

株式会社 ITS MORE

2020年4月設立

ITS more

2020年8月17日 投稿者: SATOXITS

HTMLソース印刷用ブラウザ選定

社長：開発したGo言語埋め込みHTMLテキストの印刷用ブラウザを選定したいと思います。

基盤：ページソースの表示もそうですが印刷となると各ブラウザ個性がありますね。

社長：想定しているフローはこうです。

1. G1.goをGo言語プログラムとして実行して動作を確認する
2. G1.goをG1.htmlとしてブラウザでHTMLとして表示して動作確認
3. G1.htmlのソースをブラウザで表示してPDFファイルG1.pdfに印刷
4. G1.pdfに電子署名してアップロードする
5. G1.pdfをダウンロードして電子署名を検証する
6. G1.pdfからテキスト情報をG2.htmlに取り出す
7. G2.htmlをブラウザで表示して動作を確認する
8. G2.htmlをG2.goとしてGo言語プログラムとして実行して確認する

開発：理想的には全部のステップを1つのツール、まあ第一候補はブラウザですが、一つのツールで全部できると良いですね。現実には、編集は vi (vim)、HTML表示はブラウザ、署名はAcrobat、アップロードには Finderからドラッグを使っているわけです。

社長：3,4と5,6 を自動化してワンクリックにしたいんですけどね。GShellでできるといういな。

開発：問題は、G1.go == G2.go、G1.html == G2.html が維持されるかですね。インデント情報が壊れたり、変数名の途中で改行されちゃったりとかよくあります。

基盤：Acrobat とエクスポートだなんだとか格闘したく無いんですよ。Acrobat での電子署名にこだわらなければ、物事はすごくシンプルだと思うんですが。ただG1.html.gz くらいをアップロード・ダウンロードすれば済むと思います。

開発：電子署名にこだわるなら、PGPとかS/MIMEという選択肢もあります。

社長：まあそれは… まず第一に、PDF型式は電子署名のインフラとして独走状態にあるのが現状です。クリアテキストに電子署名というのは、プログラムを印刷物として眺めるのに都合が良いわけです。そのプログラムは誰が何時作成したものか、印刷という表示型式で著者が確認した見た目の同一性も含めて、電子署名で確実に検証・保証できる。今もよくある、ファイルと別に同一性確認用のハッシュ値を提供するとか、もうそういう時代じゃないと思うんです。

開発：まあ、印刷された型式に著者として納得ができるかどうかはまた別ですけどね。これについては、ブラウザごとに非常に個性があることがわかっています。

社長：第二に、当社はすでにAcrobatを1年間約2万円でサブスクリプション契約をしており、その元はまだ取れていない。まだブログ関係で署名し直しも含めて約500回、その他の用途も含めて1000回も署名してないと思います。

経理：Acrobat は 4/26 日に 20,856円で購入しています。現在まで4ヶ月の使用料を7000円分相当と考えると、一回7円。妥当な線かとは思いますが so far。

