

株式会社 ITS MORE

2020年4月設立

ITS more

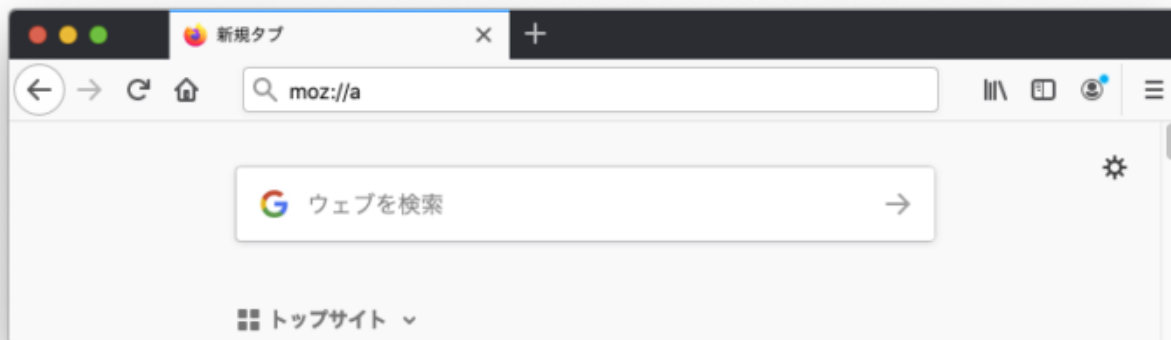
2020年9月17日 投稿者: SATOXITS

GShell 0.4.4 – Chimella

社長：Chimella。なんかいいですね。GShell はやめて Chimella という名前にしましょうか？

開発：そういう表記だと、ペプシコのラみたいですねw

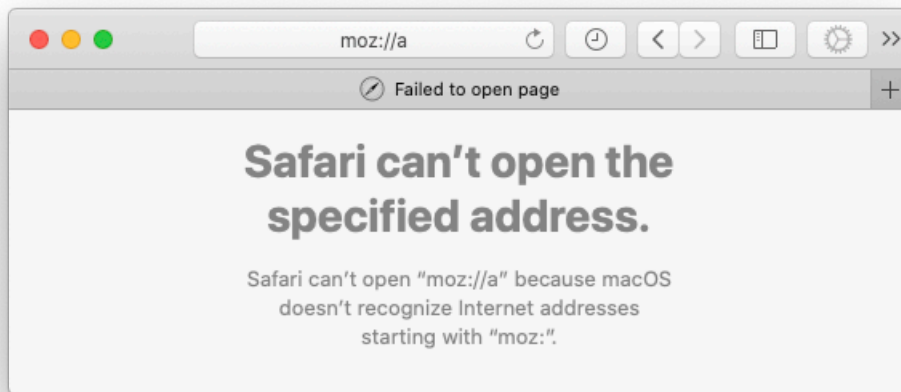
基盤：思ったのですが、moz://a というダジャレがあるなら、moz という URL スキームがあるんじゃないかと。で、やってみました。





社長：しゃれてますねー。

基盤：他のブラウザからはガン無視されてますけどねw





社長：ともかく、gsh.go.html の作り自体がまずキメラです。Chromium の ch、IME、大好きな //、付け足して a。

開発：名前自体がキメラなのも良いですね。いずれ、gsh.go.js.html という風にもでき

ればと。

基盤：やたらミドルネームが長いというw

社長：太古の昔から、人間と他の生物とか機械をかけ合わせて特殊能力を獲得するというテーマは鉄板なわけです。日本神話でもよく出てくる。

基盤：鷹の爪にも合成マシンが出てきましたね。タンス人間とか。

開発：おお、この voiceroid+ 鷹の爪 吉田くん版、めっちゃ面白いですね。

社長：9,800円ですか。ちゃんとしたサービスはソフトは1万円というのがいまどきの社会的なコンセンサスのように思えますね。

基盤：YouTubeに鷹の爪公式チャンネルあります。

社長：登録しましょう。

開発：考えてみるとこのブログ、潜在意識的に鷹の爪の影響をもろに受けてましたね。

社長：鷹の爪をNHKで見てたのは随分昔ですけどね。

基盤：YouTubeで旧作を全部見たのは2、3年前ではないかと。

社長：いっそ、総統とか戦闘主任とかいう役割名にするのも良いですかね…

開発：システムの作りとしてはキメラというのは良くない形容ですよ。

社長：良いキメラと悪いキメラがあるんだと思います。良いキメラを作る、あるいはそのためのインフラを目指せばよいのではないですかね。一つの言語で世界統一というのは幻想だと思います。そもそもウェブからしてHTML、CSS、JavaScript のキメラなんですし。

