Less is Mo(o)re: Beyond FORTH
Liang Ng and Demitri Peynado
November 2022

Since its inception in 1968, FORTH has been taken up by a great variety of programmers and engineers, applied to projects from microprocessor to mobile device applications. We believe FORTH, ***if simplified***, has the potential to be as popular as Python, Javascript or similar programming languages, so we implemented a “light” variant of FORTH, retaining its reverse polish notation syntax, stack and colon definition, while leaving out the bulk of standard FORTH vocabulary as optional extensions. We call our implementation Phoscript, where “phos” is “light” in Greek.

Minimal implementation of Phoscript, typically as a shell function in a host (third generation) programming language, starts with around 20 lines of JavaScript or equivalent (including PHP, Python, Java, C, C++, etc.) It consists of a loop, splitting a space delimited string into tokens, pushing data tokens onto the stack, and calling eval() to execute function tokens, which in turn executes a function in the host programming language. We believe such minimalist stack machine can be implemented in ANY known programming language.

Phoscript has been used to create DMeta (Decentralised Metaverse) server, a complete full stack web framework in less than 1MB worth of JavaScript and PHP code, running on web servers hosted on mobile devices running Android Userland Ubuntu Apache server.

Preliminary attempts have been made by Peynado to port Phoscript to Verilog (Project CORE-I by Golding & Peynado), breaking ground for "homotronic" computing, an extension of homoiconic programming, where a programming language (FORTH or Phoscript) is used to design a microprocessor, that is capable of executing said programming language. ("Homo" means "same" in Greek, "tron" is "thing", implying both software and hardware are "unified" with one programming language.)

Phoscript has been ported to React, presently one of the most popular frameworks for web and mobile programming, resulting in a variant called Reverse React Notation, where HTML and JavaScript are unified as reverse polish notation. Work in progress includes Freddiscript, an extension of Reverse React Notation, based on programmer's "find and edit" action on source code (hence FREDDI, for "find regular expression and edit"), that can theoretically be applied to any known programming language.

Readers may recognise that the title is a tribute to the inventor of FORTH. Beyond technical innovations, we wish to conclude this presentation by highlighting one social and financial innovation that FORTH may bring about, via DMeta Decentralised Metaverse:-

- DMeta began with DMeta Contract, which is perhaps the simplest smart contract implementation ever conceived, where hashes of public keys of signing parties and the hash of contract text are all hashed together to create the root hash. FORTH and Phoscript provide an innovative facility in the form of the colon definition, which is essentially a space delimited string, which of course can be hashed, hence called "code hash", and thus allowing programmers to claim ownership of source code with code hash. This essentially solves one of the biggest problems facing free software or open source programmers – allowing the code to be peer reviewed, but enabling programmers to charge prospective users, especially huge corporations, for their use of the code written by various programmers, traceable using code hash.

(Credit to Demitri Peynado for suggesting the title.)

References:

1. **MI4 Metaprogramming in Forth (Part I and II)**
https://www.youtube.com/watch?v=5Fm8e4LC5vo&t=252s

2. Bidirectional Shunting Yard Algorithm (BISYA) and Sandwich API Model: Unifying Programming Languages

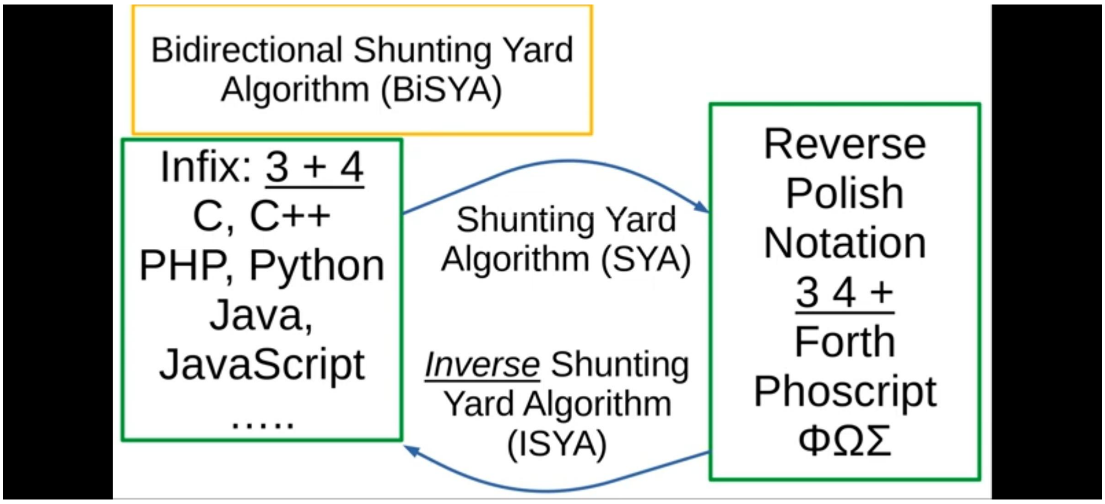
https://www.youtube.com/watch?v=mYjKS0KiJVg

3. DMeta (Decentralised Metaverse)
https://godmeta.github.io/dmeta/

4. FORTH and Homotronic Computing
https://docs.google.com/document/d/1xUGIWrviJPVTiGyCmipT_8-31mVqVz6HVEU9tY5X-4o/edit?usp=sharing

5. The aim of CORE I Project is to produce a human readable, synthesizable, FORTH FPGA computer (not processor) using SystemVerilog behavioural style code. The main applications are AI and robotics, facilitated by the ability to add new instructions and hardware blocks as required. Unlike other FORTH FPGA projects, the inner and outer are implemented in hardware, allowing for rapid compilation and interactivity out of the box.

