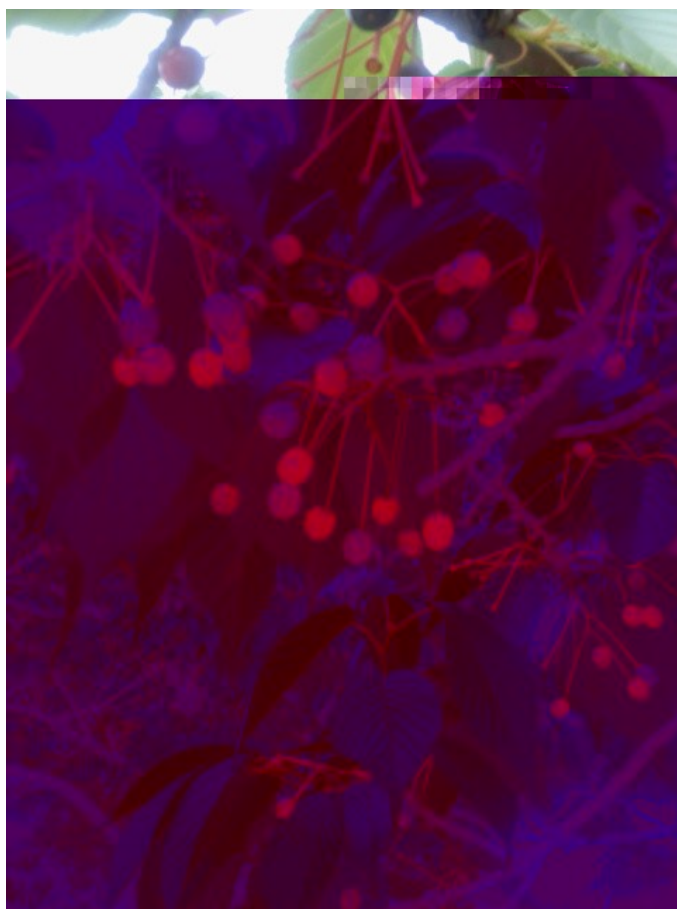


TSNET スクリプト通信



TSC 編集委員会
ISSN 1884-2798

目次

巻頭言	jscripiter ...	3
Python の文法 草稿 第 7 回	機械伯爵 ...	4
HTA でお手軽(?)GUI	ムムリク ...	9
覚醒と進化 試論 III - Compact Topic Maps	jscripiter ...	17
編集後記	jscripiter ...	23

表紙写真: あーかい、さくらんぼ
撮影: jscripiter
日時: 2012 年 5 月 20 日
場所: 宇品波止場公園(広島市)
メモ:

Wikipedia によると、サクラも 400 種もあるとかで、欧米では実を食べるための改良が行われ、日本では花の改良が行われている。突然変異も多いとか・・・サクランボは、その名の通り、サクラの実であるということを、加古川からのツイッター返信で再認識したというお話。

巻頭言

jscripiter

広島も梅雨入り、暑くなった。もちろん、夏だから当然だ。五月末に投稿していただいた二件の記事を抱えたまま、六月に入った。自分自身の記事が形を成していないからだ。

欧州債務危機は解決の兆しはなく、暗雲の中に漂流している。米国のシェールガス革命は回りまわって、日本のガソリンの価格を押し下げはじめた。来週からは、アップルのWWDC2012が始まる。すべての要素は社会の坩堝の中でかき回され相互作用するはずだが、何が起こるのだろうか。

今号では、機械伯爵氏の「Pythonの文法」草稿 第7回、バイト列、バイト配列関連のクラス・メソッドが場。ムムリクさんは、「HTAでお手軽(?)GUI」でActiveScriptRubyでHTA(HTML Application)を作られている。Windows であれ、アプリケーション作成の になる。

はと言え、DesktopWeb ームークの を うすべきかで。なDDL(Diary Description Language)! HTML! RSS"Atom 変# をさらに進化させ、日記の%&' ータ化を(し進めようという)* なのだが、記+ 形、に何を するかというところで、いろいろと- っている。なかなかよいものがない。自然言. に/ いO 1 2 3 形、で、' ータとしての4 り5 い要素を6 7 化しているものという8 9 な要: をするので; < から= ところだろう。今回は、Compact Topic Maps を4 り げて、> い? を@ っ て よう。

A /、Bacebook もCD に> っ て たのだが、Twitter もE めてF G の記H に> うのはI ? だというJ 論にK っている。なL なら、SNS はその名の通り、社会M な場でもあるからだ。F G の記H はN はりO lo (Weblo)に集中させP えるべき。それでなおさら、DesktopWeb がQ というR S T がU かV がる。これをWXYZ とも言う。

(投稿: 2012 年 [月 9 日)

Python の文法

TSNET スクリプト通信版 草稿 第7回

6-2-2-2-3-2-2 バイト列を扱う順序列型

バイト列(bytes)は、1byte\]bitのF Gを のアイ^ムとして5う、 種の_`列aです。

b c MにはPython2までは、このバイト列を5うd クシ ンとしてestrea が>用されてきました。

しかしPython3では、estreクラスはf gに「hi ルj バイトをEk l 文m」を5うd クシ ンになり、nわりにebytse(バイト列)クラスとebytarraye(バイト配列)クラスの二 のaがバイト列o作p用クラスとして 場しました。

バイト列aのメソッ は、9q分が文m列とrじで、 qバイトs t uエクトo作用のメソッ が加えられ、v要なメソッ がwxされています。

バイト配列aは2変aのバイト列で、バイト列のメソッ にy用2変_`列aのリスト(eliste)クラスとzたようなメソッ がいく か{加されています。

|下に、文m列、バイト列、バイト配列、リストのメソッ }を~\$ますhgてが•メソッ €V文m列あるいはリストの が•メソッ は•,してありますl

fメソッ } ..

