

— 短く簡素な指示で PC などに川柳を縦書きランダム呈示 —

①はダウンロードした PDF ファイルの PC 用 Chrome による表示と検索、②と③は作成した HTML ファイルの PC 用 Edge と Android 用 Chrome によるランダム呈示。

おれおれと名のって妻にすぐ切られ
 (東京都 60歳 男性)

こんには笑顔で答えて名を聞けず
 (高知県 56歳 女性)

忘れえぬ人はいるけど名を忘れ (愛知県 66歳男性) 2005=H17

優先席座って行き先 山歩き (神奈川県 66歳女性) 2007=H19

注目を一身に受け 餅食べる (千葉県 39歳男性) 2009=H21

チョイワルもチョイヨボですネと妻が言ひ (京都府 59歳男性) 2006=H18

フクチンのネット予約でひ孫借り (東京都 81歳男性) 2021=R03

②

朝起きて調子いいから医者に行く (埼玉県 77歳男性) 2018=H30

目の検査「丸」と答えるお爺ちゃん (東京都 74歳女性) 2021=R03

デパートで買ひ物よりも椅子探し (東京都 30歳男性) 2006=H18

還暦はシルバーシートを浅く掛け (神奈川県 72歳男性) 2004=H16

老医師の過労気遣う老患者 (埼玉県 69歳男性) 2005=H17

出典…全国有料老人ホーム協会「シルバー川柳」入選作

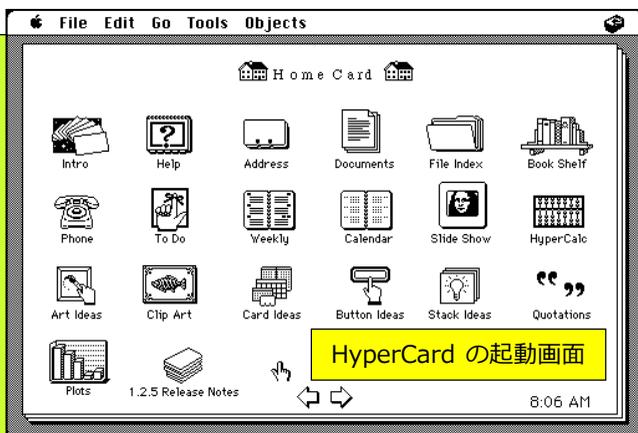
前車の覆るは後車の戒め
 (買誼：かぎ 200-168 BC)
 Let another's shipwreck
 be your seamark.

③

目次

- 01 「新米高齢者」に有益そうなシルバー川柳・・・ 1
- 02 まずは数句を縦書き呈示・・・ 3
- 03 数句を「配列」から無作為抽出・・・ 5
- 04 数句を重複なく（ほぼ）無作為抽出・・・ 7
- 05 機械的反复処理にはキーボードマクロが便利・・・ 9
- 06 タブレットなど用には「これで完成」ですが・・・ 11
- 07 短冊表示.htm の作成・・・ 13
- 08 「高齢者向け“脳トレ”言語」としての JavaScript・・・ 15

①A 5 判見開きでの閲読を想定して作成した本冊子の内容は、筆者が Windows 7 を OS とする PC 上の Chrome、Edge、AVG Secure Browser、Firefox 及び Android を OS とする tablet 上の Chrome で確認した結果をまとめたものです。お気づきの点などありましたら atsushi.kato.1958@gmail.com までお知らせください。



②本冊子の最終産物である乱数川柳.htm と短冊表示.htm は以下の link で利用できます（ファイル名を silver.htm と tanzaku.htm に変更している理由は使用中の無料 server が日本語ファイル名に対応していないためです）。

<http://mmua.html.xdomain.jp/bak/senrvu/silver.htm> ..39KB

<http://mmua.html.xdomain.jp/bak/senrvu/tanzaku.htm> .. 1KB

③本冊子の見開き閲読には Kinoppy の、script などのコピーには各種ブラウザの利用がオススメです。また、silver (=乱数川柳).htm と tanzaku (=短冊表示※).htm は、本冊子をブラウザなどで開いた状態で上の URL の右クリック→[名前を付けてリンク先を保存…]/[名前を付けてリンクを保存]/[Save Link As…]の選択で使用中の PC などに保存できます（各項目の名称は順に Chrome と AVG Secure Browser、Edge、Firefox の場合）。※function 窓()が{window.open("silver.htm","","resizable")}です。

【用語集】

CSS(Cascading Style Sheets)とHTML(HyperText Markup Language) :
Web 頁の要素(文字、画像、他頁への link…)と構造(見出し、段落、表…)を定義するのがHTML、それらの見せ方(スタイル=表示方法:背景、フォント、段組…)を指定するのがCSSです※。
※本冊子で使うCSSは<body style=…>、HTMLは
のみ。

HyperCard : (1987-2004) …目次下右の画像も参照

カードに文字や絵・画像、ボタンなどを配置でき、それらをクリックすると link 先のカードが開いたり HyperTalk で記述されたスクリプトが実行されたりする情報・開発環境だったそうです。つまり、現在の web 頁と同等の機能を(Mac 限定かつ offline で、ですが)1987年の時点で実現※していたアプリなのです。

※他方、当時の Mac は FDD 2 台の SE (モノクロ 9 インチディスプレイ)でも\$2900≒42 万円、20MBHDD モデルは\$3900≒57 万円もしました。今なら、ほぼ同じことが 3~5 万円程度の中古ノート PC やクロームブックの 10 数インチのカラーディスプレイ上でできます。よい時代になったものです。



JavaScript :

ブラウザなどに web 頁の要素を処理させるスクリプト言語です※。
※本冊子では、new Array(“”,“”,“”) (配列定義)、document.write() (頁出力)、for(i=1;条件;i++){ }と do{}while(条件) (反復処理)などを使用。