I. Introduction



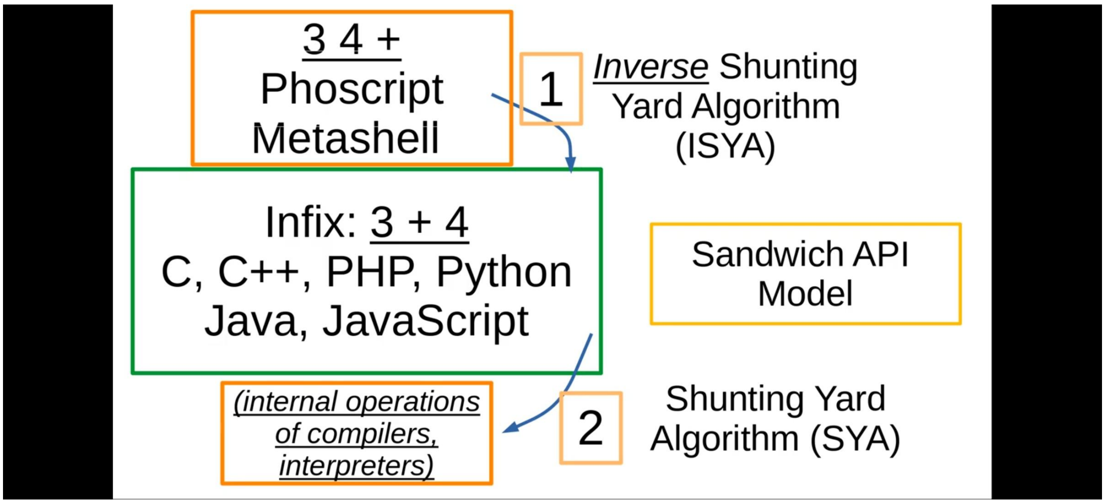
Bidirectional Shunting Yard Algorithm (BiSYA)

Infix: 3 + 4
C, C++
PHP, Python
Java,
JavaScript
…..

Shunting Yard Algorithm (SYA)

Inverse Shunting Yard Algorithm (ISYA)

Reverse Polish Notation
3 4 +
Forth
Phoscript
ΦΩΣ



3 4 +
Phoscript
Metashell

1

Inverse Shunting Yard Algorithm (ISYA)

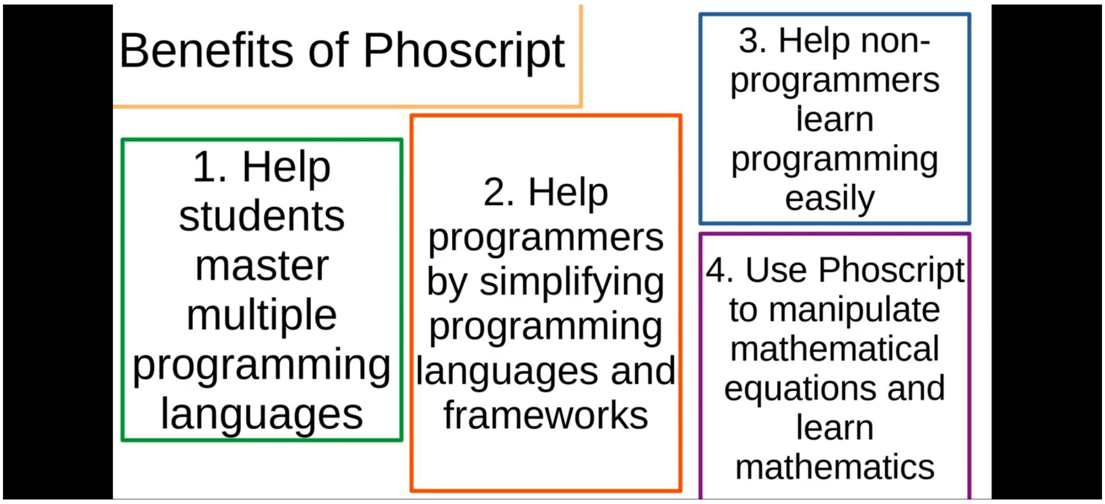
Infix: 3 + 4
C, C++, PHP, Python
Java, JavaScript

Sandwich API Model

(internal operations of compilers, interpreters)

2

Shunting Yard Algorithm (SYA)



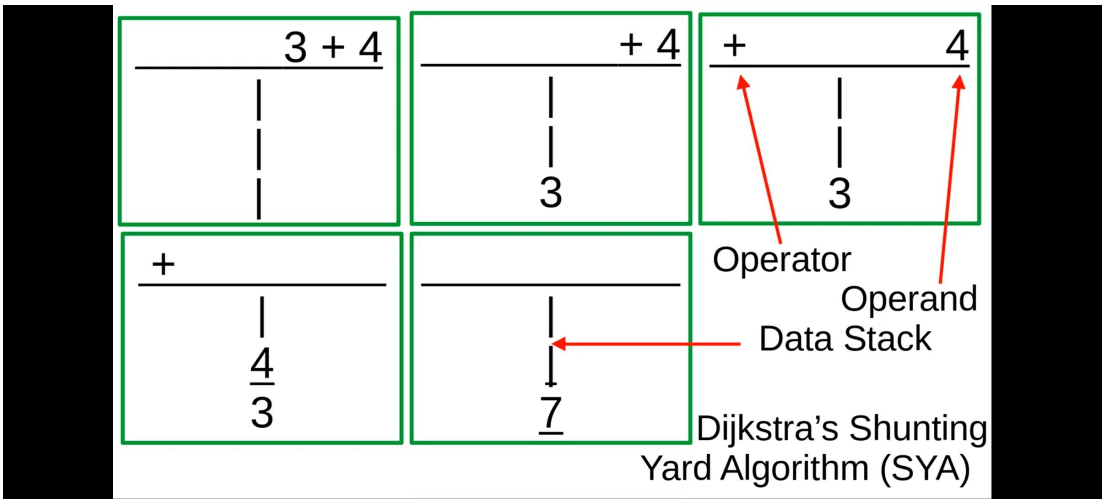
Benefits of Phoscript

1. Help students master multiple programming languages

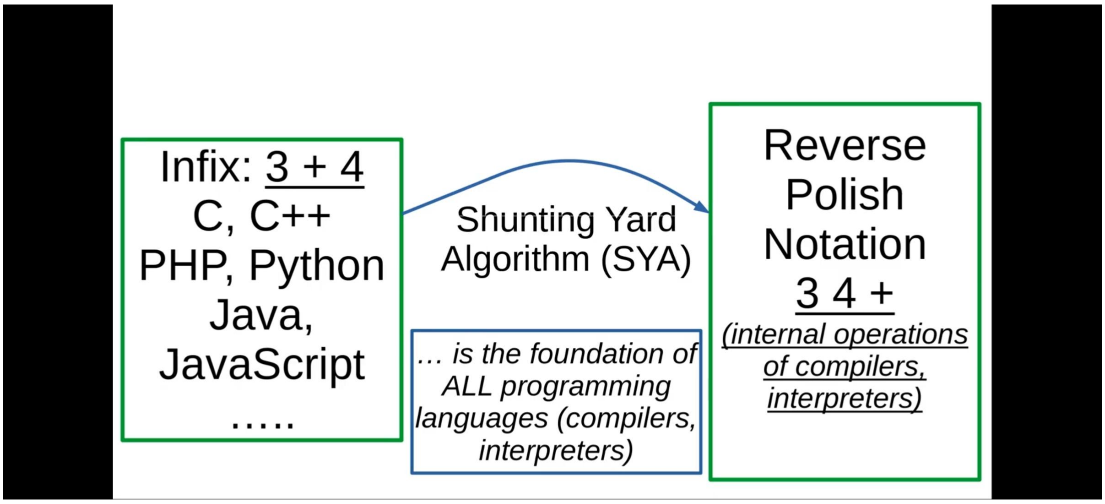
2. Help programmers by simplifying programming languages and frameworks

