株式会社 ITS MORE
2020年4月設立

Digitally signed by 佐藤 豊
Date: 2020.06.14 11:28:11 +09'00'

2020年6月13日 投稿者: SATOXITS

UEE – URL Elevator Express

URL Elevator *express*

https://its-more.jp/ja_jp/
[上のURLの一部をパッと取り出せます]

開発：例の file URL の中の上位のパスにジャンプする機能ですが。

社長：ブラウザの開発環境もできたし、そろそろまじめに作りますか。

開発：いえ、やめようと思います。

社長：あ、そうなの。なぜ？

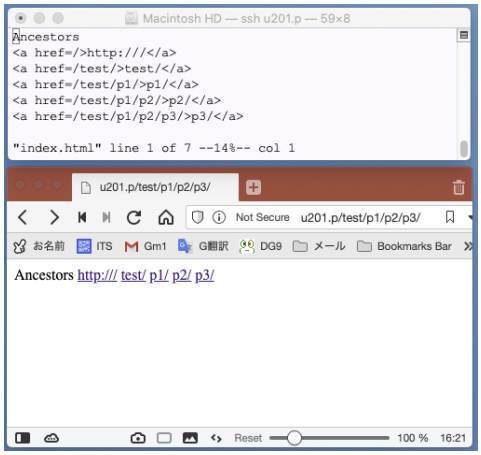
開発：これは基本、file や ftp に限らず、URL全般に適用したほうが良いと思い直したんです。やることは同じなんですが、やる場所が変わります。といいますか、やる場所の選択肢が増えます。

基盤：でも、やる場所によって使える言語が変わりますよね。

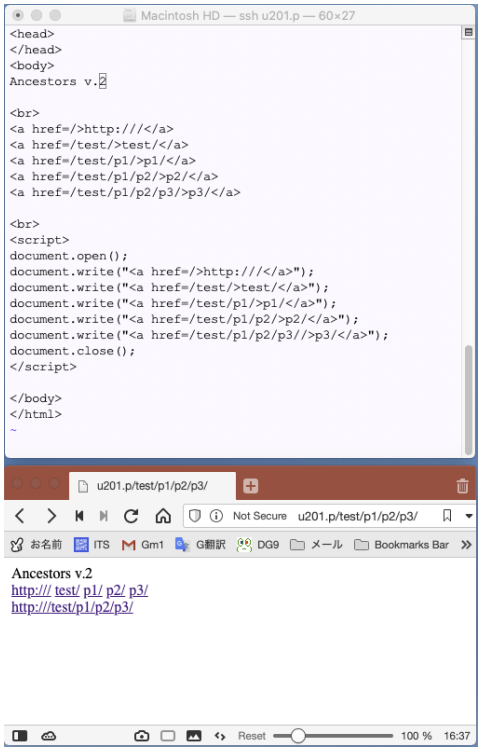
開発：文字列の変換は同じ、表示の仕方だけの違いですから、本質的には単純ですよね。

基盤：多分本命は苦手の JavaScript ということになりますが。

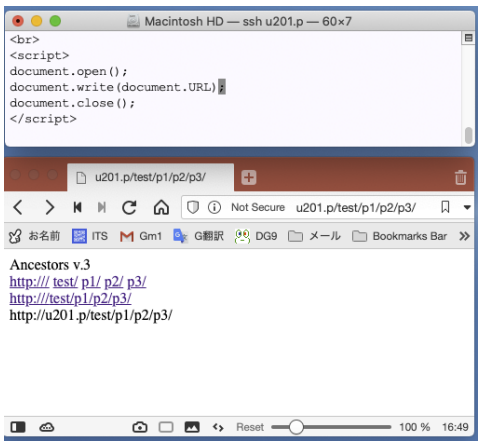
開発：ゼロからのスタートです。で、まずモックアップから。やりたいことは、こういうのを自動生成することです。