WordPressカスタムHTMLで&

社長：そう言えば、0.4.3にしたらWordPressの中のJavaScriptが動かなくなってます。

開発：diff … ああ、JavaScriptの中で && を使ってますね。拡張HTMLに貼り付けると & を & にしてしまいきますので、JavaScript としては壊れます。とりあえず、`a && b` を `and(a,b)` とかにしておきましょう。

基盤：!(!a || !b) という手もありますね。

社長：こういうのは学生時代なら即思いついたでしょうね。

開発：見た目にどうですかねえ… ともかく、`and` にしました。

社長：WordPress にペースト。ああ、直りましたね。

基盤：そう言えば以前、CRC32の中で使ってて同じように引っかかりましたが。

開発：あれはMSBの判断だったので、正負の判定結果を `a = 0 < x ? 0:1` のように0, 1にして、`(a && !b) || (!a && b)` を `a^b` にして回避しました。

社長：`bool` と `int` が生まれつき型的に違うのは不便な事がありますね。Cでは横紙破りですが、整数のビット演算と論理ちの論理演算を一緒くたにして処理してました。

開発：型といえば、Golangで整数値の型の自動キャストがされないのはびっくりでした。トンプソンも何か改心したという事ですかね。

社長：まあK&Rの軽便さはANSI化時点で否定されましたからね。

開発：私は長い間、KはKen Thompson のKかと思ってました。1stネームで形容されるって、らしいなあとかw

基盤：ANSIってエスケープシーケンスですか。

イベントのスルー

開発：ところで、単に表示用に上にかぶっているだけでも、下にいる実体に行くべきイベントが自分で止まってしまうのは困りものです。リスンもしてないと思うんですが。

基盤：これって逆に、本来のプレーンの上の一部に透明保護シートをかけるのに使うと良さそうですが。

開発：まあそれはそうですね。

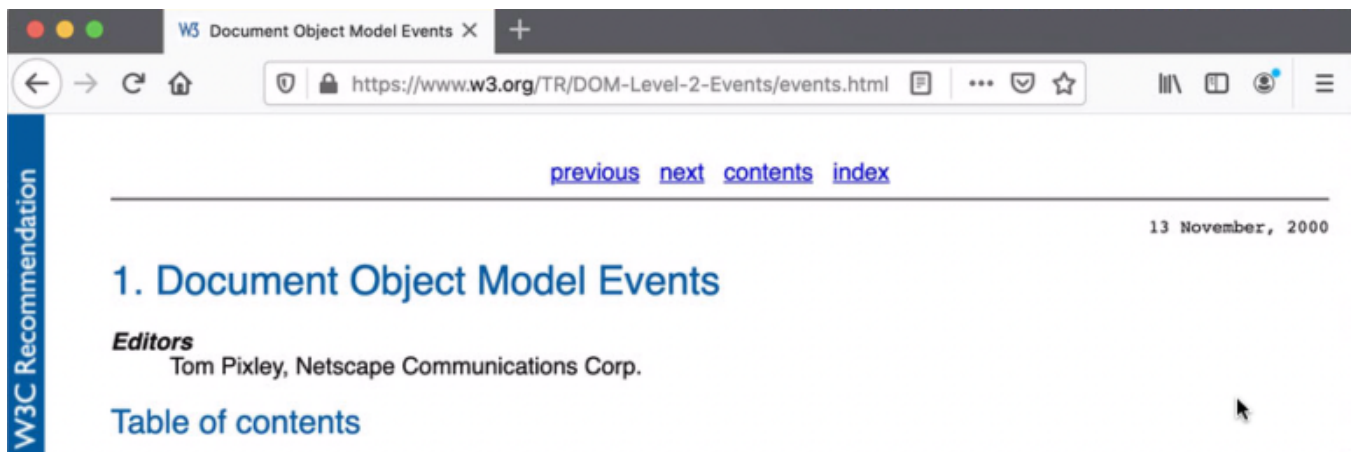
社長：キーイベントとかは逆に下とか上に行かないように明示的にブロックしてますよね。stopPropagation とか preventDefault とか。