エディタ (editor) :

本冊子では文字情報編集アプリを指します。機械的反復処理にはキーボードマクロ機能、スクリプト記述には()などの対応関係明示機能のあるもの (Mery※など) がオススメです。

※<https://forest.watch.impress.co.jp/library/software/mery/>

キーボードマクロ (keyboard macro) :

記録 (=記憶) した一連のキー操作を別のキー操作で実行する機能で、正しく使えば繰り返し作業を効率的かつ正確に行えます。

クリップボード (clipboard) :

コピー/カットされた内容 (文字列、画像…)を一時的に保存しておく共有の (=別アプリからも利用可能な) メモリ領域です。

川柳 :

前句付から独立した切れ字・季語などの制約のない十七音の短詩。

01 「新米高齢者」に有益そうなシルバー川柳

1958年1月生まれの筆者も無事(?)65歳を迎え、同級生の多くの皆さんと共に高齢者の仲間入りです。そんな「新米高齢者」にとって興味深く有益そうな資料の1つに「シルバー川柳」があります。

それは公益社団法人全国有料老人ホーム協会が2001年から行っている川柳作品の公募活動の成果で、書籍としても出版され、各年の入選作(20~29句)はネット上にも公開されています(資料1上参照)。
<https://user.yurokyo.or.jp/senryu/pdf/405-1>

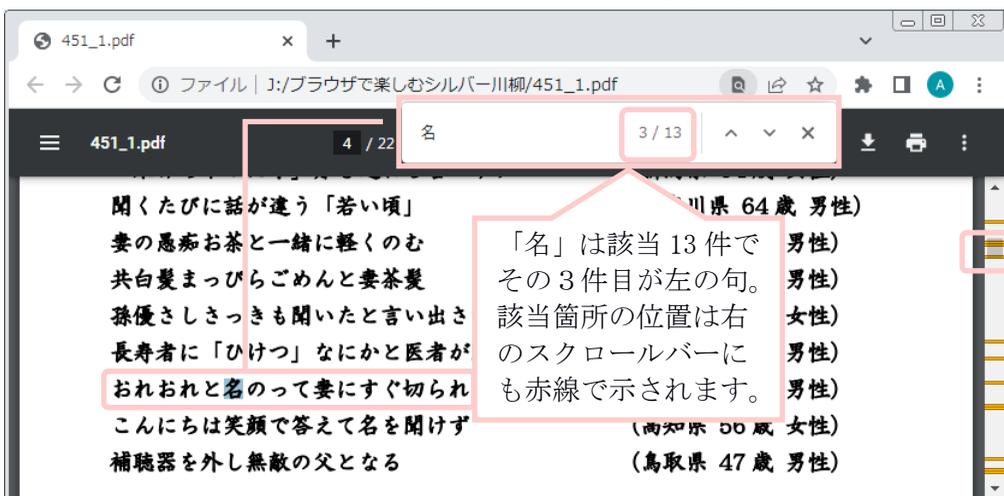
2001年の第1回から昨年の第22回までの入選作品は特別賞を除いて計456句で、年毎の「入選作品」のため分野分けなどはなされていません。従って、その楽しみ方・役立て方としては「通読しつつ気に入った作品を抜き出す」「自分流に分類し必要に応じて参照する」「キーワードで関連句を検索する」※などが考えられます。そして、これらの利用法はブラウザとワープロなどで可能です。
※ダウンロードしたPDFファイル(698KB)をブラウザなどで開けば([Ctrl]+[F]などで)ネット接続なしでの検索が可能です(資料1下)。

それらとは別に「折々にいくつかの句を無作為に抽出→呈示してその偶然性や組合せの妙(?)を楽しみ、気分転換の糧や将来の戒めなどとする」という活用法も考えられますが、PDFファイルからそれを行う機能はブラウザなどにはありません。また、川柳はやはり(横書きではなく)縦書きで読みたいものです。

実は、この無作為抽出と縦書き呈示は、川柳を書き込んだHTMLファイルにJavaScriptやCSSという「簡素な指示」を書き加えれば実現可能です。そこで、本冊子ではその具体的方法を、関連する基本的知識・技術などと共に順を追って紹介していきます。



資料1 公開されている入選作品の一覧 (上) ※及びブラウザでの検索の結果 (下) #
 ※ネット上の PDF ファイルを AVG Secure Browser で表示。
 #ダウンロードした PDF ファイルを Chrome で表示・検索。



02 まずは数句を縦書き呈示

いきなり 456 句全部を扱うのは大変なので、まずはザッと読んで目に留まった数句についてそれらを縦書き呈示してみます。

資料 1 の下に示した 451_1.pdf はテキストを含む (=コピーが可能) な PDF ファイルなので、適当な句をカーソルで反転させて範囲指定の上 (右クリックあるいは[Ctrl]+[C]などで) コピーし (資料 2 の上)、以下の囲み内の第 2 ~ 4 行のようにメモ帳などのエディタにペーストします (半角空白を挿入し入選年を追記してあります: // 下)。