開発：で、これを JavaScriptで書くとこうなる。これ、私が人生で初めて書いたJavaScript です（笑）



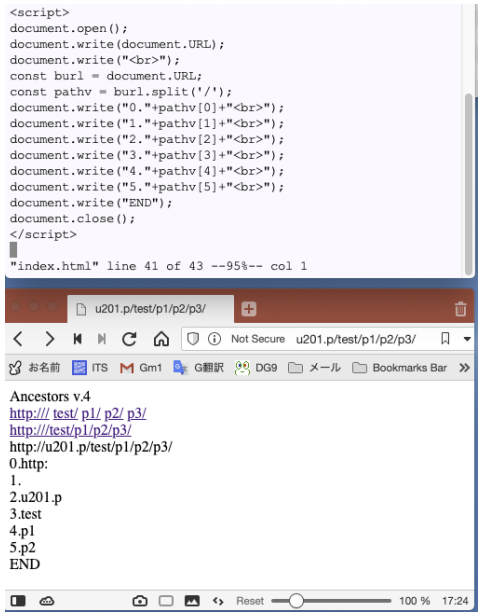
開発：で、一般化するために現在のページのURLを取ってくるわけですが、たぶん document.url とかじゃないかと思ったんですが、undefined になる。で、仕様を調べると document.URL なんですね。惜しかった。JavaScriptが case sensitive なんだということを初めて明確に理解しました。



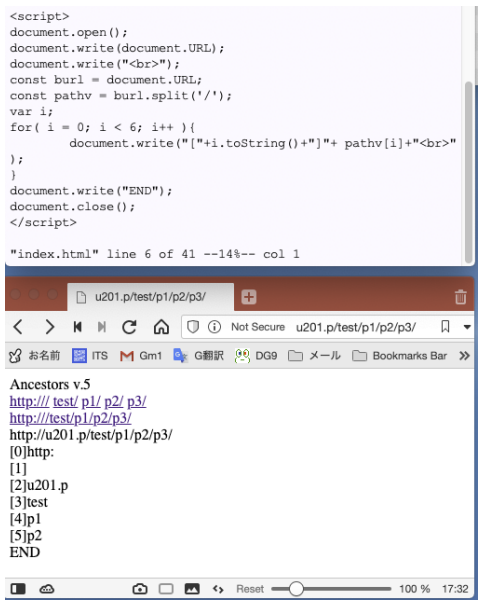
¥

開発：で、これを / を区切りに分解して、アンカーを挟み込みながら結合するわけです。そういうのは一般的に split / join というような名前の関数に違いありません。で、javascript split join でググるとたくさん例が出てきます。由緒もあるのでMDNにあるsplitのガイドから。

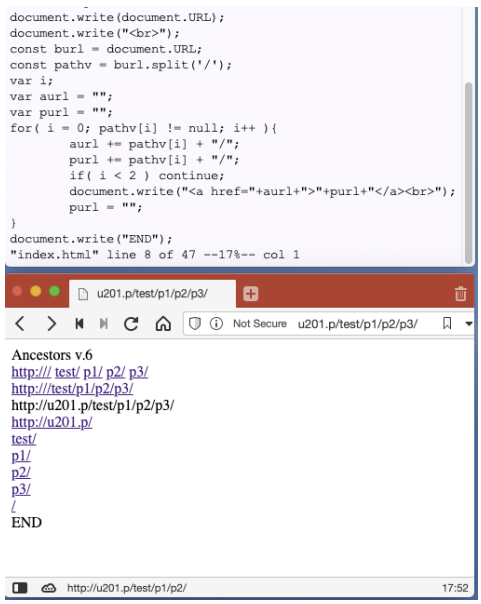
開発：余計なことが書いてないのでわかりやすいですね。それでちょっと変えてリロードしたら、何も出なくなりました。追加した部分で write() を wirte() と書き間違ってたんたんです。ステートメントの逐次実行即出力じゃないようです。また文字列の連結をとりあえずCと同様に並べて書いてしまったのですが、+ で連結することがわかりました。あと、JavaScriptのコメントがCと同じで /* */ だということがわかりました。