開発：eventのcapturingとbubblingのフェーズについて書いてある文書は山ほどあるんですけどね。

社長：やるとすると、自分の下積みになってる要素を検出して、そこに明示的に投げられて感じですかね。

基盤：そもそも、自前でイベントを作って指定したターゲットに投げたいですよね。GShell のキーボード入力のリプレイコマンドとかやるのにも。

開発：なにやら検索してもわからないので、W3Cにある原典を見ましょう… あれ？単に Event を create して dispatch すれば良いみたいです。



基盤：2000年の11月にはもう仕様は完成してたってことですね。

開発：一方そのころわたくしたちは・・・

社長：実装が進んでなくて混乱してる時代にはじめてたら、それはそれで辛い事が多かったんじゃないですかね。2020年にはじめてのは少し遅すぎたかもしれないですが。

社長：それはそうと、ことWebの仕様とGolangの仕様については、ぐぐると出てくる膨大なノイズを除去する時間が無駄だと強く感じます。ただ、W3Cのは記述の単位が大きくて、その中で探すのがブラウザの文字列検索でしかできないのが痛いところ。

開発：今のところ良いと思われるのは、検索サイトでMDNが優先されるように検索する、MDNでざっとみて、全体像を知りたい時には仕様のリンクからW3Cに行く。これがベストの探索コースかなと思います。MDNは各ブラウザの実装状況の情報が充実している点も便利です。

社長：Chimellaにgrep機能をつけて、長いページの中を一覧検索でると良いと思うのですが。ひとつのドキュメントに適用できるかですが。

基盤：ブラウザ内蔵のインスペクタなら何でもできると思いますけどね。

開発：開発者向けじゃなくて、一般ユーザ向けの全文検索インスペクタとか、概要構造表示とかできると良いのに。

社長：まあそれこそChimellaが目指すところの一つなんだと思います。理想的にはiframeの上に透明シートを被せて検索機能を追加できると良いと思うのですが。iframeを作って表示させるのが自分なら、作ったウィンドウのコンテンツをその後覗けるのか、という所が焦点かと思います。以前やりかけたwindow間通信でもこれはキモです。