3. Help non-programmers learn programming easily

4. Use Phoscript to manipulate mathematical equations and learn mathematics



Dijkstra’s Shunting Yard Algorithm (SYA)

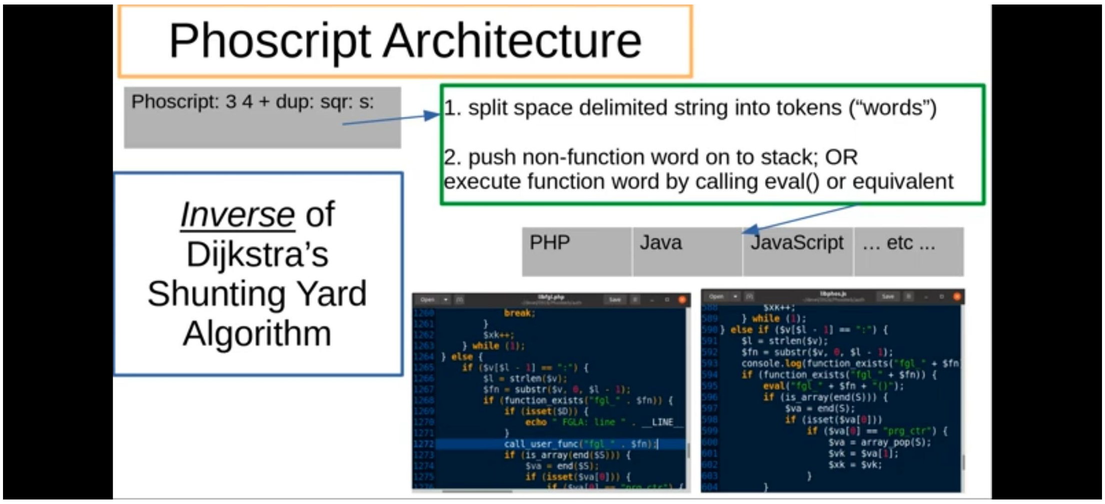


Infix: 3 + 4
C, C++
PHP, Python
Java,
JavaScript
.....

Shunting Yard Algorithm (SYA)

… is the foundation of ALL programming languages (compilers, interpreters)

Reverse Polish Notation
3 4 +
(internal operations of compilers, interpreters)



Phoscript Architecture

Phoscript: 3 4 + dup: sqr: s:

1. split space delimited string into tokens (“words”)

2. push non-function word on to stack; OR
execute function word by calling eval() or equivalent

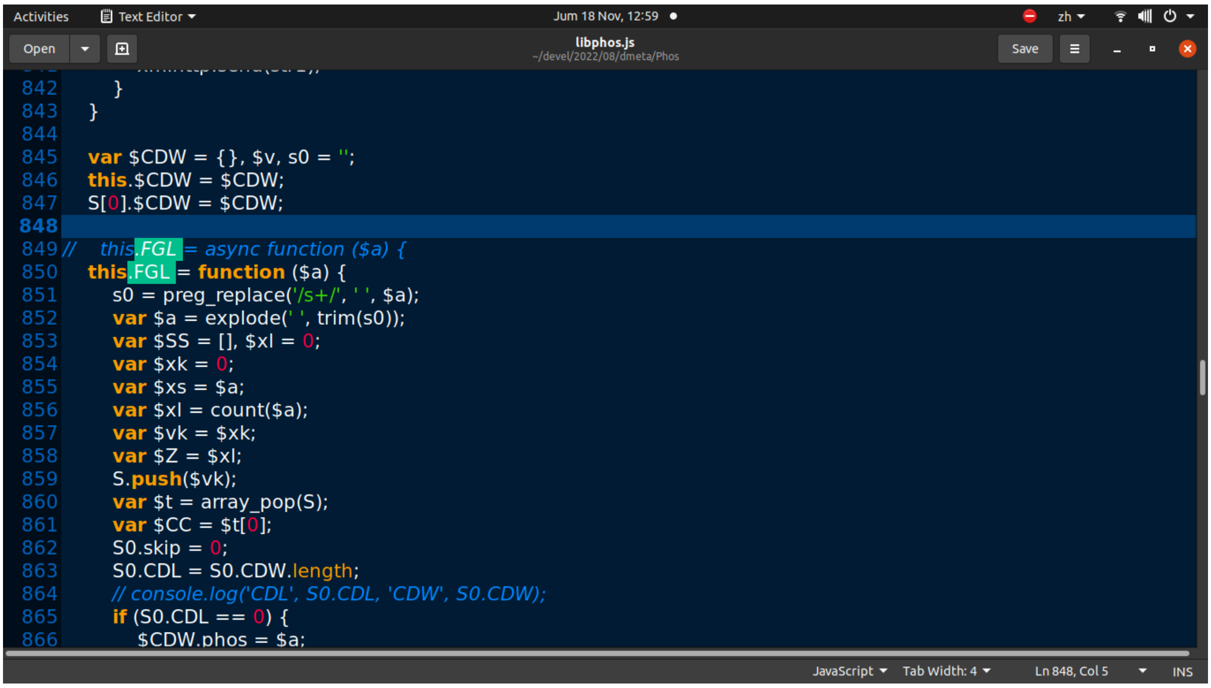
Inverse of Dijkstra’s Shunting Yard Algorithm