..method†	str	bytes	list	bytearray
††alloc††				^
††delitem††			^	^
††etnewar s ††	^	^		
††iadd††			^	^
††imul††			^	^
††setitem††			^	^
append			^	^
capitali%e	^	^		^
center	^	^		^
decode		^		^
endswith	^	^		^
eŠpandtabs	^	^		^
eŠtend			^	^
<ind	^	^		^
<romheŠ		^		^
indeŠ	^	^	^	^
insert			^	^
isalnum	^	^		^
isalpha	^	^		^

..method†	str	bytes	list	bytearray
isdi it	^	^		^
islower	^	^		^
isspace	^	^		^
istitle	^	^		^
isupper	^	^		^
join	^	^		^
ljust	^	^		^
lower	^	^		^
lstrip	^	^		^
maketrans	^	^		^
partition	^	^		^
pop			^	^
remove			^	^
replace	^	^		^
reverse			^	^
r<ind	^	^		^
rindeŠ	^	^		^
rjust	^	^		^
rpartition	^	^		^
rsplit	^	^		^
rstrip	^	^		^
split	^	^		^
splitlines	^	^		^
startswith	^	^		^
strip	^	^		^
swapcase	^	^		^
title	^	^		^
translate	^	^		^
upper	^	^		^
% ill	^	^		^

バイト列、バイト配列の☑では、• Ž のメソッ の••の ~ \$ ます。それ| ‘ は、文m列’ リストのメソッ の☑を“ ” してくださいh なお、バイト列’ バイト配列の に• 通するメソッ は、バイト列の☑で••しますl

6-2-2-2-3-2-2-1 バイト列(bytes)

■クラス名 : 'bytes'

– バイト列のシーンスを5うクラスです。(ASCII)文m列のーにebeを加えたリ^ラルで記+h ASCII文m| ‘は、~Šを™た1[Š法表記lする>、ebyteclassに|下のようなs t uエクトをœえてもバイト列は•成できます。

- 0ž 255のŸ がアイ^ムであるリスト、タプル、j c f
- バイト列、€Vバイト配列s t uエクト

さらに、文m列をY ; し、eencodeメソツ でロンドーしても、バイト列は•成できます。

```
>>> b0 = b'abcde'
>>> b0
b'abcde'
>>> b1 = bytes([97,98,99,100,101])
>>> b1
b'abcde'
>>> b2 = 'abcde'.encode()
>>> b2
b'abcde'
```

なおdソール¥の!Šは、ASCII dー が“ ©している文mがある場ªはその文mで、な™れ1[進法表«で表されます。

```
>>> bytes([124,125,126,127,128])
b'|}~\x7f\x80'
```

ただし、文m列をbytesクラスにー- ©しても、バイト列は!来ま\$ん。

```
>>> bytes("abcde")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: string argument without an encoding
```

※UnicodeのUTF-8エンコーディングは、ASCIIコード互換です。

実-のT°としては文m列というより、0ž 255の ±をアイ^ムとするタプルのようなT格のs t uエクトです。

```
>>> b = bytes((x for x in range(10)))
>>> b[0]
0
>>> b[1]
1
>>> b[2]
2
```

ただし、文m列と•通するメソツ は 多く実²されています。

|下に••するメソツ は、バイト列h€Vバイト配列l •Žのメソツ ですh文m列、リストと•通するメソツ にいては、リスト€V文m列の記+を“ ”してください。なお、「バイト列を5う_`列ª」のœに•通メソツ の }を³げてありますl

■文字列に戻す

書式: `b.decode()` ⇒ `s`

動作: バイト列`b`を文字列`s`に戻す。ただし、UTF-8 エンコードで解釈できないコードの場合はエラーが返る

■16 進法表示から、バイト列を作成する

書式: `bytes.fromhex(S)`

動作: 文字列`S`の先頭から数字(0~F)を2文字ずつセットで読み取り、16 進法と解釈してバイト列を生成する

※'fromhex'で読み取る文字列の間にホワイトスペースがあっても構いませんが、2つペアでないとエラーを返します。

つまり、

```
bytes.fromhex("FFFF FF FF")
```

は通りますが、

```
bytes.fromhex("FFFF F F FF")
```

は通しません

```
>>> s = "abcde"
>>> b = s.encode()
>>> b
b'abcde'
>>> list(s)
['a', 'b', 'c', 'd', 'e']
>>> list(b) # 似てるが文字列ではない
[97, 98, 99, 100, 101]
>>> [hex(x) for x in b]
['0x61', '0x62', '0x63', '0x64', '0x65']
>>> bytes.fromhex("6162 636465") # 16 進法で生成
b'abcde'
>>> bytes.fromhex("616 263 6465") # 2 桁ずつになっていないと・・・
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: non-hexadecimal number found in fromhex() arg at position 2
>>> ub = b"ばいそん" # ASCII 文字でないと、リテラルで書けない
File "<stdin>", line 1
SyntaxError: bytes can only contain ASCII literal characters.
>>> ub = "ばいそん".encode() # encode を使えばよい
>>> ub
b'\xe3\x81\xb1\xe3\x81\x84\xe3\x81\x9d\xe3\x82\x93'
>>> ub.decode()
'ばいそん'
>>> ub2=bytes.fromhex("e381b1e38184e3819de38293")
>>> ub2.decode()
'ばいそん'
```