基盤：別のウィンドウにイベントを送れる必要もありますね。

社長：それはそうとお腹がすいたので食事して来ます。

* * *

社長：あー、のんだかった… (^-^)//

基盤：このとうもろこしは？

社長：郵便受けの上にあった大家さんの差し入れです。

開発：これ、4本なので時価760円相当ですね。

社長：いずれあのおっきな栗の差し入れもあるものと期待しております。

基盤：スーパーで売ってるのに比べると色的に皮の緑が濃くてワイルドですね。

社長：おなかいっぱいだけど味見したいです。

開発：チンしましょう… 90度で5分30秒。

社長：そう言えば昨日meijiの販促のお兄さんが置いてったセット、かなり良いですね。

開発：そもそもびん入りというのがすごく潤いを感じます。

基盤：美味しい牛乳パックのキャップが社内で大評判でしたが、やはり紙パックは味気ないです。ゴミにもなるし。

開発：びんは回収前提なんですかね。となると、毎日お届け形式？

社長：ラベルに「宅配専用」ってありますね。

基盤：明治おばさんとかレディーとか聞いた事ないですけどね。

社長：さすがに明治生まれのおばさんで配達してる方はもういないのでは。1868-1912ですから、末年生まれで存命されてても108歳です。

開発：大正時代って15年しか無いんですね。かたや平成ですら30年。

社長：インターネットの時代でしたね。そして私の勤め人時代。

基盤：とうもろこし美味しいですねはぐはぐ。

開発：この感じだと5分ジャストでよかったのかも。もぐもぐ。

社長：もぐもぐ。焼きもろこしにしたら最適みたいな品種ですね。ご馳走様。

スクリプトの見える化

開発：ふあああ…

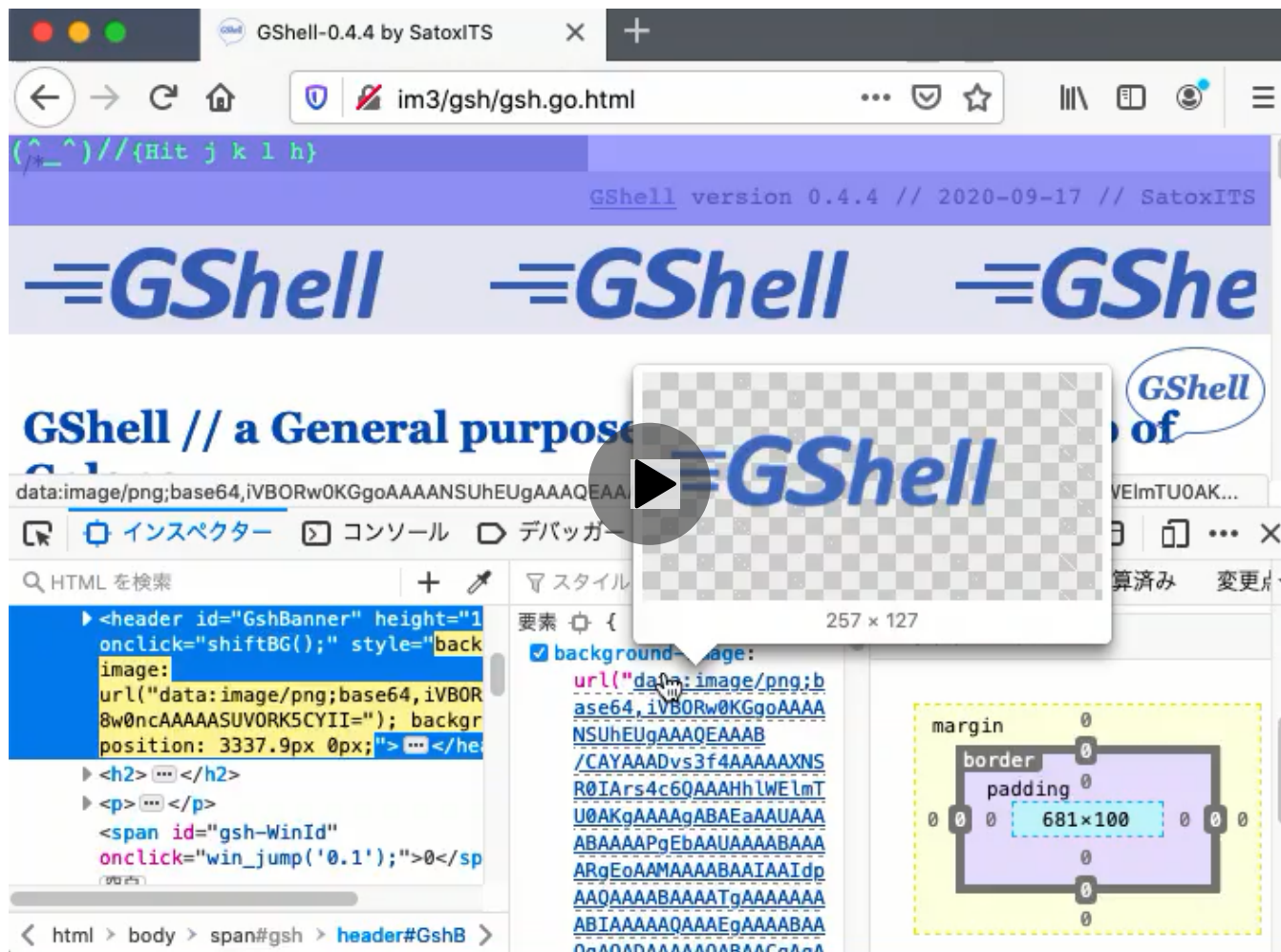
社長：午睡中に考えたのですが、インラインスクリプトを見える化してはどうだろうと。