PHP	Java	JavaScript	… etc …

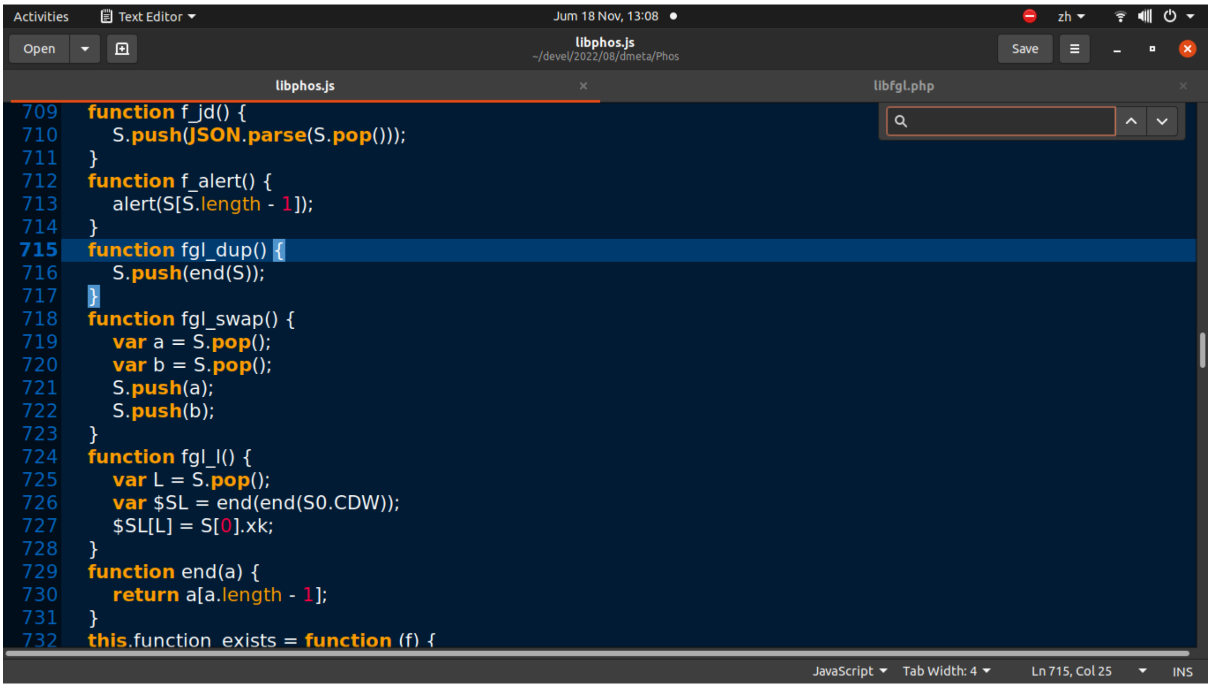
```
                break;
            }
            $xk++;
        } while (1);
    } else {
        if ($v[$l - 1] == ":") {
            $l = strlen($v);
            $fn = substr($v, 0, $l - 1);
            if (function_exists("fgl_" . $fn)) {
                if (isset($D)) {
                    echo " FGLA: line " . __LINE__
                }
                call_user_func("fgl_" . $fn);
                if (is_array(end($S))) {
                    $va = end($S);
                    if (isset($va[0])) {
                        if ($va[0] == "prg_ctr") {
```

```
            $xk++;
        } while (1);
} else if ($v[$l - 1] == ":") {
        $l = strlen($v);
        $fn = substr($v, 0, $l - 1);
        console.log(function_exists("fgl_" + $fn
        if (function_exists("fgl_" + $fn)) {
            eval("fgl_" + $fn + "()");
            if (is_array(end(S))) {
                $va = end(S);
                if (isset($va[0]))
                    if ($va[0] == "prg_ctr") {
                        $va = array_pop(S);
                        $vk = $va[1];
                        $xk = $vk;
                    }
            }
```

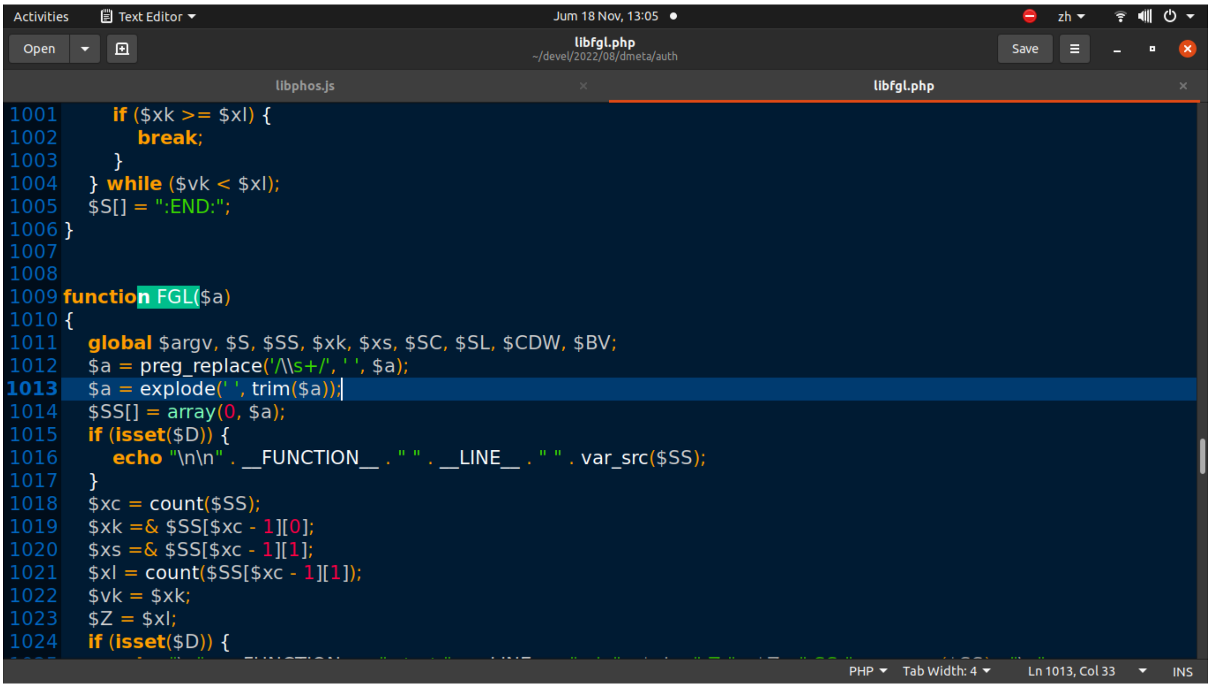
II. Javascript & PHP Implementation of Phoscript