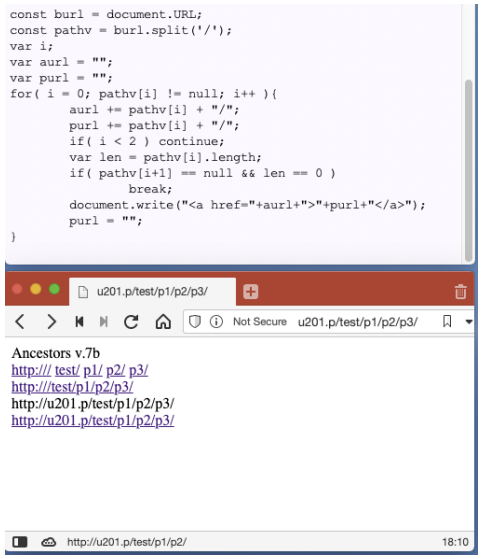
開発：for文のシンタックスがCと同じだということがわかり、少し驚きました。C風の言語は多いですが、for の構文は崩されてしまています。また、値を文字列にするのは toString() じゃないかと思ったら当たりました。



開発：ぬるぽは nil かなとおもったら null でした。



開発：ということで、文字列処理については雰囲気はつかめました。表示部分をちょっと整えて一旦休憩します。



基盤：前回、C＋＋で同じ処理を実装する時には、もっと時間がかかってましたね。

開発：あれは、C++にsplit / join がなかったことと、自作の生文字列 split / join と Mozilla のストリングクラスとのやりとりがよくわからなかったからです。

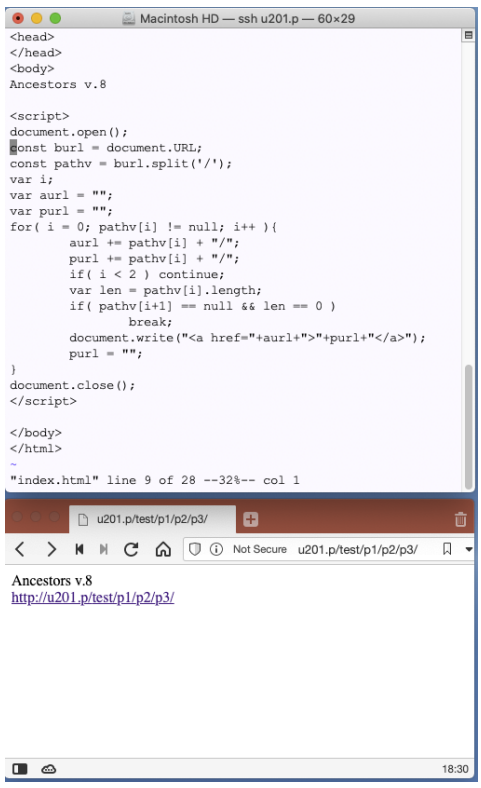
開発：ここまでのところ、C言語に慣れている人間には、ごく普通の言語だなという印象です。ただ、タイポが多い自分としては、未定義名とか実行するまでわからないのは、ちょっと厳しい感じはしました。

社長：変更するたびに100%のカバレッジでのテストが必須なんでしょうかね。いや、昔少しだけ Google Apps Script （GAS）で遊んだことがあって、あれはその点が結構問題でした。GASとか、何時間も走らせるわけですが、えらく時間がたったあとに初めて実行するステートメントで、構文エラーですって。それまでの処理がパーになるわけです。実行前にそういう静的な検証をしてくれない処理系って、私には無理ですね。