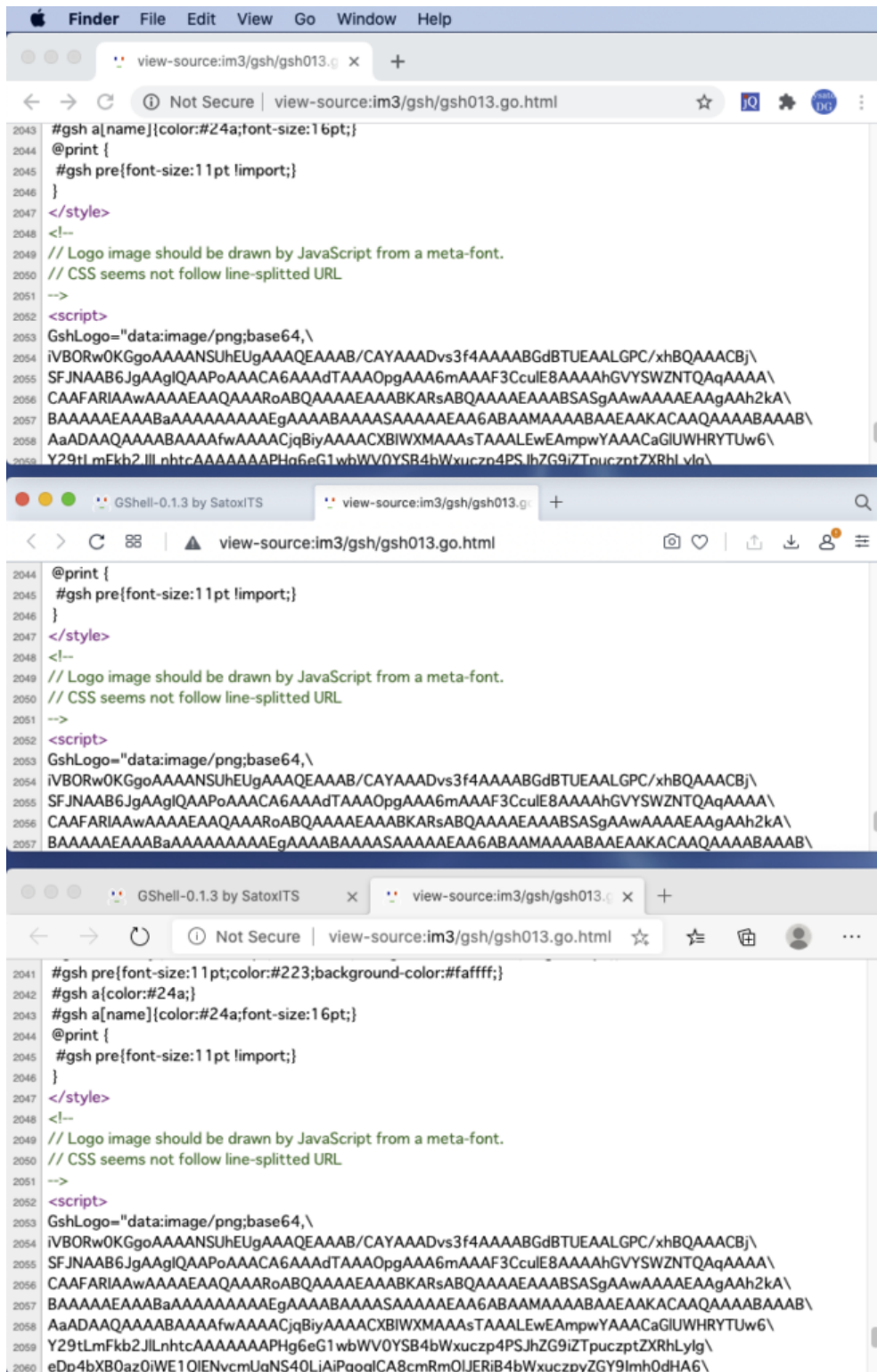
開発：現時点まで署名1回あたり20円と考えても、結構元が取れたように思われますが。署名だけに使用しているわけでも無く。時々OCR機能とか変換機能とか使いますよね。

基盤：Acrobat をコマンドモードで使うなり、マウスとキーボードの操作を自動生成するなりして、署名をバンバンさせれば良いと思いますけどね。クソ遅いから10秒くらいかかるかもしれませんが。それでも1日で8000署名できますよ。