6-2-2-2-3-2-2-2 バイト配列(bytearray)

■クラス名: 'bytearray'

2変数のバイト配列を5クラスです。バイト列を@してくる|'は、バイト列クラスがμ™4
ることの|来るタイプのst uエクトで•成することが|来ますh; %&ですが、ebyteeseクラスにバ
イト列st uエクトを@してもバイト列を•成できますので、ebyteeseクラスとebytearrayeクラスの
μ™4ることの|来るYはgくrじですl

なお、バイト配列st uエクトのリ^ラル€V表«は

bytearray(b'abcde')

のような形になります。

バイト配列のメソッドは、バイト列€Vリストと¶•通ですhバイト列にリストのメソッドを
q加えたものがバイト配列のメソッド•となっています。、しくは「バイト列を5う_`列a」の€
を“”l。

1 自のメソッドは|下に3げるe\$alloc\$teです。

■割り当てられているメモリ容量

書式: ba.__alloc__() ⇒ i

動作: バイト配列オブジェクト“ba”に現在割り当てられているメモリ容量を整数“i”として返す

※バイト配列のアイテムを増減してみると、割り当てられているメモリ容量は必ずしもアイテム数にきっちり
と比例しているわけでないのが分かります。データとして保存するならば、バイト列オブジェクトに変換した
ほうが、メモリ効率が良いことは言うまでもありません

【Tea Break】先行き予測の難しさ ~Python3への移行の現状~

「Pythonの文法」では、Python3(3.0以降)の文法に限り紹介していますが、巷ではなお、Python2系であ
るPython2.7(Python2系では最後になるとアナウンスされながら)が依然として勢力を保っています(卑近な
話で言えば、去年末からandroidスマホに替えた私は、SL4AのPythonが2.7だと知って愕然としました)

その中で、Python3最新のver.3.3に加えられた変更のうちPEP414の「Unicode文字は、また頭にuを付けて
も良い」という変更は、ある意味でPython3側からの譲歩です。とにかく2系から移行してもらわないことには、
Pythonのスタイルが恒久的に分裂してしまうかもしれないという危の表れかもしれません。

これは、2008年末にPython3が表されたには、私が予もしていなかったです。

Python3の文法にまずまずしていた(reduceくらいないか、lambdaはったし)私は、Python2から3
への移行は、くとも2年で成するとっていたのです。

2012年、あれから年以ちましたが、Python2に移行するしはえてきません(そそ移行のス
ードがくなる、という予測はしていますが、はです)

はりのコンイル言!と"なり、動的にコンイル#\$な言!の文法は(%一スコード&'ルでって
いる分)おいそれとは後互換を(てられないようです。

)の言力の*い私としては、Python3の+力を、-に紹介していくに法はありません。

オブジェクト/〇言!としてのスタイルが12に34し、また、5数型プ6グラフingのラ8イムも9:
しすくなくなったPython3は、; <の=せ>?@のPython2系よりABで「使いすい」言!のCです。Python
プ6グラマのDが、EれFしんだスタイルを(てがたくてPython3に変えないとはえません。プ6グラフ
グスタイルの変更はGHにわずかですし、IJKのLMなNOまでよりPQになることはRSはありますが、
TになるU分はそれ以にVきいといます。R分、OまでのWXのLMなのでしYう(それだけに「新しく
Zされた」CのSL4Aが2.7系なのは解せません。)か[\的にLMがあるのだうか])
せ?てOから移へされるZ系には、Python3が_ \となってもらえるとありがたいです。

(投稿: 2012年5月25日)

HTA でお手軽(?)GUI

ムムリク

Re^oIEW というプ» ¼ラムがあります。ここ 年½¾してきた¿ £ÄÄの何Ä目かというt ームもあってÄ年あたりからOÄがÄまっているツールのÆと です。もともと¡ ÇÈ稿をÉÈするために作られてきたÈÌ があるようですが、PDB N EPU0 の• 成も行うことができるようにÍ Î されてきました。

Í 本MにÈ稿は^ Ðスト Ñイルとして用%されます。そこにÒ¡ しであるとか、Ó種リストであるとかÔÕであるとか、さまÖまなÖ記を¹ 自の記法で{ 加し、Í × となるÈ稿^ Ðスト Ñイルを作成します。

この^ Ðスト Ñイルがあれば、あとは のような形にØ げたいのかをÛÛするだ™でÛ の形の Ñイルが• 成されます。

しかし、これらのタ¼ ™をÛしく行っていくのはなかなかÝP です。t » ¼の編集B Ýのようにいろいろのタ¼がä にä Ûできたらä ではなかろうか、ということでプ» ¼ラムを作っていました。

HTAhHTML Applicationl なのでÍ 本Mに Windows ä Ûです。実行には ASRhActive Script Rubyl がä 要です。

HTA はÍ 本Mにはä æt æ一u の HTML とr じです。ボタンN^ Ðスト表« のç な をè に イアä トできます。

それéれのボタンな を押したときのè< をè のスクリプト言. で記+ できますが、ここでは Ruby で記+ します。手軽に GUI なプ» ¼ラムを作れるというë でもなかなかí í です。