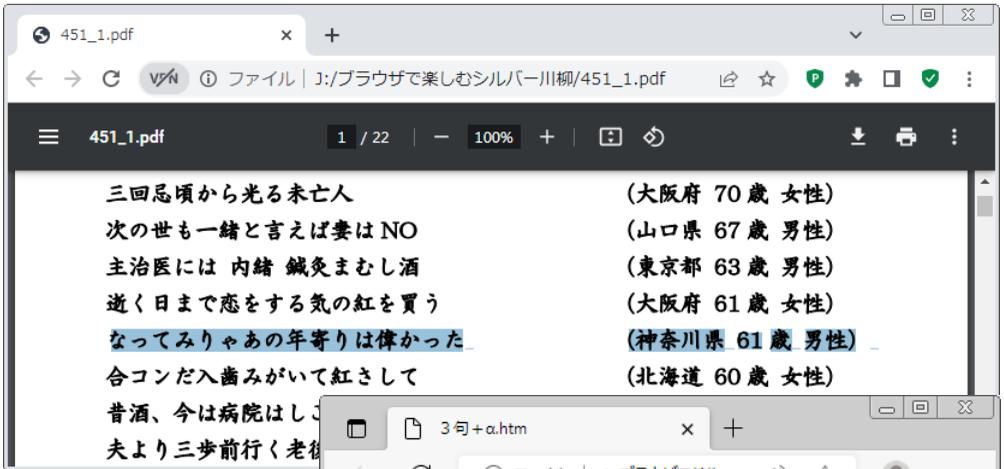
```
<body style=writing-mode:tb-rl;line-height:500%>
なってみりゃあ、あの年寄りには偉かった (神奈川県 61 歳 男性) 2001=H13<br>
八十路越え、大器晩成、まだ成らず (愛知県 81 歳 男性) 2006=H18<br>
何をしに、ここに来たかと考える (千葉県 79 歳 男性) 2018=H30
```

次に、<>で括った「見せ方に関するブラウザへの指示」を上のように書き足します。第 2・第 3 行の末尾の
は改行 (=BReak)、最初の<body style=...> (=本文の書式) の tb は縦書き (=Top to Bottom)、rl は右から左 (=Right to Left)、line-height は改行幅で 500%は行の右と左に 2 行分ずつ空ける (=200+100+200=500) という意味です※。※
は HTML の改行指定、<body style=...>は CSS の書式指定ですが、本冊子では特に区別せず「ブラウザへの指示」として扱います。また後者の指示は""で括っても ("writing-mode:500%"@資料 2 の下) 括らなくても有効です。

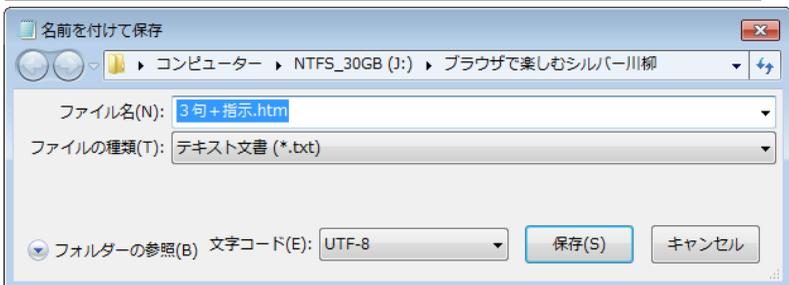
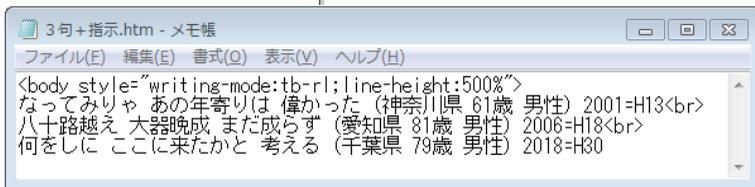
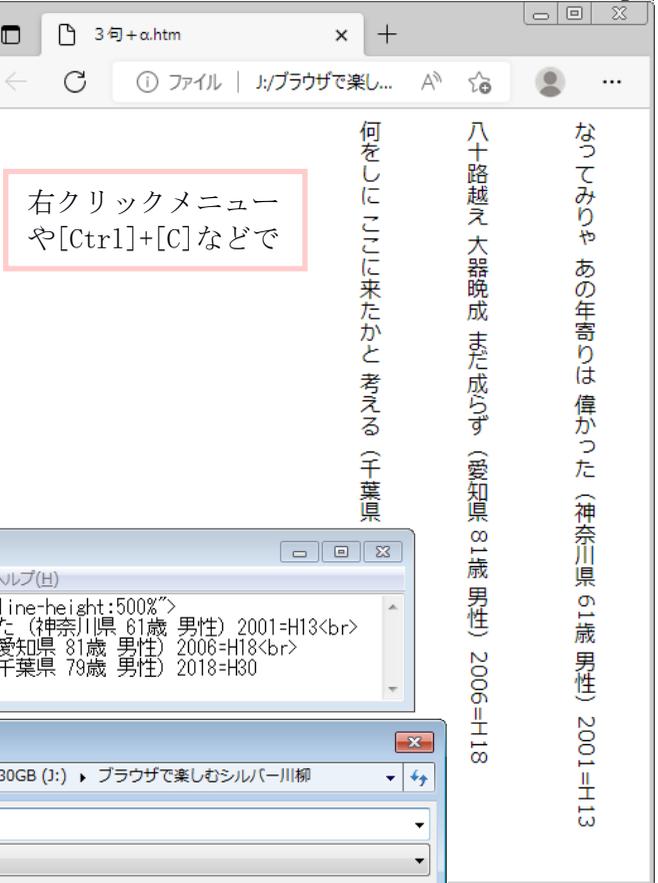
資料 2 の下のように囲み内の 4 行を「3 句 + 指示.htm」などのファイル名で保存してブラウザで開くと資料 2 の右のように表示されます。ファイルに保存したテキスト (=文) をブラウザで縦書き呈示するのに必要な最低限の指示はこれ (=1 行 + α) だけです。簡素ですネ! ; -)

【補足】

筆者の記述法は簡素を宗としており「正式」のものではありません (主なブラウザで期待通りに動けばそれで OK)。なお、拡張子は htm でも html でもかまいません。



資料2 上はテキスト
付き PDF の
範囲指定→コピーの
例、下はメモ帳へのペ
ースト→加工と保存の
例※、右は保存した
HTML ファイルの PC
用 Edge による表示。
※文字コードは UTF-8
がオススメです。