開発：そういう意味では、ちゃんとユニットに分解してユニットテストしましょうね、っていう事かもしれませんけどね。

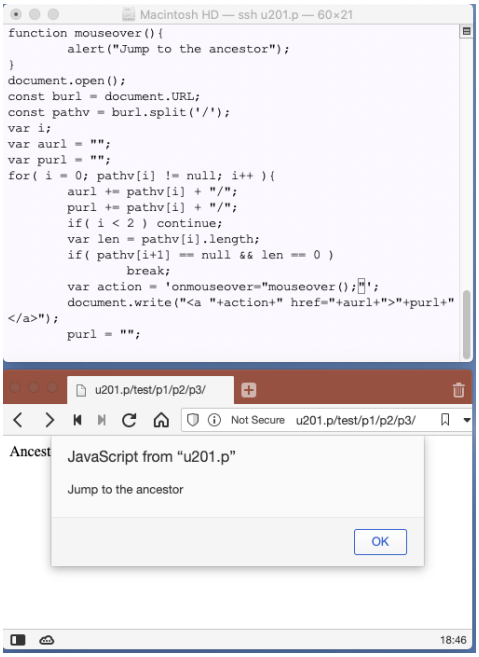
＊　＊　＊

開発：それではなんかアクションを入れてみたいですね。その前にちょっと整理。



開発：マウスカーソルをそのパスに置いたら、その部分のURLがポップするっているのをやりたいですね。まあ、ステータスバーに出てるからいいじゃんていう話ではありますけど。マウスポインターに反応して応答するというのをやってみようかと。私が見覚えがあるのは、

onMouseOver() というやつです。こんな感じ。



開発：反応しましたが、これじゃ使い物にならないですね。ただこれでわかったのは、JavaScriptはcase sensitive だけど、HTMLはinsensitiveだということです。そりゃそうですね。大文字小文字を判別しない言語とか**OSとかファイルシステム**とか、ついでに検索エンジンも、アンビリーバブルです。

開発：じゃあ alert じゃなくて何が適切だろうかというと、ポップアップ系はユーザがなにかポチらないと消えないのでダメですね。

＊　＊　＊

開発：いやはや、しばらく検索してましたが、ポップアップの仕方はすごい色々あるみたいですが、それはまあプログラム書けば何でもできますよね的な感じはします。納得感のあった一例は主にHTMLとCSSで見せ方を記述して、JavaScriptではその可視性だけを制御するという、w3scoolsの例です。

開発：それにしても感じるのは、JavaScriptで書いたページが環境依存だということです。少し昔の解説ページの例題が現在のブラウザで動かないとか。あるいは、以下には今作成中のスクリプトを貼り付けてありますが、WordPressでは動かない。編集モードやブロックのプレビューでは動くのですが・・・

Ancestors v.10

https://its-more.jp/ja_jp/?p=6255/

開発：と思ったもの、/* */でコメントアウトしつつ匍匐前進したら、JavaScriptの配列の使い方が間違っていたことがわかりました（笑）。splitした時に配列がnullで終端してるものと想定してしまったC言語育ちの悲しい佐賀でした。（実際、ゼロになっている場合が多いからタチが悪いわけですが）。

開発：ということで、どうやら**WordPressのページには特に問題なくJavaScriptを貼り付けられる**模様です。となると、何もプラグインとか使わなくてもかなりのことが出来そうな気がします。

開発：ただ、WordPressのHTML編集モードはきわめてしょぼい。まあ、これもHTML編集プラグインとかあるのかもしれませんが。デフォで提供されている「HTMLとして編集する」が寒すぎるんですよね。世の中には結構出来の良いHTMLエディタがあるのに、なんでWordPressのようなメジャーなフレームワークにそれがデフォルトで入ってないのか、理解に苦しみます。

開発：それはそうと、次に知りたいのは、JavaScriptで「現在ポイントされている部分が何であるか」をどう読み出すのか？です。ポイントされている部品は決まっているので、それぞれが「自分がポイントされている時のアクション」を記述することは簡単ですが、それはダサい。でも調べるのが面倒になってきました。指されるほうは onMouseOver とかで自分の上にマウスが出入りした事を知っていますから、onmouoverのイベントに引数を渡せれば、自分だよ、ということを知らせることができるわけです。ですが検索すると、ずばりそういうQ&Aがありました。