開発：ブラウザで見えるという意味ですね。

社長：そう。普通にspanにしておいて、ブラウザで表示する時に内容を取り出して、表示をhiddenにして、evalすると。

開発：やってみましょう。

開発：ちょっとその前に。開発者ツールを開いたら面白いものが見えました。



開発：バナーイメージをスライドするために background-position を書き換えている様子が見えます。

基盤：インスペクター、最高ですね！

社長：ディープ・インスペクターですね。

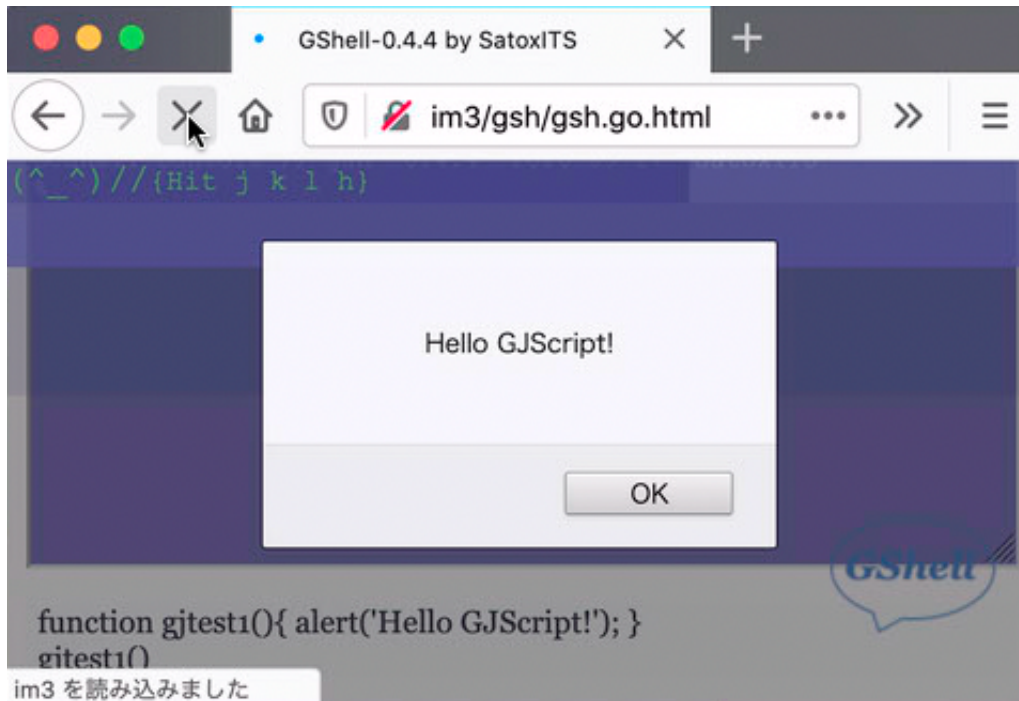
開発：で、まず仕込みはこう。

```

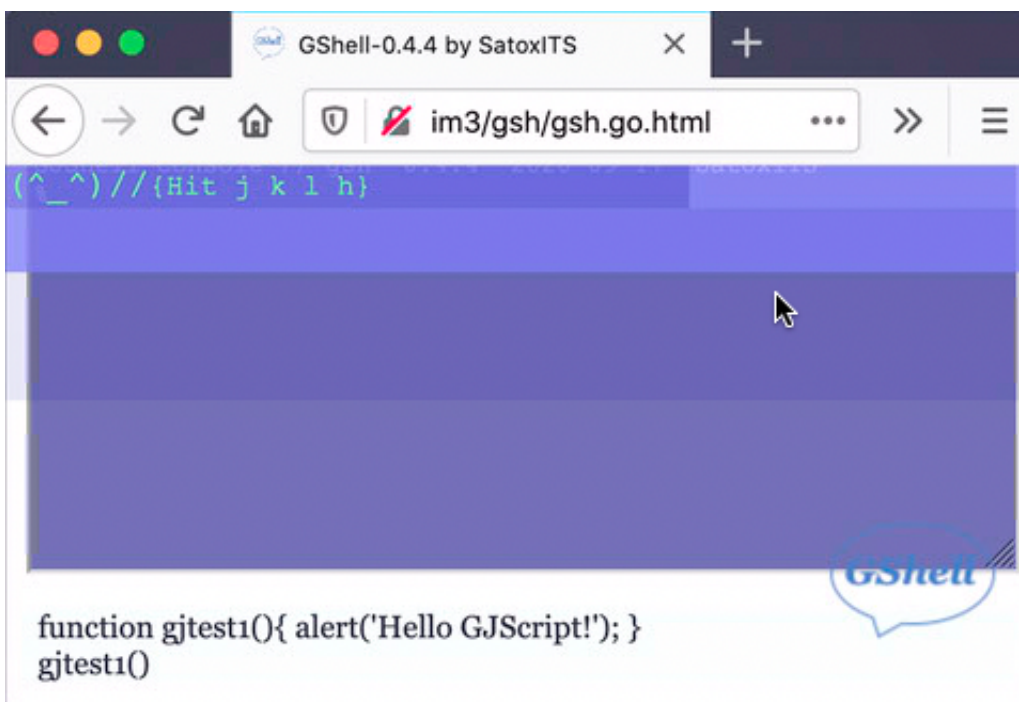
<!-- 2020-09-17 SatoxITS, visible script -->
<style>.gjscript { font-family:Georgia; }</style>
<pre id="gjscript_1" class="gjscript">
  function gjtest1(){ alert('Hello GJScript!'); }
  gjtest1()
</pre>
<script>
  gjs = document.getElementById('gjscript_1');
  eval(gjs.innerHTML);
</script>