î ä 要なもの

Ruby 1.9.Š hhttp:""www.artonŠ.or "data"asr"l

ASR がä 要なのでこちらのバイī リðッ 一u をインストールしてください。

î > いR

ここではñくä なのからはじめて ます。機ðとしてはこんなó じです。

– ô ÑイルをÛÛして く。

õ ô 記法にõったp 用タ¼を ™る。

÷ ô Ñイルをøùする。

edit.hta のアイdンをú t ルクリックして実行します。Window が きます。「 ” 」ボタンを押して編集する^ Ðスト Ñイルを し、ú 当する文md一 をj ェックして、「3 ü 」ボタンを押します。^ Ðストが表« されます。



タグを解除する場合^aにはタグをE k q 分をi a スで し、「タグ解除」ボタンを押します。

編集が完了したら「保存」ボタンを押します。ファイル名は元のファイル名に -edit を付加したものに なります。例 sample.txt であれば、sample-edit.txt

1 プログラムについて

本体は edit.hta をご覧ください。冒頭の head 部分 script 6 に Ruby のプログラムを記します。続く body 6 には通常の HTML の様に a e t a e u のイメージで HTML タグを記します。なお、この プログラムでは Re IEW とは異なるタグの様です。

ファイルインポートな それぞれの q 分には異なる id を付与しておきます。

ファイル名の取得は、

```
`input type="file" id="path" size="60"
```

です。

```
`input type="button" id="do" value="読み込み" onclick="load"
```

こちらは3 番 ボタン。ボタンがクリックされると、onclick で指定された load にカーソルが移ります。ボタンに入力されたデータのそれぞれのアクセスは、

```
findog.document.all("id名")
```

もしくは、

```
findog.id名
```

でアクセスできます。その数は、value で4 通りあります。

load では、データが指定されていて その ファイルを いてすべて3 番 、それを表すの

te\$starea の± に代入して表« さ\$ています。

文mdー のラus ボタンは de< encode; end q 分でê< します。ラus ボタンの± は配列で返されるので、j エックされているか うかをÒて、j エックされている± を @encode に代入します。

^ Ðストαリアに表« するに文m列にたいして encode("Windows-31J") としていますが、これは現7のところ ASR のØ様 UTB-] な に" ©していないためです。よって UTB-] な に固Žの文m を> 用している場ª にはÛしくê< することができま\$ん。これはøù に関してもr じです。

de< ta s; end はタ¼の9 きな種p 分™のリストを表« するq 分です。ここで した種p に©じ て、細なリストを <"span> のなかに表« さ\$ます。

その、細なタ¼のリストに©じたê< が、de< level; end N de< note; end です。level ではÒ | しを、note ではÒÒをê< します。

タ¼解x ボタンが押されたときには、de< o<<; end が呼 れます。実に解x しているのは de< ta o<<; end q 分です。

te\$starea のなかの された文m列を4 得しているのは、

s a fgindog.editarea.document.selection.createhande()

のq 分です。

Ī 本Mなê< はこれだ™です。あとはそれé れのタ¼ñ とのリストとそれé れにそったê< を施す q 分を作ることさまÖまなタ¼をà に1 違いなくá Ū・解x することができるようになります。

ĭ UTB-] ¥の暫Ū" ©

うしても UTB-] を> わÖるを得ないときには、α' イタな で Ñイルを いておいてからg 体をd ピーして、- 編集用のαリアに貼りý™ます。øù の時には逆に編集αリアの^ Ðストg 体をd ピーしてα' イタな に貼りý™てからα' イタ側でøù します。

ĭ まとめ

HTA で作るうえでの悩 はF Gがあまりないということかもしれま\$ん。今回もずいぶんと検索なして調べたのですが、 文m列F Gな を う4 得するのかになかなか! 会えま\$んでした。g 般MにF Gは少なめです。i イク» ソ のサイトですら今ではあまりÒ かりま\$ん。

イアã トに いても css で調Ÿしていくにはなかなか慣れがã 要なようで、あまりスi ートなものではないかもしれま\$ん。そのあたりをあまりÄにしな™れ、手軽に くれるというのはあると思います。

今となつてはNNFG も少なめではありますが、ちょっとしたプ» ¼ラムではある™れ GUI にしたい。できるだ™{ 加でインストールな v 要なほうがよい。という時には案' > えるかもしれま\$ん。

i edit.htaj

```

`htmlc
`headc
`meta http-ekuiva`content-type` contenta`textlhtmlm charsetanindogs-31o` lc
`titlecEplT.qTA`ltitlec
`script landuadea`hubyScript`c
LErEL_LIST a sti`select namea`headlevel` onchandea`level`c
  `option valuea`0` selectedcなし
  `option valuea`1`c&' ルu
  `option valuea`2`c&' ル2
  `option valuea`3`c&' ル
  `option valuea`4`c&' ルv
  `option valuea`w`c&' ルx
  `option valuea`6`c&' ルy
`lselectcj
NzTE_LIST a sti`select namea`note` onchandea`note`c
  `option valuea`0` selectedcなし
  `option valuea`1`c{ 釈| }
  `option valuea`2`c{ 釈
  `option valuea`3`c~ 作• { (ブラック)
  `option valuea`4`c~ 作• { (インライン)
`lselectcj

def load
  encode()
  edit a fgindog.editarea
  ff a fgindog.document.all("path").value
  unless "" aa ff
    lines a ""
    File.open(ff€ "r•,ffencode„") f...file...
    file.each_line f...line...
    lines ta line
  "
  "
  edit.value a lines.encode("nindogs-31o")
else
  fgindog.alert "㊦ イルを / 4 してください"
end
end

def save
  lines a fgindog.editarea.value.split(l%nl)
  exname a File.exname(ff)
  basename a File.basename(ff€ exname)
  fname a ",fbasename„-edit,fexname„"
  p a File.dirname(File.expand_path(ff))