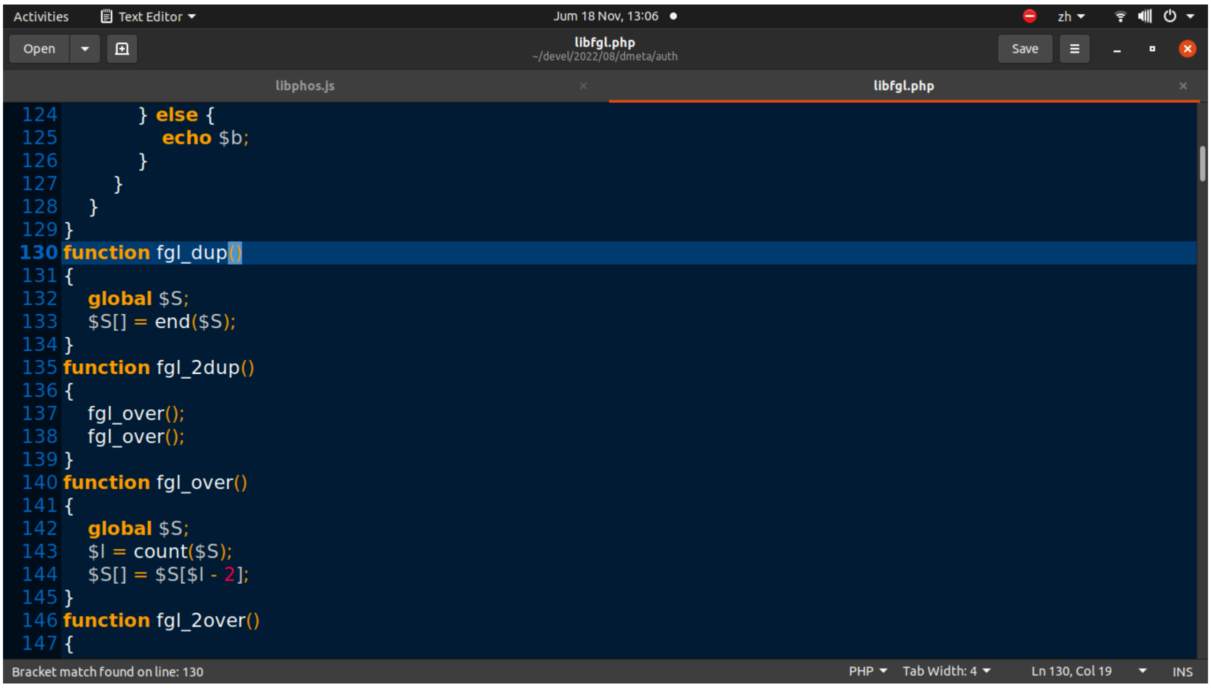
```
        }
    }

    var $CDW = {}, $v, s0 = '';
    this.$CDW = $CDW;
    S[0].$CDW = $CDW;

//   this.FGL = async function ($a) {
    this.FGL = function ($a) {
        s0 = preg_replace('/s+/', ' ', $a);
        var $a = explode(' ', trim(s0));
        var $SS = [], $xl = 0;
        var $xk = 0;
        var $xs = $a;
        var $xl = count($a);
        var $vk = $xk;
        var $Z = $xl;
        S.push($vk);
        var $t = array_pop(S);
        var $CC = $t[0];
        S0.skip = 0;
        S0.CDL = S0.CDW.length;
        // console.log('CDL', S0.CDL, 'CDW', S0.CDW);
        if (S0.CDL == 0) {
            $CDW.phos = $a;
```



```
    function f_jd() {
        S.push(JSON.parse(S.pop()));
    }
    function f_alert() {
        alert(S[S.length - 1]);
    }
    function fgl_dup() {
        S.push(end(S));
    }
    function fgl_swap() {
        var a = S.pop();
        var b = S.pop();
        S.push(a);
        S.push(b);
    }
    function fgl_l() {
        var L = S.pop();
        var $SL = end(end(S0.CDW));
        $SL[L] = S[0].xk;
    }
    function end(a) {
        return a[a.length - 1];
    }
    this.function_exists = function (f) {
```



```
        if ($xk >= $xl) {
            break;
        }
    } while ($vk < $xl);
    $S[] = ":END:";
}


function FGL($a)
{
    global $argv, $S, $SS, $xk, $xs, $SC, $SL, $CDW, $BV;
    $a = preg_replace('/\\s+/', ' ', $a);
    $a = explode(' ', trim($a));
    $SS[] = array(0, $a);
    if (isset($D)) {
        echo "\n\n" . __FUNCTION__ . " " . __LINE__ . " " . var_src($SS);
    }
    $xc = count($SS);
    $xk =& $SS[$xc - 1][0];
    $xs =& $SS[$xc - 1][1];
    $xl = count($SS[$xc - 1][1]);
    $vk = $xk;
    $Z = $xl;
    if (isset($D)) {
```



```
            } else {
                echo $b;
            }
        }
    }
}
function fgl_dup()
{
    global $S;
    $S[] = end($S);
}
function fgl_2dup()
{
    fgl_over();
    fgl_over();
}
function fgl_over()
{
    global $S;
    $l = count($S);
    $S[] = $S[$l - 2];
}
function fgl_2over()
{
```

Bracket match found on line: 130

III. DMeta (Decentralised Metaverse) Server Demo

godmeta.github.io/dmeta/

IV. Examples (YouTube video)

https://www.youtube.com/watch?v=5Fm8e4LC5vo

6m 30s AJAX Phoscript in BOTH front end (JavaScript) and back end (PHP)