開発：それやったら、無償で使わせてもらっている公証タイムスタンプ局から怒られるかもね。

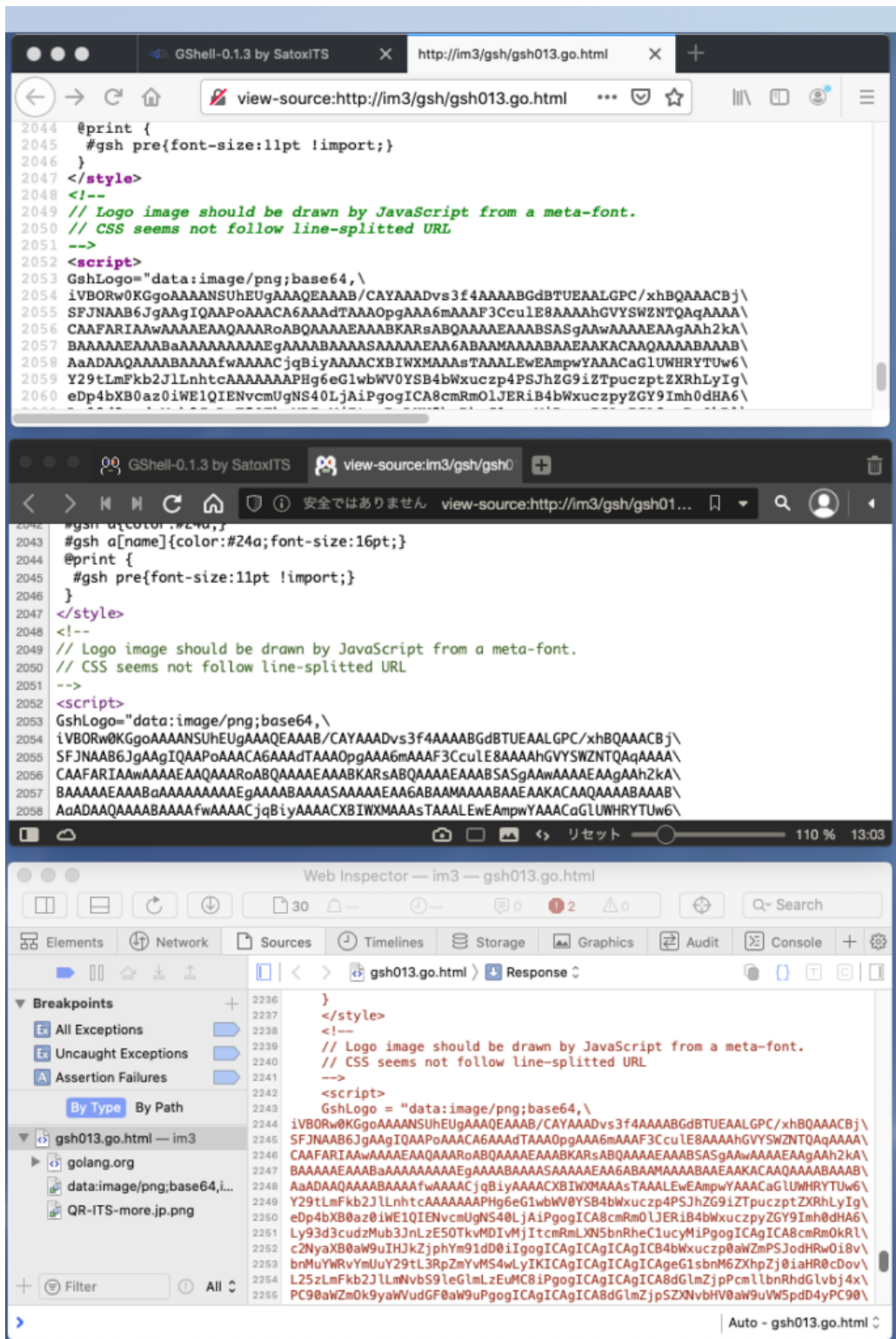
社長：それで、一応上のフローで考えるとすると、HTMLのソースを表示してPDFにする、というところが第一関門です。

開発：それで各ブラウザでやってみたのですが、やはりとてもそれぞれ個性が強い。まずダメ組トリオ。



開発：上から Chrome、Opera、Edge ですが、ソースを表示するのに固定ピッチフォントで無い時点で終わっています。インデント情報も適当に生成している。使えません。