```

```

    pir.chdir(p)
    File.open(fname€ "g") f...f...
        lines.each f...line...
            f.print line
    "
"
end

def encode
    fencode a ""
    enc_chk a fgindog.document.all("encode")
    enc_chk.each f...enc... fencode a enc.value if enc.checked,
end

def tads
    tad_dloup a fgindog.document.all('tads').value
    span a fgindog.tadlist
    case tad_dloup
    when '0'
        select a ""
    when '1'
        select a LErEL_LIST
    when '2'
        select a NzTE_LIST
    end
    span.innerqTSL a select
end

def level
    s a fgindog.editarea.document.selection.createhande()
    l a fgindog.document.all('headlevel').value
    unless "" aa s.text
        if "0" aa l
            tadoff(s)
        else
            level a "a" < l.to_i
            s.text a ",flevel_", fs.text,
        end
    else
        fgindog.alert "文字列が正・されていません"
    end
end

def note
    s a fgindog.editarea.document.selection.createhande()
    n a fgindog.document.all('note').value
    unless "" aa s.text
        case n
        when '0'

```

```

        tadoff(s)
    ghen '1'
        s.text a "f`fncf€,fs.text,, "
    ghen '2'
        s.text a "||footnoteijf%n,fs.text,,%n|| "
    ghen '3'
        s.text a "||prodnoteiojf%n,fs.text,,%n|| "
    ghen '4'
        s.text a "f`pnciojf,fs.text,, "
    end
else
    fgindog.alert "文字列が正・されていません"
end
end

def off
    s a fgindog.editarea.document.selection.createhande()
    unless "" aa s.text
        tadoff(s)
    else
        fgindog.alert "文字列が正・されていません"
    end
end

def tadoff(select)
    t a select.text
    if lf`fncfiŽ€j<€(iŽ,j†)„l a• t
        t.sub•(''€ '1)
    elsif lf`pnc%i orj%j f(iŽ,j†)„l a• t
        t.sub•(''€ '1)
    else
        t.sub•(lat%stl€ "")
        t.dsub•(sr•%Alia-bjia-bj†.<("•%n...%b)•€ "")
        t.dsub•(sr•%nll„%b•€ "")
    end
    select.text a t
end

`lscriptc
`style typea"textlcss"c
`•—
body ffont-sibe• 1.1emm„
,path ffont-sibe• 0.9emm„
,box fmardin-top• 0.wemm gidth• 41emm„
,encodes fgidth• 2wemmfloat• leftm„
,load fgidth• 4emmfloat• leftm„
,save fmardin-left• 0.4emm gidth• 4emm float• leftm„
,box2 fgidth• 42emmardin• 10 10 10 10m„
,select fgidth• 2wemmfloat• leftm„

```

```

, tadoff fwidth• wemfloat• leftm,
, text ffloat• clearm,
, do€, save€, tadoff ffont-sibe• 1emm,
textarea€ select ffont-sibe• 1emm,
--c
`lstylec
`lheadc
`bodyc
`divc
± `イル名: `input typea"file" ida"path" sibe60c
`ldivc
`div ida"box"c
  `div ida"encodes"c
文字コード: `input typea"radio" namea"encode" valuea"UTF-8" checkedcUTF-8
  `input typea"radio" namea"encode" valuea"nindogs-31o" cShift-oIS
  `input typea"radio" namea"encode" valuea"EUC-oP" cEUC-oP
  `ldivc
  `div ida"load"c
    `input typea"button" ida"do" valuea"読み込み" onclicka"load"c
  `ldivc
  `div ida"save"c
    `input typea"button" ida"save" valuea"保存" onclicka"save"c
  `ldivc
`ldivc

`div ida"box2"c
  `div ida"select"c
タグ: `select namea"tads" onchanda"tads"c
      `option valuea"0" selectedcタグの" • を正•
      `option valuea"1" c し
      `option valuea"2" c{ 釈
  `lselectc
  `span ida"tadlist" c`lspanc
`ldivc
`div ida"tadoff"c
  `input typea"button" ida"tadoff" valuea"タグ解—" onclicka"off"c
`ldivc
`ldivc

`div ida"text"c
  `textarea rogsa"20" colsa"84" grapa"soft" ida"editarea" c`ltextareac
`ldivc
`lbodyc
`lhtmlc

```

(投稿: 2012 年 5 月 2[日])

覚醒と進化 試論III - Compact Topic Maps

jscripiter

1. 更新日記rewrite 計B

いよいよと言うか、いN、今更と言うかŽŽm)、待て も待て も(実は! てくることをDの片隅では恐れているのだが) の眼鏡に叶う究極のアプリは 場しないので、J 局、自分で作るということになる。Ø様を « \$ よと言われそうだが、á 計図は断片Mであり、ある%&既に断片Mにù 7しているものであり、新しい要素はWeb の発 とともに 場し ある。