7m 00s Simple one cell HTML table using Phoscript

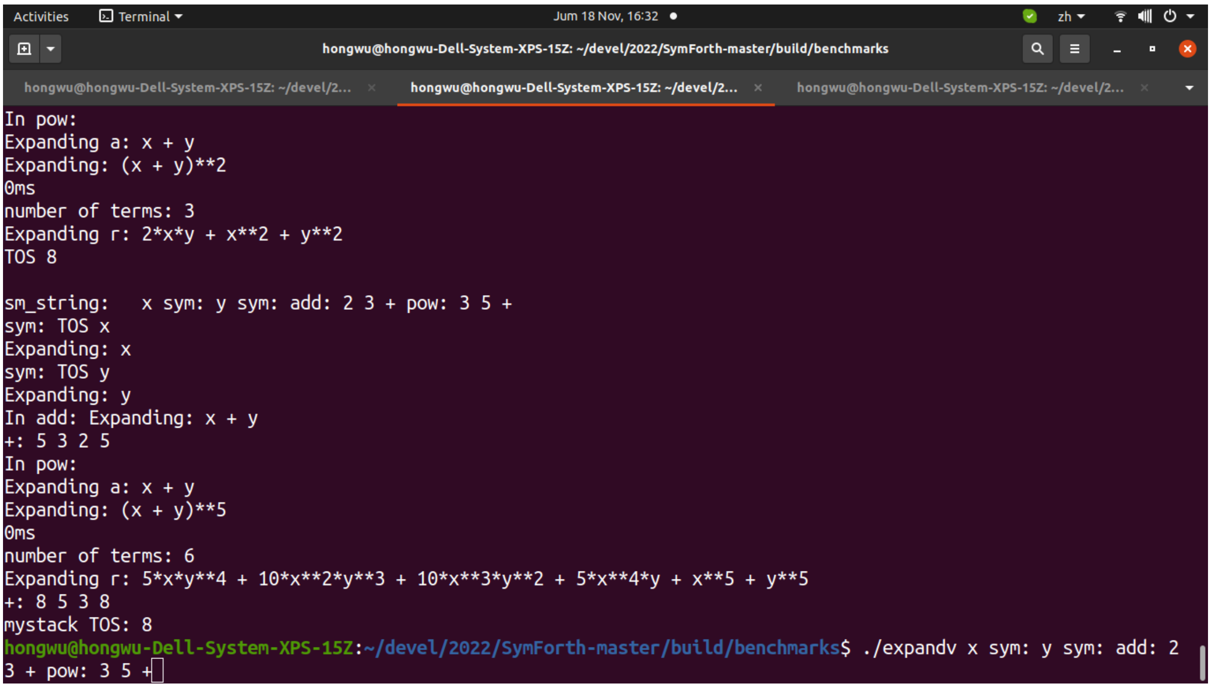
7m 30s Three.js Tank

8m 15s Jitsi Meet Three.js

10m 00s Fire4x (FORTH in Firefox) vs. WebAssembly

10m 45s SymFORTH (SymEngine with Phoscript)

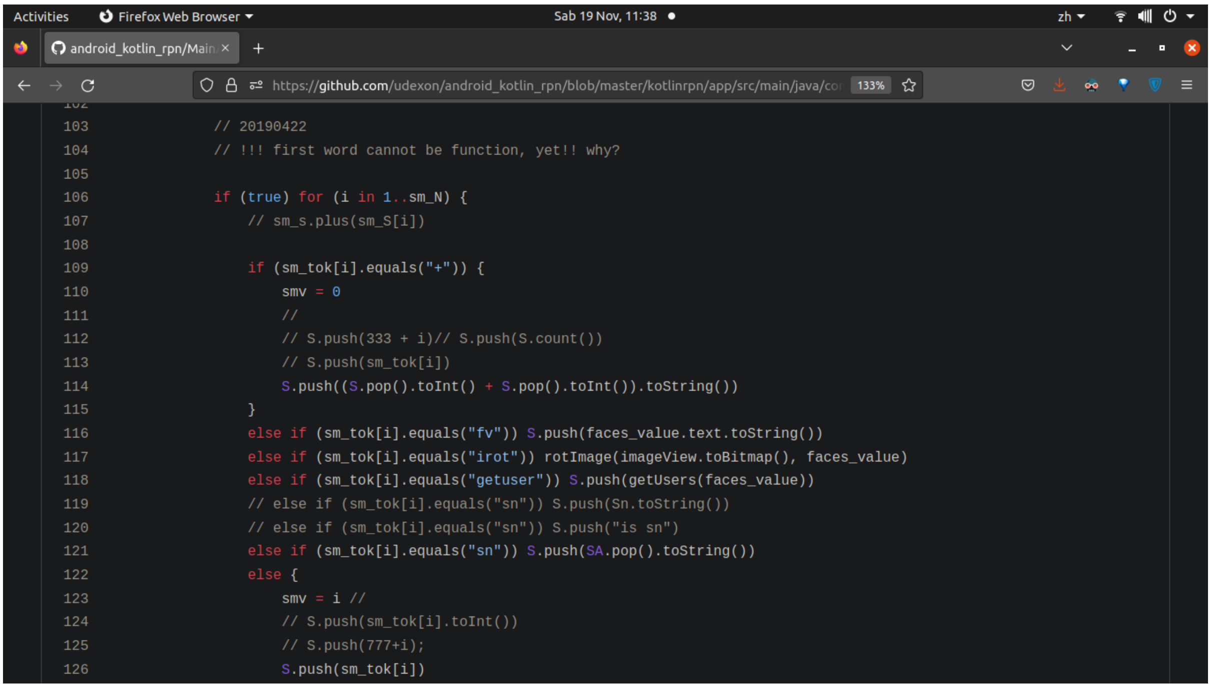
12m 00s BTOSG Bulet OpenSceneGraph Phoscript on C++


Activities
Terminal
Jum 18 Nov, 16:32
zh
hongwu@hongwu-Dell-System-XPS-15Z: ~/devel/2022/SymForth-master/build/benchmarks
hongwu@hongwu-Dell-System-XPS-15Z: ~/devel/2...
hongwu@hongwu-Dell-System-XPS-15Z: ~/devel/2...
hongwu@hongwu-Dell-System-XPS-15Z: ~/devel/2...

```
In pow:
Expanding a: x + y
Expanding: (x + y)**2
0ms
number of terms: 3
Expanding r: 2*x*y + x**2 + y**2
TOS 8

sm_string:   x sym: y sym: add: 2 3 + pow: 3 5 +
sym: TOS x
Expanding: x
sym: TOS y
Expanding: y
In add: Expanding: x + y
+: 5 3 2 5
In pow:
Expanding a: x + y
Expanding: (x + y)**5
0ms
number of terms: 6
Expanding r: 5*x*y**4 + 10*x**2*y**3 + 10*x**3*y**2 + 5*x**4*y + x**5 + y**5
+: 8 5 3 8
mystack TOS: 8
hongwu@hongwu-Dell-System-XPS-15Z:~/devel/2022/SymForth-master/build/benchmarks$ ./expandv x sym: y sym: add: 2
3 + pow: 3 5 +
```

Phoscirpt in Java / Kotlin (Android)