Determine which element the mouse pointer is on top of in Javascript

開発：この中で、後続のコメントでobsolete扱いされてますが、XY座標から何がさされているか知る方法として、document.elementFromPoint(x,y) というのが紹介されています。で私はこのQ&A本体よりもその例の中で使われている event.clientX,

event.clientY というものにおーっと思ったわけです。ああ、Google の reCAPTCHA もこれ使ってるのかなって。

＊ ＊ ＊

基盤：それはそうと、私はこの、今年に入ってからだと思いますがウエルシアで山積みイチオシの「辛辛魚」に非常み興味を持ってまして、ゲホッ、大変面白いコンセプトのカップ麺だなと。ズズッ。

社長：これは、スープを最後まで飲もうかどうか検討する価値のある品ですね。ただ残念なことに、この魚介粉がいかにも安物。もしちゃんとした花かつおでも使ったら、その味はいかばかりかと期待してしまいます。

開発：辛さと旨味が不可分に存在しているという食材の例は多いですね。食われるほうは旨くありたいなんて意図は無いんでしょうけど。

＊　＊　＊

開発：それでは続きを進めたいと思います。で、上のv.10からコピーして来たわけですが、そのままでは動かなかったわけですよ。なぜかと言うと、CSSにもJavaScriptにも名前のスコープが無いからです。名前がぶつかったら最初の人に飛んじゃうんですね。なのでしょうがないから名前を全部11に書き変えました。

Ancestors v.11

https://its-more.jp/ja_jp/?p=6255/

基盤：プログラミングの分野で育った人間にはちょっと想像がつかないですね。マクロですか？BASICですか？って。

社長：それ、ずっと不思議だったんですけど、HTML5でも同じなんですかね？

開発：さあ。そもそもの出発点からそうだったっていう時点で、なんかもう価値観が違う人たちなのかなって思うじゃ無いですか。プログラム言語の世界では太古に完成してたやり方を無視して劣化版の再発明ですか。いや、スコープとかって確かに処理には邪魔臭いかもしれませんけど、それをいったらCSSの上書き上書きで一体何が最終的に採用されてんだかワカメっていうあれは何なんだと思いますね。だもんで、WordPressで指定したクラスのフォントサイズを変更するのって、今の私には無理ゲーですよ。

基盤：何かの雇用対策かもしれないですね。コロナだし。

開発：それはそうと、私のJavaScriptに関する疑問は、このElement.**innterHTML**の例を見て氷解しました。シンプルで、全てを物語っているような例題です。それで、今日のお題はこのような解答になりました。

Ancestors v.12 — *Exp. Elevator*

https://its-more.jp/ja_jp/?p=6255/
[↑ お望みの階層へ直行します]

社長：なんだ、そういう事だったのですか。長年の疑問が解けました。とてもわかりやすいのですが、JavaScript って昔からこう言うものだったのでしょうかね？

社長：ところで、HTMLを書き換える方法はわかったのですが、CSSを動的に生成するってできるんでしょうか？

社長：あと、JavaScriptでマンデルブロー書いたら面白そうですね。ベンチマークにもなりそう。

社長：あと、なんちゃってreCAPTCHAも作ってみたいですね。あれ、特許に触れるのかな？

開発：今日できたものを少し整理しまました。はじめての JavaScript、記念写真をパシャ。

urlexp 2020-0613 コード

```
<!-- ---------- --
    UEE - URL Elevator Express (2020-0613)
        for extracting partial URL
    (c) 2020 ITS more, Licensed-Under(CC BY)
---- ---------- -->
<style>
/*.urlexp13 { position:relative; }*/
</style>
```

基盤：Creative Commons てプログラムコードとかにも使うもんですかね。

開発：さあ。単なるノリです。

社長：え？CCって当社クリエィティブコモデティの意味じゃないの？

営業：それなら CC BUY ですよね。

—
2020-0613 SatoxITS

UEE---URL-Elevator-Express---株式会社-ITS-more [redacted]