```

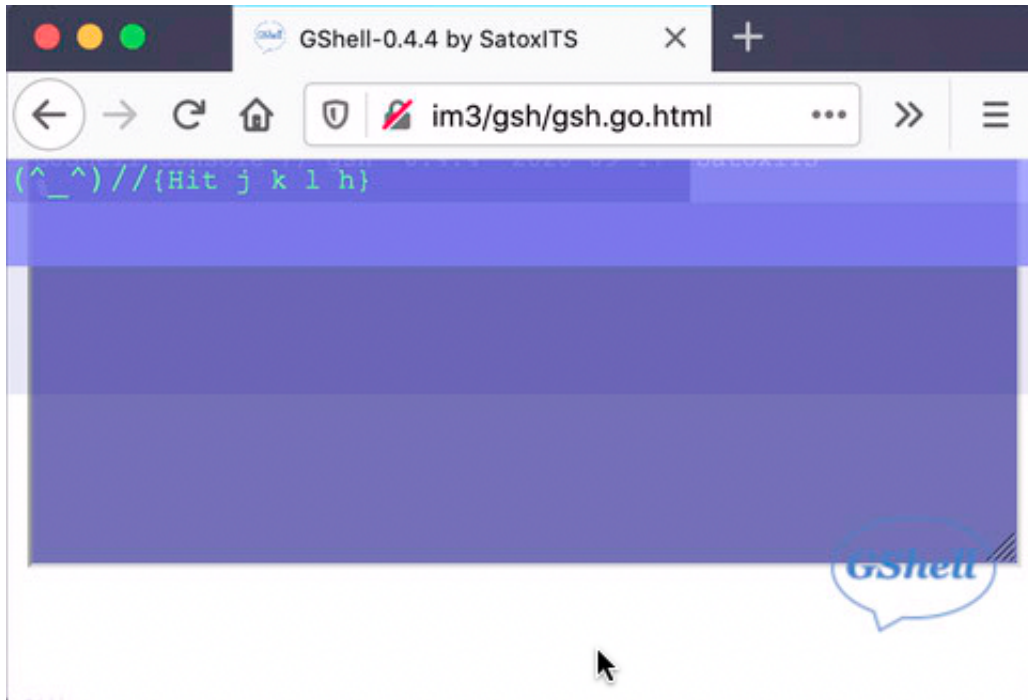
開発：これを開くとこうなります。



開発：もちろんこのままだと表示が残ります。



開発：消したかったら outerHTMLごと消してしまえば良い。



社長：ブラボー！期待通りの動作ですね。

開発：うーん。ソースコードのインタプリタって、どうやれば遊びきれのかわからないほど面白いですね。

社長：きっと Lisp の人たちはそうやって遊んでたんでしょう。

基盤：消さなくても、details で畳んでおくとか、hiddenをオフにするスイッチをつけるでも良いわけですね。

開発：そうですね。やはり details にしておきましょう。

社長：この方法でインラインの style も見える化したいですね。

クラス単位でのスタイルの動的書き換え

開発：ただ、実行時にインライン style エレメントを生成して有効化すると、セレクターが効かないようにも思えます。一昨日、table を生成して試したところでは。あれは大変困ります。

社長：限定したかったら、対象の style 属性に書き込むしかないんですかね？

開発：適用したい対象のクラスのインスタンスが複数ある場合には嫌ですね。クラスが
ムルとしたら、せめて自分にしか適用されない style というのができれば良いのです
が。

社長：そういえば抹殺されてしまった scoped というのがありました。

開発：MDNで検索。

開発：ポリフィルというのはクラスもサポートしてるようには見えるんですが。どのみ
ち JavaScript でやるなら、自前でやったほうが面白いですね。

社長：ひょっとしてと思ったのですが、ひょっとしてタグの style 属性の string って
複製してやらないと、ポインターとして共有されるとかは無いんですかね？

開発：！… 可能性はありますね。あるいは、style = “StyleFunction(className)”
みたいに与えてやって、動的に評価して反映してくれるとか。

社長：W3Cでのstyleの定義によると、media の変化には追従してくれるみたいで
すね。まあ、縦横が変わったりしますし。

開発：<style>@media screen { }</style> とかで括ってやったらいいんですかね？

社長：それはできないって、StackOverflowにありますね。… というか、そもそも、
styleSeets.insertRule で、セレクタ付きのCSSを挿入する例が載っていますが… こ

れって、インラインじゃ無いスタイルシートを変えられるみたいな。

開発：MDNで調べる… [CSSStyleSheet.insertRule\(\)](#)。ルールをひとつずつしか加えられないようではありますが、まあOKですかね。変更する時には、`deleteRule` して追加し直せば良いと。

社長：再びW3Cに戻る… 要するにこれはDOMではなくて、CSSOMをいじるということですね。

開発：GJShellにコマンドを加えてやってみます。

* * *

開発：うーん。どうもわかりませんね。結局、`insertRule` でもセレクトはタグ名しか使えないように見えます。idも class名も効かない。style要素のinnerHTMLに書き込んだのと同じ効果のようです。多少ありがたいのは、DOMをいじらずにスタイルを変更できるって事くらいでしょうかね。ですが、DOMの中でのインラインスタイルとバッティングしてるのか、関係無いところが変わったりする始末。