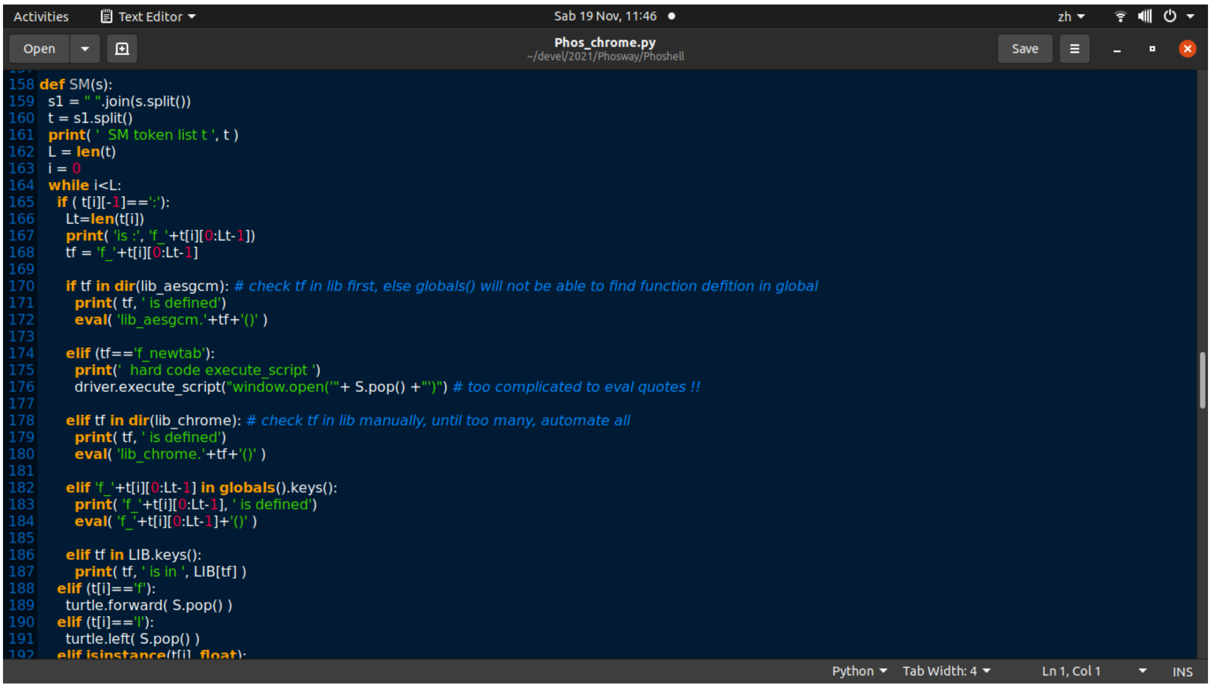
https://github.com/udexon/android_kotlin_rpn/blob/master/kotlinrpn/app/src/main/java/com/projects/facedetector/MainActivity.kt



Activities
Firefox Web Browser
Sab 19 Nov, 11:38
zh
android_kotlin_rpn/Main
https://github.com/udexon/android_kotlin_rpn/blob/master/kotlinrpn/app/src/main/java/co
133%

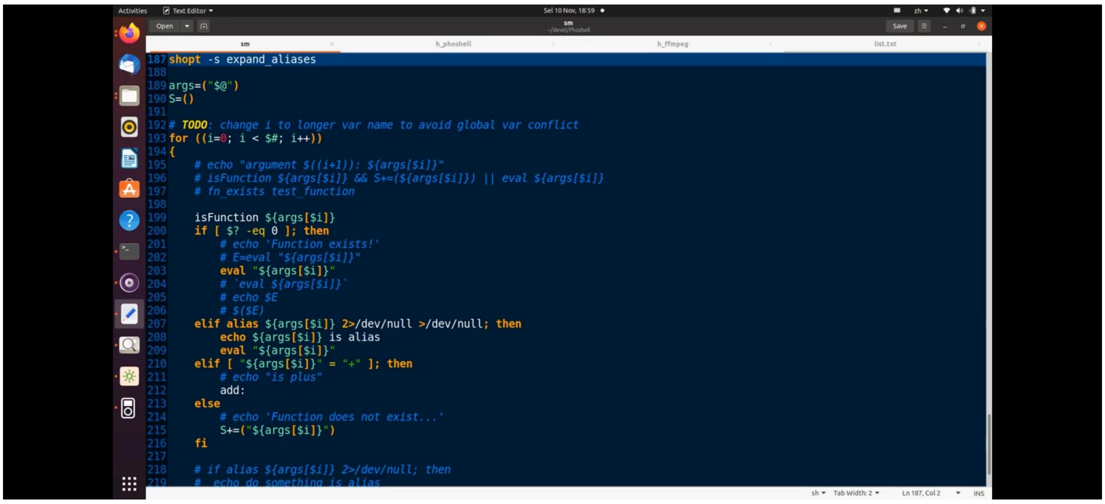
```kotlin
            // 20190422
            // !!! first word cannot be function, yet!! why?

            if (true) for (i in 1..sm_N) {
                // sm_s.plus(sm_S[i])

                if (sm_tok[i].equals("+")) {
                    smv = 0
                    //
                    // S.push(333 + i)// S.push(S.count())
                    // S.push(sm_tok[i])
                    S.push((S.pop().toInt() + S.pop().toInt()).toString())
                }
                else if (sm_tok[i].equals("fv")) S.push(faces_value.text.toString())
                else if (sm_tok[i].equals("irot")) rotImage(imageView.toBitmap(), faces_value)
                else if (sm_tok[i].equals("getuser")) S.push(getUsers(faces_value))
                // else if (sm_tok[i].equals("sn")) S.push(Sn.toString())
                // else if (sm_tok[i].equals("sn")) S.push("is sn")
                else if (sm_tok[i].equals("sn")) S.push(SA.pop().toString())
                else {
                    smv = i //
                    // S.push(sm_tok[i].toInt())
                    // S.push(777+i);
                    S.push(sm_tok[i])
```

Phoscript in Python (Selenium Browser Control Script)