03 数句を「配列」から無作為抽出

数句を無作為抽出する方法としては「全句を1つの組にして連番を付け、必要回数発生させた乱数で選ぶ」のが最も素朴な案でしょう。

「1つの組」にするには配列 (Array) という機能を使います。以下の例の3行目の「句」が新しく作った組=配列 (new Array) で、(と)の間に" "で括った(今回は12の)要素を、で区切って書き並べます※。※
 (=改行) は各要素を頁に出力する時 (=末尾から2行目) に付けます。

```
<body style=writing-mode:tb-rl;line-height:500%>
<script>
句=new Array(
"なってみりゃ あの年寄りは 偉かった (神奈川県 61歳 男性) 2001=H13",
"天国の 友へ昼寝で 会いにゆく (北海道 81歳 男性) 2002=H14",
"聞かたびに 話が違う「若い頃」 (神奈川県 64歳 男性) 2004=H16",
"老医師の 過労気遣う 老患者 (埼玉県 69歳 男性) 2005=H17",
"八十路越え 大器晩成 まだ成らず (愛知県 81歳 男性) 2006=H18",
"振り返り 犬が気遣う 散歩道 (北海道 44歳 女性) 2011=H23",
"上向いて 歩こう今では 下見よう (埼玉県 69歳 男性) 2015=H27",
"こんにちは 思い出せずに さようなら (大阪府 73歳 女性) 2016=H28",
"遺言書「すべて妻に」と 妻の文字 (愛知県 59歳 女性) 2017=H29",
"何をしに ここに来たかと 考える (千葉県 79歳 男性) 2018=H30",
"じいちゃんの 敵は段差と パスワード (福岡県 53歳 男性) 2020=R2",
"おはようの ラインがくるのは 朝の五時 (沖縄県 17歳 女性) 2021=R3"
)
for (i=1; i<6; i++) {
番=Math.floor(Math.random()*句.length); document.write(句[番]+"<br>")
}
</script>
```

まず Math.random() で 0 以上 1 未満の乱数が、句.length (配列の長さ) で要素数=12 が得られ、両者の積 (乱数×要素数=0 以上 12 未満の小数値) を Math.floor で小数切捨した 0~11 のいずれかが選ばれる句の番号になります。その値に基づいて配列「句」の変数「番」目の

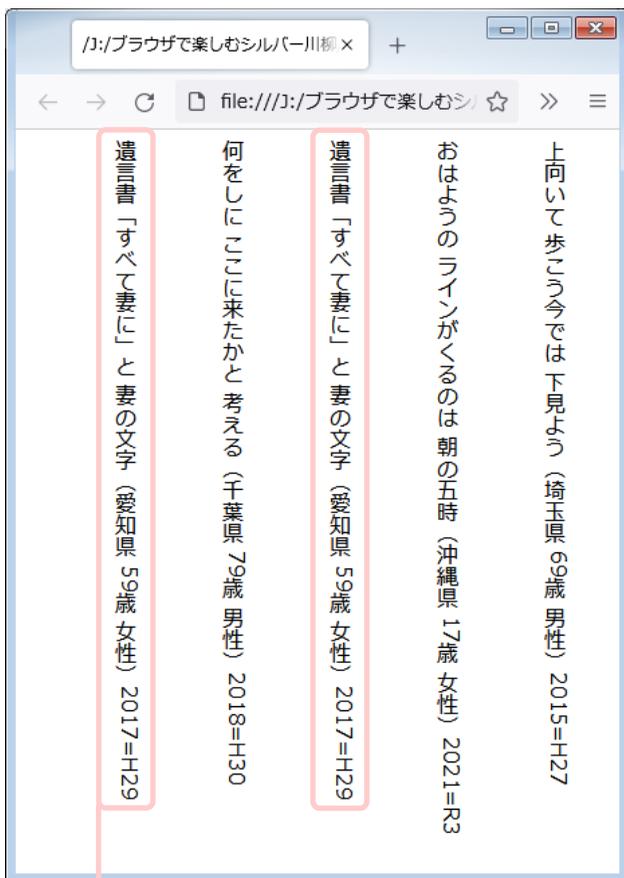
配列の最初の要素は 0 番!

要素の内容を改行付きで頁出力する処理 (= から) までの document.write(句[番]+
”)を5回繰り返さない (= for(i=1, i<6; i++)) という指示 () が JavaScript でブラウザに与えられています※。
 ※JavaScript の指示は<script>~</script> でその範囲を明示します。

囲み内の18行を「配列と乱数.htm」などのファイル名で保存してブラウザで開くと、例えば資料3のように表示されます。しかし、この方法では、12句から5句を機械的に抽出しているため、しばしば同じ要素=句が複数回選ばれてしまいます※。
 ※資料3の例では「遺言書…」が2回。

全456句から抽出する場合には、句の重複は頻繁には起こらないでしょうが、それを確実に回避するためには「もう一工夫」が必要です。

資料3 配列と乱数.htm
 の PC 用 Firefox
 での実行結果の一例※
 ※[F5]や更新アイコンなどで頁を更新するたびに異なる結果が得られます#。
 #Androidでは下向きスワイプなどで。



自筆でない遺言は無効ですが…

04 数句を重複なく（ほぼ）無作為抽出

2 句目以降の番号に「それ以前の値と異なる」という制約を課せば句の重複は起こりません。このような「条件付きの実行」には `do{処理}while(条件)`※という指示を使います。