開発：次に見込みのあるトリオ。



開発：上から Firefox、Vivaldi, Safariです。

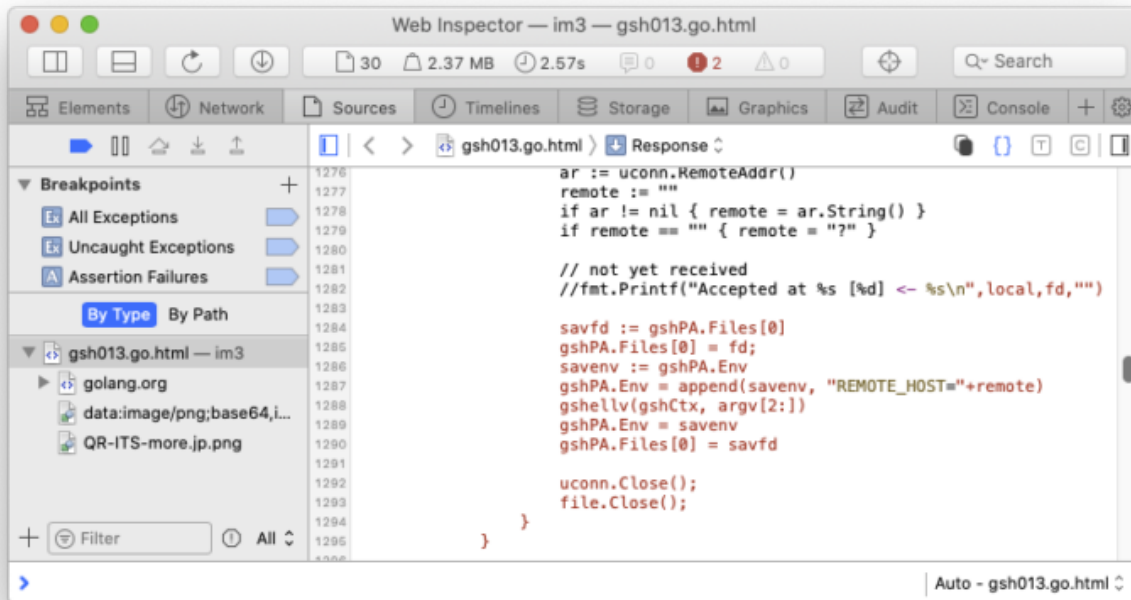
社長：いつも思うんですが、Firefox の色使いは只者でないというか、常人とは色彩モデルが違うのを感じますね。

