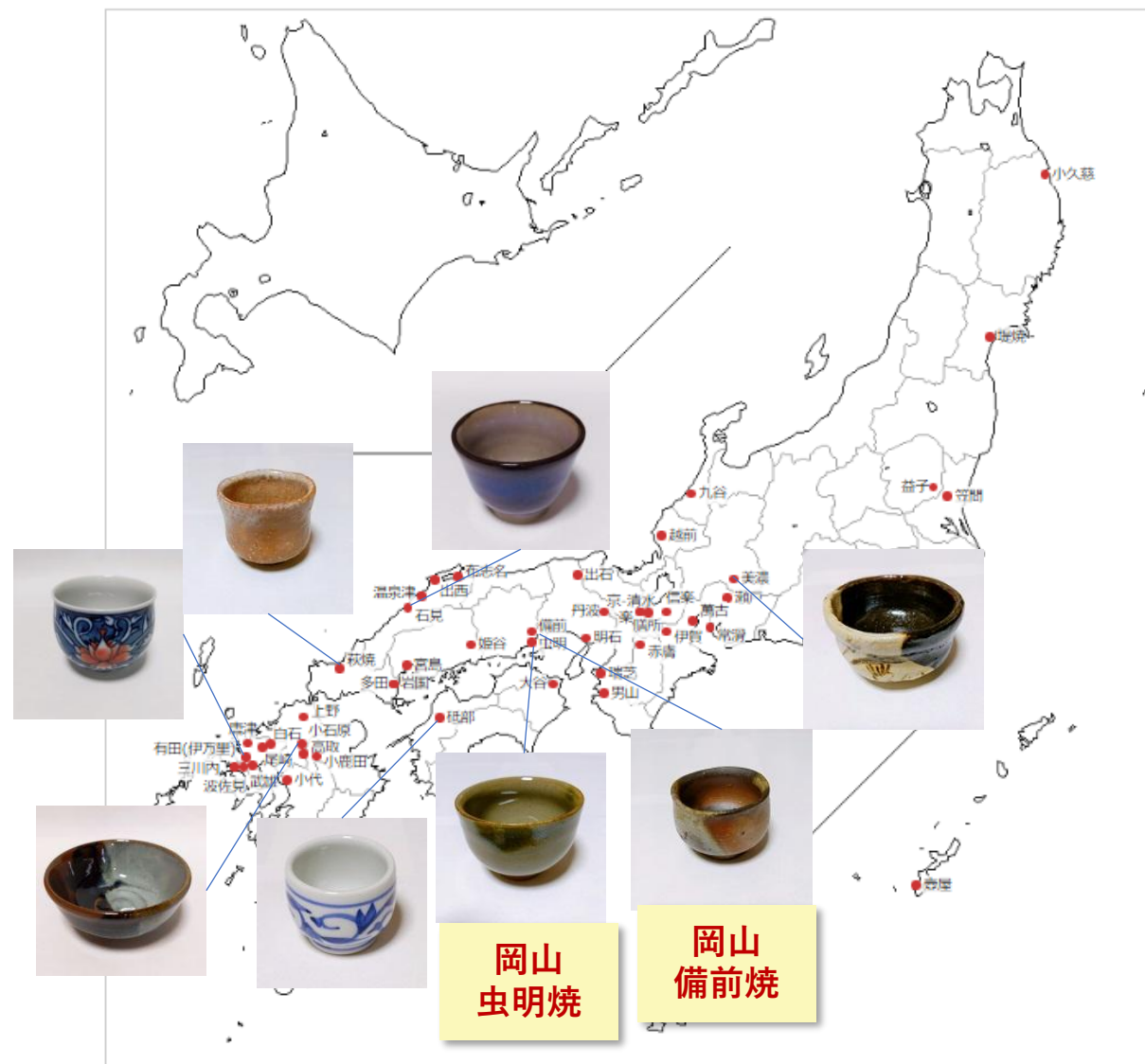


こちらへも
息抜きに
見てみてください

- 食べ歩き (1,376店舗) や スーパー銭湯巡り (30店舗くらい?) も**
紹介していけたら...

とほほの陶磁器入門

- 日本各地を回って集めた
陶磁器を紹介



白地図: <http://www.craftmap.box-i.net/>

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

ご清聴ありがとうございました