※「()内の条件が真である間{}内の処理を繰り返す=偽なら次に進む」

以下の例では、下から 7 行目で最初の値を求めて変数 N に代入します（最初の N は制約無し）。次行では「N と異なる」という制約の下で `0` を求め、以下 P~R を「それ以前の値と異なる」ように求めます※。※==は「等しい」、||は「または」です。例えば R は N,O,P,Q のいずれかと同じ値である限り繰り返し求められる結果、いずれとも異なる値になります。

```
<body style=writing-mode:tb-rl;line-height:500%>
<script>
句=new Array(
“なってみりゃ あの年寄り は 偉かった (神奈川県 61 歳 男性) 2001=H13”,
“天国の 友へ昼寝で 会いにゆく (北海道 81 歳 男性) 2002=H14”,
“聞きたびに 話が違ふ「若い頃」 (神奈川県 64 歳 男性) 2004=H16”,
“老医師の 過労気遣う 老患者 (埼玉県 69 歳 男性) 2005=H17”,
“八十路越え 大器晩成 まだ成らず (愛知県 81 歳 男性) 2006=H18”,
“振り返り 犬が気遣う 散歩道 (北海道 44 歳 女性) 2011=H23”,
“上向いて 歩こう今では 下見よう (埼玉県 69 歳 男性) 2015=H27”,
“こんにちは 思い出せずに さようなら (大阪府 73 歳 女性) 2016=H28”,
“遺言書「すべて妻に」と 妻の文字 (愛知県 59 歳 女性) 2017=H29”,
“何をしに ここに来たかと 考える (千葉県 79 歳 男性) 2018=H30”,
“じいちゃんの 敵は段差と パスワード (福岡県 53 歳 男性) 2020=R2”,
“おはようの ラインがくるのは 朝の五時 (沖縄県 17 歳 女性) 2021=R3”)
N=Math.floor(Math.random()*句.length)
do{O=Math.floor(Math.random()*句.length)}while(O==N)
do{P=Math.floor(Math.random()*句.length)}while(P==N||P==0)
do{Q=Math.floor(Math.random()*句.length)}while(Q==N||Q==0||Q==P)
do{R=Math.floor(Math.random()*句.length)}while(R==N||R==0||R==P||R==Q)
document.write(句[N]+"<br>"+句[O]+"<br>"+句[P]+"<br>"+句[Q]+"<br>"+句[R])
</script>
```

前節の 03 では頁出力を for() の繰り返しごとに改行 (“
”) 付きで行いましたが、今度は全 5 句 (句[N]～句[R]) が決まった後に各句を改行で連結(+)した文字列を document.write() で頁出力しています。

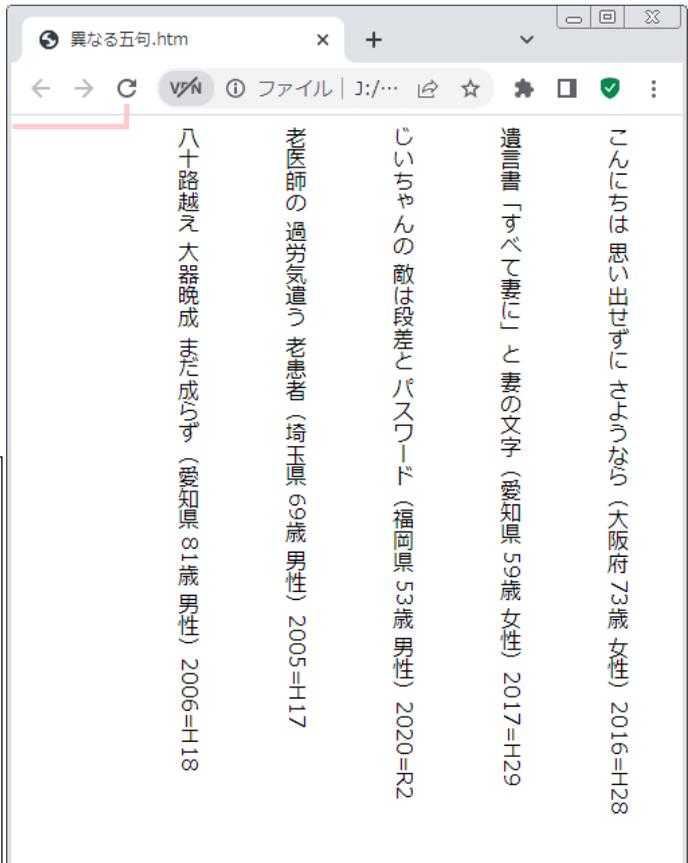
囲み内の 22 行を「異なる五句.htm」などのファイル名で保存してブラウザで開くと、例えば資料 4 のように異なる 5 句が呈示され、[F5] や更新アイコンなどで頁を更新すると次の異なる 5 句が呈示されます。今回の抽出は 0 以降の値に「それ以前の値と異なる」という制約が課されるため無作為ではありません。また、次の 5 句に前の 5 句中の句が選ばれる可能性は残ります。

更新アイコン

資料 4 異なる五句
.htm の A
VG Secure Browser
での実行結果の一例

【補足】

1. JavaScript は<script>と</script>の間に半角英数字で記述します。
2. ""で括った文字列や変数名・関数名などには全角文字も使えます。
3. =は右辺を左辺に代入、==は等しい(!=が等しくない)、||はまたは(&&がかつ)です。



05 機械的反復処理にはキーボードマクロが便利

異なる数句の重複のない（ほぼ）無作為な抽出とその縦書き呈示ができたので、ここで全 456 句を配列にしておきます。そして、形式を整えた配列要素化、つまり今回の場合なら「各句の末尾に入選年を追記して全体を””で括り、で区切る」といういささか面倒で退屈な加工については（メモ帳ではなく）「キーボードマクロ」機能付きのエディタを使うことで大幅な効率化が図れます。