社長：結局、DOMにある個別の要素のstyle属性に直接書いて回るのが、現状では正解なのかもしれませんね。

開発：クラスでのデフォルト値を変えたら、そのインスタンスの値も一斉に変わるってのは、30年前の idraw でも出来ていた事だと思うんで… できないはずないと思うんですが…

* * *

開発：ヒィヒィ… なんとかできました。CSSOMのいじり方がなんとかわかりました。まずデフォルトのルールで生成するところになります。

```
GJShell Console // gsh--0.4.4--2020-09-17--SatoxITS
% new table
created TABLE id="table_1"
% new table
created TABLE id="table_2"
% print
<span>
<span id="table_1-span" contenteditable="true" draggable="true">
<span id="table_1" class="table_1">
<table onkeydown="GJE_TableKeyCommand(event,this)">
<tbody>
<tr>
<td>A0</td>
<td>B0</td>
<td>C0</td>
</tr>
<tr>
<td>A1</td>
<td>B1</td>
<td>C1</td>
</tr>
</tbody>
</table>
</span>
<textarea id="table_1-style_def" class="_style_editor" spellcheck="false" cols="30" rows="7">
#table_1 *{
border:1px solid #aaa;
background-color:#efe;
color:#222;
} /* hit ESC to apply */
</textarea></span>

<span id="table_2-span" contenteditable="true" draggable="true">
<span id="table_2" class="table_2">
<table onkeydown="GJE_TableKeyCommand(event,this)">
<tbody>
<tr>
<td>A0</td>
<td>B0</td>
<td>C0</td>
</tr>
<tr>
<td>A1</td>
<td>B1</td>
<td>C1</td>
</tr>
</tbody>
</table>
</span>
<textarea id="table_2-style_def" class="_style_editor" spellcheck="false" cols="30" rows="7">
#table_2 *{
border:1px solid #aaa;
background-color:#efe;
color:#222;
} /* hit ESC to apply */
</textarea></span>
</span>
%
```