基盤：開発者がすごく特殊な発色をするディスプレイを使っているのかも。開発費が無くて90年代のとか。

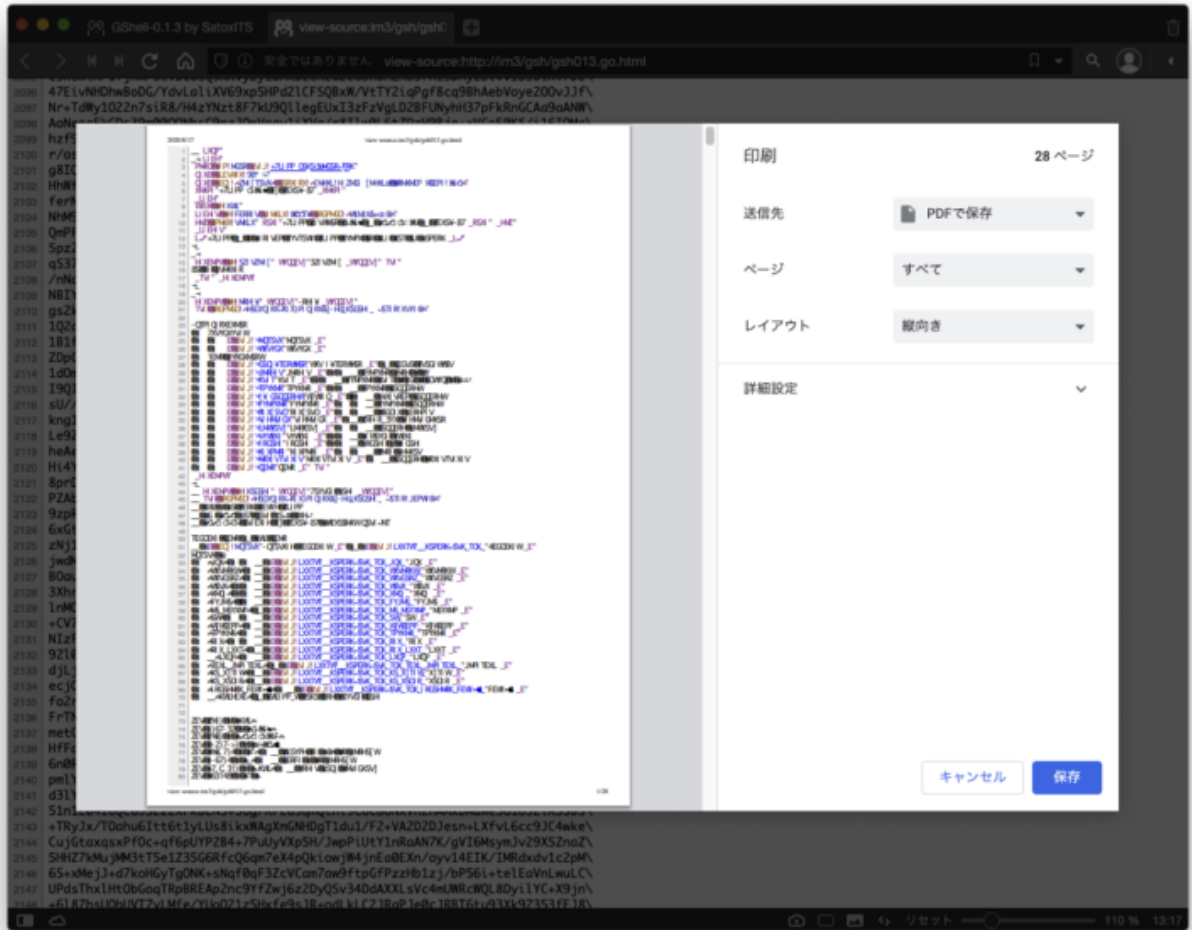
経理：電気代がかさむんじゃないですかね。

社長：でもFirefoxには世界中の色々な人が開発に参加してるんですよね。Mozilla MDNのウェブページにもただならぬ色彩が漂ってるので、何かそういうカッコとした色彩ポリシーのようなものがあるようにも思います。