筆者が常用しているエディタの Mery（最新版は 3. x. x ですが 2. 5. 6 を使用中）では、キーボードマクロは[Shift]+[F4]で記録の開始と停止がなされ、その間の全てのキー操作が（停止後の）[F4]で自動実行されます（資料 5 の上）。より具体的には PDF ファイルから 1 年分の入選作（20 句程度）を各句が 1 行になるようコピーした上で、まず下準備として最初の句の末尾に入選年と”を追記し（例：句の内容（作者の属性）2001=H13”）、この 2001=H13”の部分でカーソルを反転させて範囲指定→[Ctrl]+[X]で削除します。これで、その年の句に共通するこの文字列（2001=H13”）がクリップボードに保存されペースト可能になります（これで「下準備」は完了）。

続いて、最初の句の冒頭に（[Home]=[Fn]+[←]で）カーソルを移動し「①[Shift]+[F4]で記録（=記憶）の開始→②”を入力→③その行の末尾に（[End]=[Fn]+[→]で）カーソルを移動→④[Ctrl]+[V]でクリップボードの内容（例：2001=H13”）をペースト→⑤（[Home]=[Fn]+[←]で）句の冒頭にカーソルを移動の上[↓]で次行=次の句の冒頭にカーソルを移動→⑥[Shift]+[F4]で記録の停止」と操作します。これで[F4]を 1 回押す毎に次行=次の句に対して記録されている②～⑤のキー操作が自動的に行われます※。

※他方、五七五の切れ目への半角ブランクの挿入は、位置が不定なため句毎に自分で判断して行う必要があります。

要素の中では半角ではなく
全角の（と）を使います。

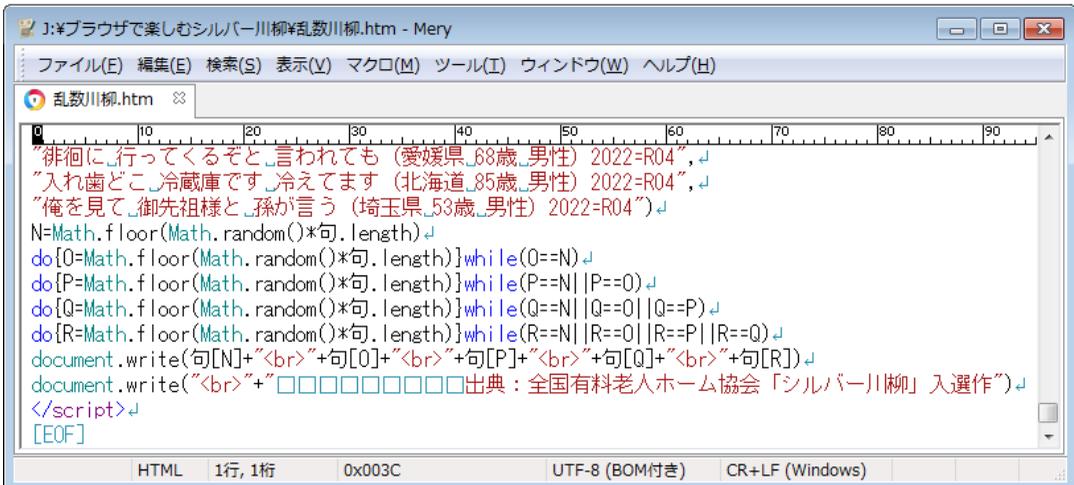
全 456 句の配列（第 4～第 453 句は省略）は以下のとおりです：

句=new Array(

「お若いわ」その一言で お得意さん（東京都 81 歳 女性）2001=H13”
“あなたには 言えない遊び 軽い嘘（千葉県 80 歳 女性）2001=H13”
“口づけも 入歯ガクガク 老いの恋（東京都 79 歳 男性）2001=H13”
：（中略）
“徘徊に 行ってくるぞと 言われても（愛媛県 68 歳 男性）2022=R04”
“入れ歯どこ 冷蔵庫です 冷えています（北海道 85 歳 男性）2022=R04”
“俺を見て 御先祖様と 孫が言う（埼玉県 53 歳 男性）2022=R04”