' スクトップには、文ÀN写真、音â・αAj エックな のメ' イアの Ñイル管<シス^ムが »ーカルのâ ェt NiTunesに ながっている。更新 日記の! Çシス^ムはZedと»ーカルのâ ェt で動作している。その>、メモカ ンúーと 一緒に' スクトップにメモるシス^ムがあるがほとん > わない。

Ú期Mに確実に動いているのは、更新日記の! Çシス^ムであり、αAj エックのPodcast 配信サーバーである。その>、Aノでは、Twitter ¥の日記のカ^ゴリとタイトルとリンクの配信を再 し ている。

Webの新しい要素としては、Evernote N DropboŠな のクラâ サービスがあるが、モバイル環境を > うことはほとん ないこととd ストをE めて記憶容量にã 界があることは• 白なので、今のところ、 Ñイル・Ž がä 要な時だ™のã ÚMなí 用に留まっている。Evernote ¥のノートの自動Ã きù は Aノ試して たところだが、これも Evernote の> 用がメモMなものに留まっている|、今のところ ÕS するä 然Tに乏しい。

- [日記st u エクトブ» トd ル入門](#) (03"04"2012)

R、家庭6で日常Mに> 用するd ンピュー^ イン¼・' バイスが増え、iPod touch"iPad から、Windows の' スクトップ(»ーカル)サーバーにアクセスするというニーズが! てきた。DHCPによる IP ア スを4 得するR 法にはこれまで検討してきたようにã 界がある。Ã 年末に知った Pa ekite(=townCloud)をí 用すれ Ý 白いことにÃ づいていたのだが、Ubuntu のアップ¼ー があつたりして、そのままになっていた。

- [Pa ekite と ownCloud](#) (2011"12"03)

話は少し変わるが、Aノ、柴Xさんの drive-cli の記事をÒて、LinuŠで動かして るかと思い立ち、次いでにeban さんのRSS で 知ったUbuntu の12.04のアップ¼ー を試 た。確かめて ると、—には標x ではインストールされていなかったownCloudが3にアップ ' ートされていた。t ラâ ザから pa ekiteには繋がらなかったの、A初から、curl でインストールするところから始めた。アカã ントは残っていたようだが、—のð スー も記Hできていなかったの、変更してトライ。当然のことながら、これを> え、モバイルをE めて、様々な' バイスからアクセスできるようにすることも2ðだ。実、Windows から Ñイルをアップ»ー して、iPad な からアクセスできる。

pa ekiteは、ý Oí 用(individuals)は; 料か、自分の払いたいだ™払ってくださいということになっている。できれ \$3"月 払ってくださいということ。本格Mにí 用するようになれ 払っても良いかな。> のí 用は\$5.99"月。様々なý O用のクラâ サービスが増えているので、ä 要Tが こまであるかというó じはするが、自—のサーバーがx 備できれ いろいろと©用は2ðだろう。

サーバー環境はともかく、表題のリライト計B。Topic Maps の概念を導入することによって、日記の%&Mな) 造化を図ろうというのが目Mにある。サーバーの移転の問題もあるが、日記の表« も% &Mな) 造に依ù することになる。

現7の記事の作成はZed の手作りMな要素が多いのだが、次' ータを元に自動化したいというニーズがある。もう はタ¼ý™による分p の強化。記事の%&Mな関連ý™の新しいR 法をP え

ること。そして、記事に関連する新しい記事の作成方法を提示している。手作り的な要素と記事に関連する要素を連動させること。

回顧 2011 年、2012 年 - アップル編の記事によれば、音楽配信の動向 - 新しいサービスの世界は どのように変わっているか。

・・・iTunes のようなソフトウェア iPod のようなハードウェアを扱う音楽配信システムは、音楽サービスにメタデータ関連のデータ等を加えて、如何に実体のあるもののように扱うことができるかが重要である。これもデータを量に集積し、本体とデータを通しよく取り扱う問題に通ずる。いずれは、ハードウェア自体をコンピュータに代わって音楽サービスを提供するサービスをベース化し、データを PC で閲覧しながらサービスを制御したいものだと思う。

に必要な日記システムは、各種のデバイスと iTunes などの Web アプリケーション、各種のサイトや Web サービスを統合するものだ。サービスアプリケーションやサービスを糊でつなぐことで、デバイスとして動作させるのが、スクリプト言語なのである。

2. Compact Topic Maps

CTM (Compact Topic Maps) のように厳密には正確でなく、正確でないことも恐れず、* 像の翼を上げて、更新日記を CTM で記述する方法を提示しよう。トピック (topic)、連*・連 (association) と事象 (occurrences) という主要な三要素で日記はどのように表現できるか。

更新日記を作成する要素を列挙しよう。

- ・カテゴリ (カテゴリ年 1 記事リストのリンク)
- ・タイトル (記事のアンカー、アンカーの URL はカテゴリと記事作成時の時刻を指定)
- ・概要 (A 初の概要 RSS 1.0 の description に用いられる。Web 更新日記の 1 記事のリンクを指定)
- ・リスト (リンクを指定場所がある)
- ・表 (リンクを指定場所がある)
- ・写真、図などの画像 (画像サービスの URL とタイトル、... のセンスを指定)
- ・更新日時

日記記事の topic として取り上げられるのは、まずは記事のタイトルだろう。[月 2 日の記事を 4 取り上げてみよう。

[2012 (Sat.)

