

10-1 素 data 入力用 csv や外部ファイル用配列が作れたら便利です。

なぜなら、既存 data の再利用、(外部ファイル入れ替えによる) 既存 script の汎用的活用などが可能となるからです(後者の例: 加藤 2006、「02: 画像提示」や「03: 金言三題」の汎用化など)。

そこで今回は、表計算ソフトの一行の data からカンマ区切り=CSV の素 data 一行を出力する script (04: 調査への回答を・・の「関連情報」で既提示)、及び加藤 (2006) で使用している外部ファイル用 data 配列を出力する script を作成・解説します。

10-2 表計算ソフトの一行は・・

textarea に paste すると「改行区切り」になります。

改行記号は複数存在する※ため、そのいずれか (|=or) なら全て (g) 半角カンマ (,) に置き換えることでカンマ区切り素 data が得られます。

※Windows(や MS-DOS)では\r\n、UNIX と OS X では\n(OS9 までは\r)。

10-3 表計算ソフトの特定範囲 (r 行 x c 列) を・・

textarea に paste すると「列間はタブ区切り、行間は改行区切り」になります。従って、文字要素を""で括り、タブを、欠測(空白文字の誤入力を含む)を null に置き換えて事例毎に()内に収め、名前と連番を付ければ配列になります。

そこで指定された名前を変数「名」に代入して=new

Array()と結合し、改行(\r\n)で分割した textarea の内容を更にタブ(\t)で分割した要素(素[j])が数値でない(isNaN)なら""で括ります。

欠測及び半/全角の空白のみの要素を文字列 null に置き換えたなら要素を、で結合して行に戻し、名前と連番を付けながら()で括って出力用変数の「配」に結合し終わったら elements[0]に代入(=textarea 出力)します。

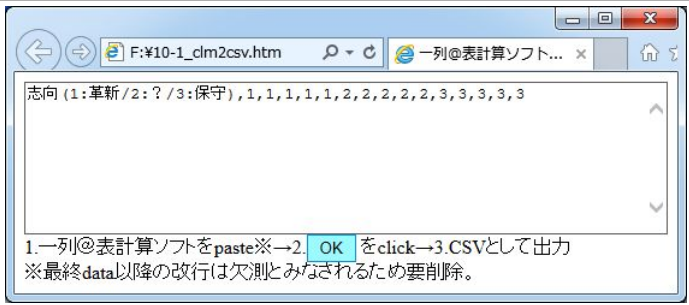
```

<title>一行@表計算ソフトの CSV 化</title><!- saved from url=(0008) about: internet ->
<form>
  <textarea rows="8" cols="64"></textarea><br>
  1. 一行@表計算ソフトを paste※→2. <input type="button" value="OK" onClick="OK()">
  を click→3. CSV として出力<br>※最終 data 以降の改行は欠測とみなされるため要削除。
</form>
<script>
  例="志向(1:革新/2: ? /3: 保守) \r\n1\r\n1\r\n1\r\n1\r\n1\r\n2\r\n2\r\n2\r\n2\r\n2\r\n3\r\n3\r\n3\r\n3\r\n3"
  document.forms[0].elements[0].value=例
  function OK () {
    素=document.forms[0].elements[0].value.replace(/\\n|\\r\\n|\\r/g, ",")
    document.forms[0].elements[0].value=素
  }
</script>

```

文字列中の全て(g)の\r\n,\r\n,\r を", "に置換。
一行にできる内容ですが「幅の節約」のため2行に分割。

資料 23 表計算ソフトの一行からカンマ区切りの素 data 一行を出力する JavaScript の例



資料 24 資料 23 の内容の実行結果

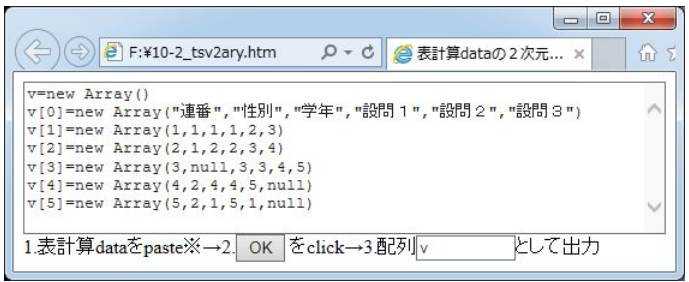
```

<title>表計算 data の 2次元配列化</title><!- saved from url=(0008) about: internet ->
<form>
  <textarea rows="8" cols="64"></textarea><br>
  1. 表計算 data を paste→2. <input type="button" value="OK" onClick="OK()">
  を click→3. 配列<input type="text" size="8" value="v">として出力
</form>
<script>
  例="連番\t 性別\t 学年\t 設問 1\t 設問 2\t 設問 3\r\n1\t1\t1\t1\t1\t2\t2\t3\r\n2\t1\t1\t2\t2\t3\r\n3\t4\t4\t4\t5\r\n4\t2\t4\t4\t5\r\n5\t2\t2\t1\t1\t5\t1\t1"
  document.forms[0].elements[0].value=例+"3\r\n3\t4\t4\t5\r\n4\t2\t2\t4\t4\t5\t5\t  \r\n5\t2\t2\t1\t1\t5\t1\t1"
  function OK () {
    名=document.forms[0].elements[2].value;配=名+"=new Array()"
    行=document.forms[0].elements[0].value.replace(/\\r\\n|\\r/g, "\\n").split("\\n")
    for (i in 行) {素=行[i].split("\\t")
      for (j in 素) {if (isNaN(素[j])===true) {素[j]=""+素[j]+" ";
        if (素[j].replace(/\\s/g, "")=="") 素[j]="null"}
        行[i]=素.join(",");配=配+"\\n"+名+"["+i+"]"+"="+new Array(" "+行[i])+""}
    document.forms[0].elements[0].value=配
  }
</script>

```

気づきにくい「空白文字誤入力」の例。
初期値 v は spda (加藤 2006) の data 配列名。
例 2。
素[j]が数値でない(=文字列)ならその前後に二重引用符(")を結合。
要素(素[j])中の全て(g)の空白文字※(\\s)を""に置換した上で該当要素に文字列"null"を代入。 ※全・半角 space, tab

資料 25 表計算ソフトの r 行 x c 列の data から配列を出力する JavaScript の例



資料 26 資料 25 の内容の実行結果

- ①9 \r\n, \r\ns .. を // で挟む正規表現でパターン指定が可能。
- ②0 配列名.join(記号)は要素を記号で結合して文字列化。

↓ 2006 の論文とその下の例: などを参照。
関連情報: <http://hp.vector.co.jp/authors/VA021211/>