資料 5 エディタ Mery のマクロ関連のメニューと配列の最初の 3 要素（上）
及び配列の最後の 3 要素と典拠表示など（下）



06 タブレットなど用には「これで完成」ですが・・

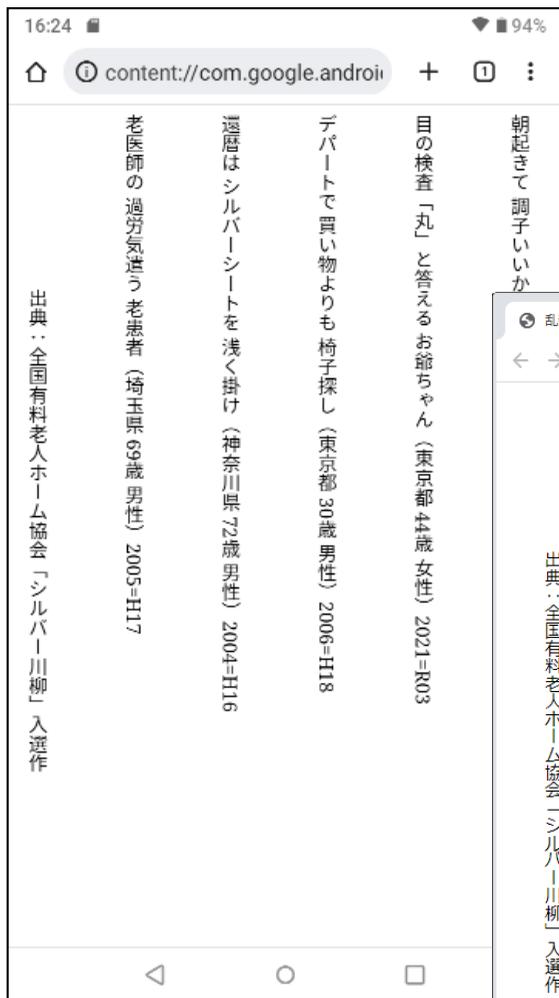
04の囲み内の「句=new Array(・・)」の・・の部分を実作成した456句の配列に置き換え、**ブラウザ画面の大きさ**と**出典表示**に関する指示を加えると以下の囲み内及び前頁の資料5に示したようになります。

```
<body style=writing-mode:tb-rl;line-height:500%>
<script>resizeTo(425,700)
句=new Array(
"「お若いわ」その一言で お得意さん（東京都 81歳 女性）2001=H13",
"あなたには 言えない遊び 軽い嘘（千葉県 80歳 女性）2001=H13",
"口づけも 入歯ガクガク 老いの恋（東京都 79歳 男性）2001=H13",
:（中略）
"徘徊に 行ってくるぞと 言われても（愛媛県 68歳 男性）2022=R04",
"入れ歯どこ 冷蔵庫です 冷えてます（北海道 85歳 男性）2022=R04",
"俺を見て 御先祖様と 孫が言う（埼玉県 53歳 男性）2022=R04")
N=Math.floor(Math.random()*句.length)
do{O=Math.floor(Math.random()*句.length)}while(O==N)
do{P=Math.floor(Math.random()*句.length)}while(P==N||P==0)
do{Q=Math.floor(Math.random()*句.length)}while(Q==N||Q==0||Q==P)
do{R=Math.floor(Math.random()*句.length)}while(R==N||R==0||R==P||R==Q)
句=句[N]+"<br>" +句[0]+"<br>" +句[P]+"<br>" +句[Q]+"<br>" +句[R]
出="          出典：全国有料老人ホーム協会「シルバー川柳」入選作"
document.write(句+"<br>" +出)
</script>
```

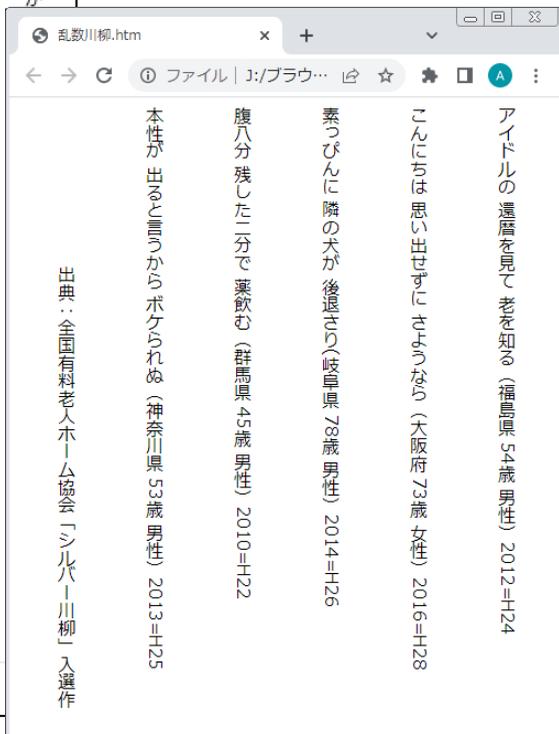
2行目のresizeTo(幅,高さ)の指定は本ファイルのみでは無効ですが、次節の07で有効にします。また、囲み内の</script>の前3行と資料5の下の</script>の前2行とは書き方が異なります※が結果的には同じ内容が頁出力されます。

※囲み内では、囲みの幅の1行に収まるように変数（句と出）を利用しています。

この内容を「乱数川柳.htm」などのファイル名で保存してブラウザで開くと、例えば資料6のように表示されます。左のAndroidでは通



資料6 乱数川柳.htm の Chrome
による実行例※
※Android tablet の場合（左）と
Windows PC の場合（右）



常 1つのアプリ画面のみが表示されるので、これで完成と言えます※。
※（下向きスワイプなどで）頁を更新すると、新たに選ばれた五句が呈示されま
す。必要なら画面の形状に合わせて改行幅・句数などを調整してください。

他方、右の Windows PC の場合、「乱数川柳」の出力はデスクトップ
上のブラウザの 1 タブとして（広い余白と共に）表示されがちです（…
資料6の右を含め、これまでの出力例は頁幅を手動で小さくして示し
ています）。乱数川柳を（短冊のような）独立した小窓（資料7）に
出力するには resizeTo を有効にする「もう一工夫」が必要です。

【補足】

JavaScript は<body>や<input>などのタグ (=要素) 中にも記述できます (囲み内の 1~2 行)。

07 短冊表示.htm の作成

Web 頁の大きさを変更する指定である `resizeTo` を有効にするには、乱数川柳.htm を「新たな窓」として開く別の HTML ファイルが必要です。他方、新たな窓を開く = ポップアップは広告表示などに濫用されがちなため、多くのブラウザではその自動実行をブロックする設定が標準になっています。

そこで①ポップアップが可能ならそれを自動実行して自分を閉じる → ②自動実行がブロックされたら手動実行用のボタンを呈示し、そのクリックなどでポップアップを実行して自分を閉じると指示する HTML ファイルを作ります (以下の囲み内がその一例)。

```
<body onLoad=窓() onBlur=window.close()>  
<input type="button" value=短冊表示 onClick=窓() onBlur=window.close()>  
<script>  
function 窓() {window.open("乱数川柳.htm", "", "resizable")}</script>
```

「新たな窓として乱数川柳.htm を開く」動作は①②に共通なので、4 行目の `窓()` という `function=関数` にまとめます。() 内の 3 つの "" は順に「新たな窓として開く HTML ファイル、その窓の名前 (省略可)、その窓の属性」なので、ここでは乱数川柳.htm、省略、resizable (サイズ変更可) とします※。