[「アップル」ソフトウェア Windows 7](#)]

スティーブ・ジョブズ氏インタビュー：発責任者に聞く Windows 7 の世界——「2 年後、タッチできない PC は欠陥品に思われる」 (1/3) - ITmedia + PC USER ネット。9 体の疑問に答えている・・・これはよい記事だね。

Windows 7 と Windows RT (ARM 版) には Windows として根本的に差はないということがはっきりした。デスクトップも 2 年後にはタッチすることになると予想している。

これらのことが示していることは、PC だけでなく、スマートフォンにも新たな費用が発生することや、既に懸念したように、Windows RT バイオスは iOS タブレット、iPad より

もA価になることは1 違いないし、消費とS もO くなるだろう。d ストか、O オーi ンスかとい
う に・ちü めるか うかが勝敗の分かれ目かな。

- [Microsoft Windows Developer Days : Windows Store と Windows 10 の潜7 M2 ò T を日本で紹介 \(1"2\) - ITmedia +D PC USER](#)
- [本X雅 のク» スーバー' u タル : 解像A の呪縛から逃れようとする Windows 10 \(1"2\) - ITmedia +D PC USER](#)

アップルの Mountain Lion のR S T が影響をμ™ るのか うか、WWDC2012 の成り行きがO 変ã し
だ。

更新: 2012-0[-02T11:30:43+09:00

HTML は次のような記+ になっている。

```
`pTc6l2l2012 (Sat.)`ppc
`div class="emph" c`A qhEfa`renegal_index.html,ms`ci マイク6%± トj`lAc `A
NA$Ea`ms_1338w99791`cnindogs 8`lAc`ldivc
`pc`a href="http://plusd.itmedia.co.jp/lpcuser/articles/1204126lnegs090.html" c`スティーブン・
―`± ス™―š インタ> œ―:・ `ž Ÿ・に `く nindogs 8 のj `¢ £ £ 「2 年後、タッx できない PC
は¥ | S に `われる」 (113) - ITmedia tp PC USEh`lAc`タ。V © のª L に« `えてくれている・・・
これはよい― `だ-。`lpc
`pcnindogs 8 と nindogs hT (Ah$ 版) には nindogs として _ \ 的に® はないということがはっきりした。
デスクトップも 2 年後にはタッx して使うことになると `4 している。`lpc
`pcこれらのことが意味していることは、PC だけでなく、ディスプ&i にも新たな`。が `生すること
を意味している。また、以±² に³ ` したように、nindogs hT デバイスは izS タブ&ット、iPad より
もμ¶ になることは間・ いないし、`´¹ カもV くなるだ う。コストか、`± °ーマンスかとい
う¢・ に» `ちe? るかN うかが¼½ の分かれ¾ かな。`lpc
`ulc
`lic`a href="http://plusd.itmedia.co.jp/lpcuser/articles/120412wlnegs02w.html" c`$icrosoft
nindogs peveloper pays : nindogs Store と nindogs 8 の¿ 在的# $ Å をÅ \ で紹介 (112) - ITmedia
tp PC USEh`lAc
`lic`a href="http://plusd.itmedia.co.jp/lpcuser/articles/1204112lnegs074.html" c` \ Å Å Å のク
ラスオーバーデジタル : 解Å 在のÇ È からÉ れようとする nindogs 8 (112) - ITmedia tp PC
USEh`lAc
`lulc
`pcアップルの Sountain Lion の `O Å がÊ Ê を`けるのかN うか、nnpC2012 の成り行きがV 変T しみ
だ。`lpc
`div class="updated" c`更新・ 2012-06-02T11:30:43+09:00`ldivc
```

CTM でÅ くと、例え `次のようになる。

```
def reference-link('title€ 'uri)
  title• 'titlem
  uri• 'uri.
end
```

http://homepdae1.nifty.com/kabufirenegal.html,ms_1338w99791

title• "nindogs 8"m

catedory• "マイクロ%ト"m

date• 2012-06-02m

description• reference-link(title• "スティーブン・ー ス™ー インタ> œー : • ー・
に く nindogs 8 の 2 年後、タッチできないPCは¥に われる」 (113) - ITmedia tp
PC USEh€ uri• http://plusd.itmedia.co.jp/pcuser/articles/1204/26/1negs090.html)m"タ。V©
の L に« えてくれている・・・これはよいだ-。"m

paradraph• "nindogs 8 と nindogs hT (AhS 版) には nindogs として 的に®はないということが
はっきりした。デスクトップも 2 年後にはタッチして使うことになる 4 している。"m"これらのこ
とが意味していることは、PC だけでなく、ディスプレイにも新たな が 生することを意味してい
る。また、以 ± 2 に 3 したように、nindogs hT デバイスは izS タブ&ット、iPad よりも μ になる
ことは間 いないし、 力も V きくなるだ う。コストか、 ー マンスかという E に» ち
e? るか N うかが 1/4 1/2 の分かれ 3/4 かな。"m"アップルの Sountain Lion の OÄ が E E を けるのか N
うか、nnpC2012 の 成り行きが V 変 T しみだ。"m

list• reference-link(title• "Sicrosoft nindogs peveloper pays : nindogs Store と nindogs 8
の 在的# \$Ä を Ä で紹介 (112) - ITmedia tp PC USEh"€ uri•

http://plusd.itmedia.co.jp/pcuser/articles/1204/12/1negs02w.html)mreference-link(title• "Ä
ÄÄÄ の クラスオーバーデジタル : 解ÄÆ の ÇÈ から E れようとする nindogs 8 (112) - ITmedia tp
PC USEh"€ uri• http://plusd.itmedia.co.jp/pcuser/articles/1204/12/1negs074.html)m生返せく」以.加i"恒・.u

updated• 2012-06-02T11:30:43+09:00.