開発：さて、この3者のうち、SafariはHTMLの中にあるGoの特殊記号省略記法に着いて来れないようなので脱落。



開発：で、あとは印刷対決なんですが、なーんと、Vivaldiは人間非可読なPDFをしれっと生成して来ます。



2020/8/17

view-source:im3/gsh/gsh013.go.html

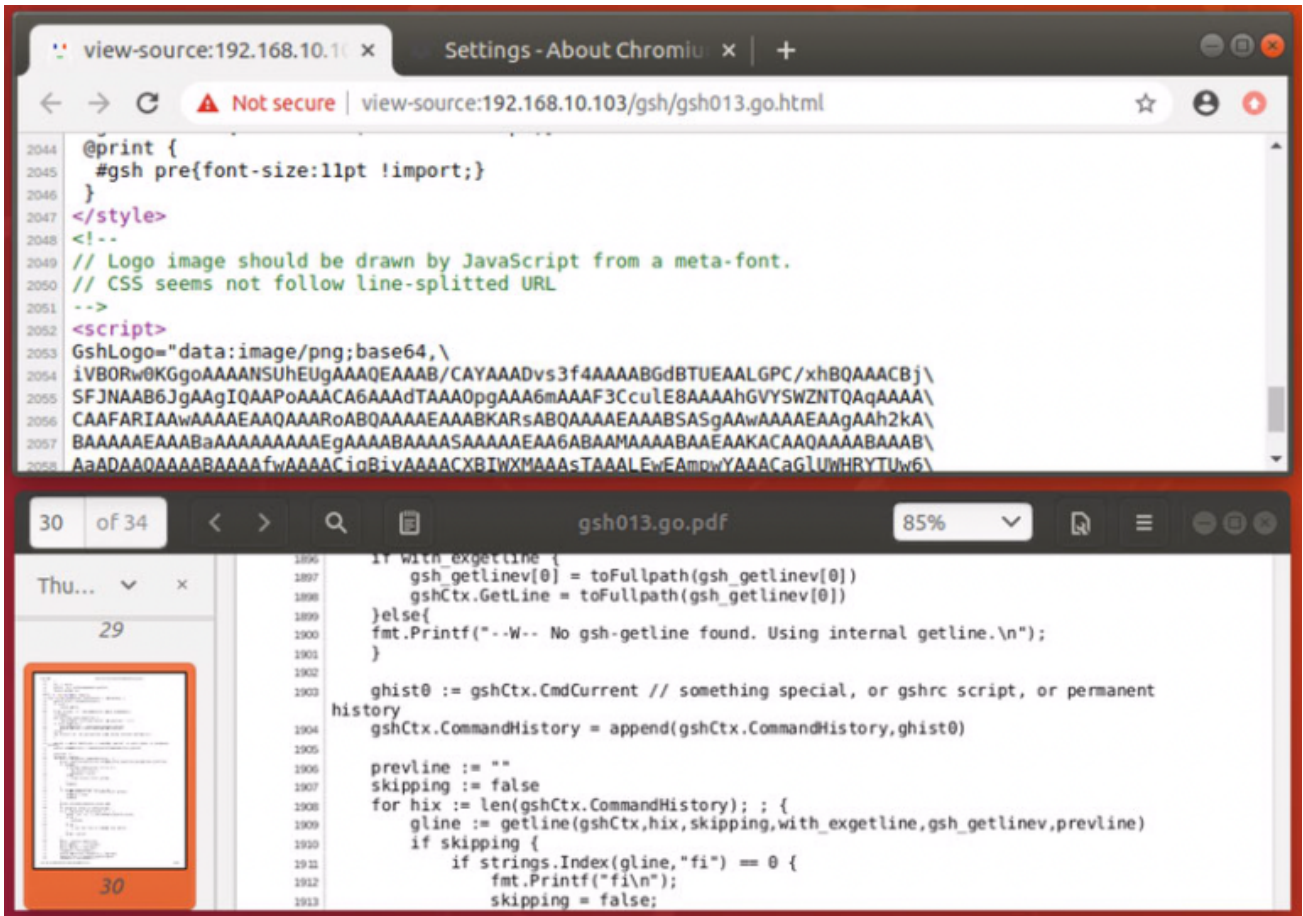
```

1  LXQP"
2  LI EH"
3  PMROPI MGRM JI +7LI PP OSKScMGR-TRK"
4  Q XEWELEWXI 98* c"
5  Q XEWELEWXI 98* c"
6  XMXP " +7LI PP c-88-88" EXSY- 87 _XMXP "
7  LI EH"
8  WERH KVL"
9  LI EH VEH FERRI VEH MKLX! 98cYMRGPMGO VLMJX&+@ @"
10 HMZPMKR! WKLX" RSX! +7LI PP WSR-88-88" EXSY- 87 _RSX! " _HMZ"
11 LI EH V"
12 +7LI PP WSR-88-88" RI VEP YVTSW WSR-88-88" YMPX WSR-88-88" STW WSR-88-88" SPERK _"
13
14
15 H XEMPV H SZI VZM [ " WQCEV" ] 3ZI VZM [ _WQCEV" ] TM "
16 8SRM WMXI R
17 _TM " _H XEMPV"
18
19
20 H XEMPV H MRH ¥" WQCEV" -RH ¥ _WQCEV"
21 TM WSRGPMGO +HGQ RX-KI X)PI Q RX&]-H KSGSH _ +STI RI XVI @"

```

社長：さすが、独創の雄ですね。

開発：ということで、Firefox 一択となりました。と、そこでふと Chromium のことを思い出したので特別参加。でしたが、表示は良いのですが、印刷がペケで脱落です。



社長：印刷もよさそうに見えますが。

開発：確かにそう言われてみれば… そうか、自分でPDFを表示する時にはいいんですね。これを Acrobat で表示するとこうになってしまう。

```

1896     if with_exgetline {
1897         gsh_getlinev[0] = toFullpath(gsh_getlinev[0])
1898         gshCtx.GetLine = toFullpath(gsh_getlinev[0])
1899     }else{
1900     fmt.Printf("--W-- No gsh-getline found. Using internal getline.\n");
1901     }
1902
1903     ghist0 := gshCtx.CmdCurrent // something special, or gshrc script, or permanent
1904     history
1905     gshCtx.CommandHistory = append(gshCtx.CommandHistory,ghist0)
1906
1907     prevline := ""
1908     skipping := false
1909     for hix := len(gshCtx.CommandHistory); ; {
1910         gline := getline(gshCtx,hix,skipping,with_exgetline,gsh_getlinev,prevline)
1911         if skipping {
1912             if strings.Index(gline,"fi") == 0 {
1913                 fmt.Printf("fi\n");
1914                 skipping = false;

```

開発：我社の基準ではソースコードの表示は Courier New でなくてはならないのです。

社長：まあ、固定ピッチならとりあえずいいんじゃないですかね…

開発：Chromium のHTMLソースPDF化では、常識的なフォントサイズで出力するので100%表示で読める。ただ、そのためにGShell コードの横幅が足りないことがある。100