※関数名や value の "" は省略可ですが、これらの "" は省略不可です。

ポップアップ可なら `onLoad` (自分の読み込み完了) 時に `窓()` が実行されて新しい窓が active になり `onBlur` (自分の非 active 化) が発生するため `window.close()` で自分を閉じます。ポップアップ不可なら `窓()` が実行されないため `onBlur` → `close()` は起こらず、2 行目の [短冊表示] ボタンが呈示されます。そのクリックなど (`onClick`) で `窓()` が手動実行されると `onBlur` が発生し `window.close` が実行されます。

短冊表示

ポップアップがブロックされました

短冊表示

ポイントをクリック (あるいは[Tab]→[Enter])

資料7 ポップアップを
ブロックするブ
ラウザ※で短冊表示.htm
(乱数川柳.htm を新た
な窓として表示するた
めのファイル) を開いた時
の表示 (後景) 及びボタ
ンクリックなどで手動実
行された乱数川柳.htm
の表示例 (前景=右)
※この例では Edge

忘れえぬ人はいるけど名を忘れ (愛知県 66歳男性) 2005=H17
優先席座って行き先 山歩き (神奈川県 66歳女性) 2007=H19
注目を一身に受け 餅食べる (千葉県 39歳男性) 2009=H21
チョイワルもチョイヨボですネと 妻が言う (京都府 59歳男性) 2006=H18
ワクチンの ネット予約で ひ孫借り (東京都 81歳男性) 2021=R03

囲み内の4行※を「短冊表示.htm」などのファイル名で保存してPC上のブラウザ (例ではEdge) で開くと、アドレス欄の「ポップアップがブロックされました」というアラート (注意) と共に広い余白の左上隅に[短冊表示]ボタンが呈示されます (資料7の後景)。このボタンをクリック (あるいは[Tab]キーで選択→[Enter]) すると後景の窓は閉じて前景の短冊的な小窓に川柳が呈示されます。
※囲みの幅に収めるための<script>の後の改行を省けば3行になります。

08 「高齢者向け“脳トレ”言語」としての JavaScript

本冊子で紹介してきた「ブラウザへの指示」は、改行の
と縦書きの<body style=⋯>以外は JavaScript です。

1995年に Brendan Eich (1961-) によって Netscape Navigator 2.0 用のスクリプト言語

(当初の名称は LiveScript) として開発され



た JavaScript は 1996 年には Internet Explorer 3.0 にも搭載され、WWW の一般化と共に急速に普及しました。NN は Firefox に発展的に継承されてもはや存在せず、IE のサポートも 2022 年 6 月で打ち切られましたが、JavaScript は Mozilla などの非営利団体に加え、Google、Microsoft、Apple、Adobe などの企業にも広く支持され、標準化と機能拡張を重ねて ECMAScript 規格を基本仕様とする（「正しく」書くのは面倒な）本格的プログラミング言語に成長しているようです。

実を言うと、筆者は一昔前の JavaScript しか知りません。この言語に興味を持ち、あれこれ調べながら書き始めた 2005 年夏頃※の印象は「ファイル入出力とグラフィックス以外は十分な機能を備えたブラウザ汎用の BASIC」でした。その後、画像なども扱えることを知り、在職中には①など、2015 年の退職後には②・③などで楽しんできました。※「ネット経由で統計処理が可能な data archive の作成」が当初の目標でした。二次分析機能を備えた online data archive の利点とその要件に関する検討 <http://mmua.html.xdomain.jp/kato/study/2006oda.pdf>

①調査報告と統計教育における推定提示の現状とその促進方策

<http://mmua.html.xdomain.jp/kato/study/2011estimation.pdf>

②ブラウザに計算をさせてみよう！（加藤 2018）

<http://mmua.html.xdomain.jp/kato/pdf/keisan@browser.pdf>

③昔の写真でスライドショー（加藤 2019）

http://mmua.html.xdomain.jp/kato/pdf/slide_show.pdf

JavaScript の最大の魅力は何といっても“OS 不問”。「同じプログラムが Windows/Mac/Linux…の全てでそのまま動く」というかつての“見果てぬ夢”はブラウザ上に限れば既に叶っています。加えてエディタのみで書け、web 頁の部品（例：ボタン、入力欄…）とブラウザの機能（例：更新、拡大・縮小…）が使える、ネット上に情報が豊富に存在するといった手軽さ・便利さ・学びやすさもその追加的な魅力です。さらに、昔流の記述のスク립トが最新のブラウザで動く後方互換性（backward compatibility＝新しい環境でも古い指示が有効なこと）、簡単な処理なら短い指示で実現できる簡便性なども嬉しい点です（本冊子の乱数川柳.htm は配列を除けば 11 行、短冊表示.htm は実質 3 行）。

“面白そうなこと”のやり方を考え、とりあえず書いて動かしてみ、その結果からあれこれ調べ工夫する…という高齢者向け“脳トレ”には、Scratch などよりもむしろ向いているように感じます※。
※プログラミングに興味のある方には以下の小冊子が有益かもしれません。

部品型言語を 3 つ試してみました。（加藤 2021）

http://mmua.html.xdomain.jp/kato/pdf/visual_pls.pdf

残念ながら筆者には使用経験がありませんが、昔の Mac には HyperCard（【用語集】参照）という、現在でもその再現が試みられる※ほど優れた情報・開発環境がありました。そして、JavaScript を組み込んだ web 頁には、より手軽にその「代役」となりうる可能性を感じます。それでは皆さん、シルバー川柳も参考にしつつ（？）#、これからの高齢期を楽しく過ごしましょう！
※例：<https://goo.gl/jmqHiY>・HyperCard Stacks@INTERNET ARCHIVE
#シルバー川柳の利用方法については目次下の囲み内の②・③をご参照ください。