日記記事を、| のような記+ に変# することに れほ の実用Mな%&があるかはまだ疑問なë
がある。I 本Mには、HTML の記+ から機械Mに容易に3 4 れる程Ä のものでしかないからだ。その
ような次第で、なかなか を一に | S ないのが実F である。* 像S の翼が閉じたままだ。

本稿をÄ くにあたり、Compact Topic Maps の Ø 様に いて“ P にしたのは、次のリンクにある文Ä
である。

ISZl IEC 132w0-6• 2010€ Information technolody í Topic Šaps í Part 6•

編集後記

jscripiter

ようNく編集後記に辿り着いた。

編集から話は変わるが、今日は、J 果としてCAT°の二台のj ューī一をHDMI - 続の新しいものに4りn えることになり、CAT°Ē 由のメタルプラスゑ 話の契約を{ 加することになった。

本来、CAT° 契約を二台を 一台にしてd ストúā ンし、さらにiLink ° ©のj ューī一にQりn えてnasne 導入に備えようとしていたのだが・・・実は関係者ŽŽm) が地 波' u タルだ™でなく 0S もÒていたことが工事後にわかって、連絡したところ、再契約で手 料がā 要になることがわかって、がっくり。j ューī一がないと地 波' u タルしかÒれないよと急押ししていたのに、CAT° ° 自のj ャネルがÒれないだ™との思いū 。ØRがない。iLink にはnasne が° ©していないこともわかって二重にがっくり。すべて; 料だったはずなのに、契約を元に戻して再契約料がā 要とは・・・とはほ

CAT° の営業から契約解x のÐキャンセルという4り5いにしようとのゑ 話があって、工事のO から紹介のあったメタルプラスゑ 話が話題になった。インターネットĒ 由のゑ 話は、家6から相手にこちらの声が聞こえないというク ームがýいて困っていたので、Qりn えることに。NTTよりは安いし、^ ビ契約とセットでほんのわずかに割Yもある。ゑ 話の置き場所もリビン¼に戻\$る。

iLinkはソニーは積極Mに4り5っていないらしい。j ューī一はÐī ソニック。このようなメーカー1の互# T問題はユーザーのī 益にならない。自由に機器pを組 a わ\$て> うことができないのは9いに困るのである。ビu ネス はd ストN販売戦、 の問題があるのだろう™、もっとゝのところ9いに競って欲しいものだ。

それはともかく、さて、暑い夏が来る。皆様、お体にはおÄを °™て。

(投稿：2012年[月9日)

Ũ Ũ ĐÑŨ スクリプト通信

ISSN 1884-2798 版, • ÒÓÒ

2012 年 6 Ò 9 Á 4.4.001 版

2012 年 6 Ò 14 ÁÒ行 4.4.002 版

× 稿ÒŨ

[TSNETniki](#) • 「[× 稿ÒŨ](#)」のページをŨŨのこと

Ũ>ÝƧƧ(× 稿順)

à á â kikgail atj livedoori dotj com
ムムリク kubli laboi atj dmaili dotj com
jscripiter jscripiter9i atj dmaili dotj com

ã 作ä

1. ã Ñ æÇ その. のã 作èについては、ã 作• がã 作ä を保» します。
2. 「TSNET スクリプト通信」のé è ã 作ä はã Ñ æÇ その. のã 作è のã 作• より構成される Ũ>ÝƧƧが保» します。

使° èì • 配í î ï

1. Ũ>ÝƧƧは「TSNET スクリプト通信 4.4. xxx 版」を、± ^ イル名が「tsc_4.4. xxx. pdf」の PpF ± ^ イルとして* Òで配í します。また、± ^ イル名、± ^ イルñ 容をÄ Ò ó 変しない状での¹ K 的ò 配í およÇ Ò Ò によるò 配í を* Òでèì します。
2. 5 ÷ するスクリプト± ^ イルなNのプらグラムについては、使° およÇ Ò 配í を* Òでèì しますが、ó 変後のò 配í についてはオリジナルのã 作ä をø Ñ することをî ï に* Òでèì します。
3. Ñ およÇ スクリプト± ^ イルなNのプらグラムにã 作• の使° èì • 配í î ï のÑ ù がある場合は、ã 作ä のú およÇ Ñ 2 ú にú 先するものとします。

Ũž Ũ

「TSNET スクリプト通信」のñ 容およÇ ý Ƨに配í されるスクリプトなNの使° は、すý て使° • の自己ž Ý によるものとし、使° によって生ずるÄ Ò の結果等について、Ũ>ÝƧƧおよÇ ã 作• はž Ý を負いません。

Ũ>%± トウェア

zpenzffice. ord 3.2.1 nriter

行所

Ä è 配í 所• TSNET スクリプト通信Ò行リスト

[http://ltext.gor ld. coocan. jpl TSNETI "TSNETsE3s82sB9sE3s82sAFsE3s83sAA](http://ltext.gor ld. coocan. jpl TSNETI)

[sE3s83s97sE3s83s88sE9s80s9AsE4sBFsA1sEws88s8AsE8sA1s8CsE3s83sAAsE3s82sB9sE3s83s88](#)