文字くらいになることはあるわけです。そうするとおり返して印刷するけれど、ここは一つの行でしたという事は行番号でわかります。

```

1901     }
1902
1903     ghist0 := gshCtx.CmdCurrent // something special, or gshrc script, or permanent
history
1904     gshCtx.CommandHistory = append(gshCtx.CommandHistory,ghist0)
1905
1906     prevline := ""
1907     skipping := false

```



基盤：これはふつうに Courier になってますね。

開発：あれ… そうか、Acrobat が勝手にフォントを選択してるみたいですね。ぷんぷん。

開発：一方FirefoxでのPDF化では、フォントサイズをすごく小さく印刷するので、まあ折り返される事は無い。あれで折り返されるとしたら、コードの書き方が変。もし折り返されるとしてもChromiumと同じ行番号処理をするでしょう。他のブラウザでもそうなりますし。で、同じ部分を表示するとこうなります。ただし200%表示です。

```

1901     }
1902
1903     ghist0 := gshCtx.CmdCurrent // something special, or gshrc script, or permanent history
1904     gshCtx.CommandHistory = append(gshCtx.CommandHistory,ghist0)
1905
1906     prevline := ""
1907     skipping := false

```



開発：ソースコードを紙に印刷する事はまあ無いと思いますし、やたらPDFの改ページで切られるとコードが読みにくいですから、私はFirefoxの方針が妥当だと思います。

社長：でもそのへんって、印刷時のプロパティで設定するとか、もし出来ないようならソースコードがあるんで自由にいじれますよね。CSSの@printとか… ああ、ソースコードの印刷までは制御できないかな。

開発：デフォルトをどう設定しているかで、そのツールの姿勢というのがわかるわけで、そこは尊重したいと思います。

社長：了解。では、HTMLのソースの印刷はFirefoxということで。

基盤：ちょっとちゃぶ台返しですが、見た目の問題に関しては、そもそもソースとしての印刷用のHTMLを生成してやればよいのではないかと。適切な場所に改ページを入れたり。そうすれば、ブラウザ依存性も回避できると思いますが。

開発：… うーむ、gsh.go.htmlに「ソースを印刷する型式にする」的なボタンを付けると…

基盤：必要なら行番号もつけちゃったりして。ついでに電子署名も付けちゃったりする。

開発：まあ、電子署名はちょっとわかりませんが、印刷用のHTMLは JavaScript で生成できそうですね。電子署名は・・・空白情報の保存が保証されないから、空白を正規化した形にして署名するんですかね・・・公開側の鍵情報というか証明書も内蔵すれば、署名を検証するのも簡単・・・

社長：自分を印刷する型式とか電子署名するアルゴリズムを内蔵するソースコードって、なんだか素敵ですね。いずれそうしましょう。でも当面は Firefox でも良いです。

開発：えーと。第7の選択肢としてXcodeでの印刷もなかなか良いと言う件を書こうと思ったのですが、自分で自分を印刷して署名する方法の話が面白そうなので、どうでも良くなりました（笑）

社長：HTMLに関しては、署名・検証アルゴリズム内蔵方式で、脱PDFできるかもですね。

開発：といたしますか、data URIがかなり使えそうなので、ディレクトリ階層をごっそりHTMLにしてしまう「htar」というアーカイブ型式を考えているんです。これは形式としてはとてもシンプルです。ウェブページで言えば、引用しているCSSとかJavaScriptとか、インライン画像もdata URIで埋め込めば、そのままウェブ魚拓にもなるわけです。CSSにも自己署名させるようにしたら、部品単位でされた署名も自動的に引き継げる。まとめるのも展開するのもGShellのコマンド列でやるとすれば、GShell Archive つまり gshar 型式というようなものになります。

社長：となると何でも署名付きHTMLになるから、バイバイPDF署名って事が可能になるかもですね。

社長：それはそうと、GShell を始めたのが 8/7なので、もう11日目になりました。先週は没頭していて曜日を忘れて月曜のボウリングの試合に出られませんでした。今日は8時に出かけられるように、計画的に続きの仕事を進めたいと思います。

— 2020-0817 SatoxITS