A0	B0	C0
A1	B1	C1

```
#table_1 *{
border:1px solid #aaa;
background-color:#efe;
color:#222;
} /* hit ESC to apply */
```

A0	B0	C0
A1	B1	C1

```
#table_2 *{
border:1px solid #aaa;
background-color:#efe;
color:#222;
} /* hit ESC to apply */
```

開発：次にスタイルシートを編集して適用するようになります。

```

#table_2 *{
  border:1px solid #aaa;
  background-color:#efe;
  color:#222;
} /* hit ESC to apply */
</textarea></span>
</span>
% print
<span>
<span id="table_1-span" contenteditable="true" draggable="true">
<span id="table_1" class="table_1">
<table onkeydown="GJE_TableKeyCommand(event,this)">
<tbody>
<tr>
<td>A0</td>
<td>B0</td>
<td>C0</td>
</tr>
<tr>
<td>A1</td>
<td>B1</td>
<td>C1</td>
</tr>
</tbody>
</table>
</span>
<textarea id="table_1-style_def" class="_style_editor" spellcheck="false" cols="30" rows="7">
#table_1 *{
  border:1px solid #f00;
  background-color:#efe;
  color:#222;
} /* hit ESC to apply */
</textarea></span>
<span id="table_2-span" contenteditable="true" draggable="true">
<span id="table_2" class="table_2">
<table onkeydown="GJE_TableKeyCommand(event,this)">
<tbody>
<tr>
<td>A0</td>
<td>B0</td>
<td>C0</td>
</tr>
<tr>
<td>A1</td>
<td>B1</td>
<td>C1</td>
</tr>
</tbody>
</table>
</span>
<textarea id="table_2-style_def" class="_style_editor" spellcheck="false" cols="30" rows="7">
#table_2 *{
  border:1px dashed #00f;
  background-color:#aaa;
  color:#fff;
} /* hit ESC to apply */
</textarea></span>
</span>

```

(^_^)//{Hit j k l h}

A0	B0	C0
A1	B1	C1

```

#table_1 *{
  border:1px solid #f00;
  background-color:#efe;
  color:#222;
} /* hit ESC to apply */

```

A0	B0	C0
A1	B1	C1

```

#table_2 *{
  border:1px dashed #00f;
  background-color:#aaa;
  color:#fff;
} /* hit ESC to apply */

```

開発：セクタは自由に指定できるので、大変危険です (^-^);

社長：オリジナルを全部飛ばして、だましの背景画像ですり替えたりできるわけですね。

開発：CSSOMに書き込む前にチェックが必要と思います。もともと保護の機構もあるのでそれに任せるだけで良いのかもしれませんが。でも、もう疲れました。

社長：今日はここで終了にしましょう。

— 2020-0917 SatoxITS

gsh-0.4.4.go

ダウンロード

/* */ /*

GShell version 0.4.4 // 2020-09-17 // SatoxITS

≡GShell

≡GShell

≡GS

GShell // a General purpose Shell built on the top of Golang

It is a shell for myself, by myself, of myself. -SatoxITS(^-^)

0 Fork Stop Unfold Digest Source */ /*

▶ Statement

/ /

▶ Features

/ /

▶ Index

*/ //

▶ Go Source

//

▶ Considerations

// /*

▶ References

/ /

▶ Raw Source

/ /

▶ GJScript

/ /

GJ Factory_1

GJ

```
GJShell Console // gsh-0.4.4-2020-09-17-SatoxITS
%
```

/ /