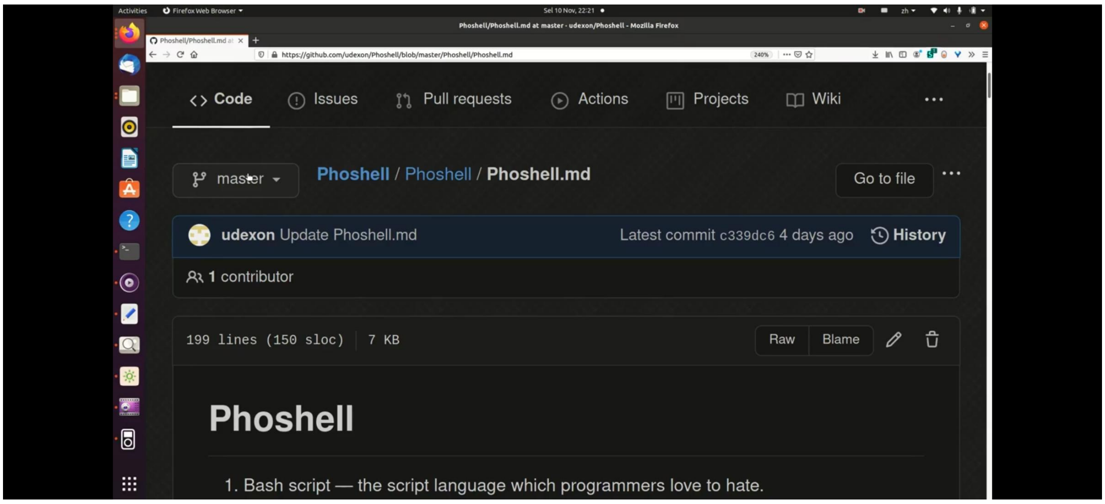


```python
def SM(s):
 s1 = " ".join(s.split())
 t = s1.split()
 print( '  SM token list t ', t )
 L = len(t)
 i = 0
 while i<L:
  if ( t[i][-1]==':'):
   Lt=len(t[i])
   print( 'is :', 'f_'+t[i][0:Lt-1])
   tf = 'f_'+t[i][0:Lt-1]

   if tf in dir(lib_aesgcm): # check tf in lib first, else globals() will not be able to find function defition in global
    print( tf, ' is defined')
    eval( 'lib_aesgcm.'+tf+'()' )

   elif (tf=='f_newtab'):
    print('  hard code execute_script ')
    driver.execute_script("window.open('"+ S.pop() +"')") # too complicated to eval quotes !!

   elif tf in dir(lib_chrome): # check tf in lib manually, until too many, automate all
    print( tf, ' is defined')
    eval( 'lib_chrome.'+tf+'()' )

   elif 'f_'+t[i][0:Lt-1] in globals().keys():
    print( 'f_'+t[i][0:Lt-1], ' is defined')
    eval( 'f_'+t[i][0:Lt-1]+'()' )

   elif tf in LIB.keys():
    print( tf, ' is in ', LIB[tf] )
  elif (t[i]=='f'):
   turtle.forward( S.pop() )
  elif (t[i]=='l'):
   turtle.left( S.pop() )
  elif isinstance(t[i], float):
```

Phoshell (Phoscipt in Bash Shell)



```
shopt -s expand_aliases

args=("$@")
S=()

# TODO: change i to longer var name to avoid global var conflict
for ((i=0; i < $#; i++))
{
    # echo "argument $((i+1)): ${args[$i]}"
    # isFunction ${args[$i]} && S+=(${args[$i]}) || eval ${args[$i]}
    # fn_exists test_function

    isFunction ${args[$i]}
    if [ $? -eq 0 ]; then
        # echo 'Function exists!'
        # E=eval "${args[$i]}"
        eval "${args[$i]}"
        # `eval ${args[$i]}`
        # echo $E
        # $($E)
    elif alias ${args[$i]} 2>/dev/null >/dev/null; then
        echo ${args[$i]} is alias
        eval "${args[$i]}"
    elif [ "${args[$i]}" = "+" ]; then
        # echo "is plus"
        add:
    else
        # echo 'Function does not exist...'
        S+=("${args[$i]}")
    fi

    # if alias ${args[$i]} 2>/dev/null; then
    #  echo do something is alias
```



Phoshell / Phoshell / Phoshell.md

udexon Update Phoshell.md Latest commit c339dc6 4 days ago History

1 contributor

199 lines (150 sloc) 7 KB

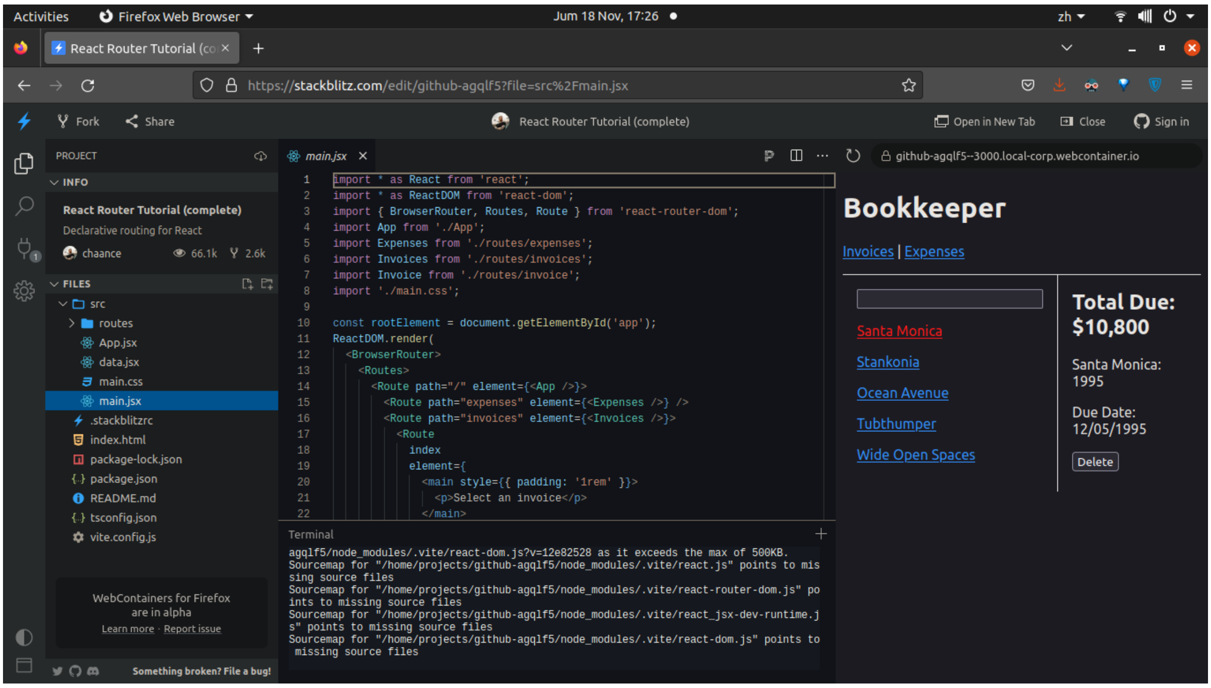
Phoshell

1. Bash script — the script language which programmers love to hate.

V. Reverse React Notation

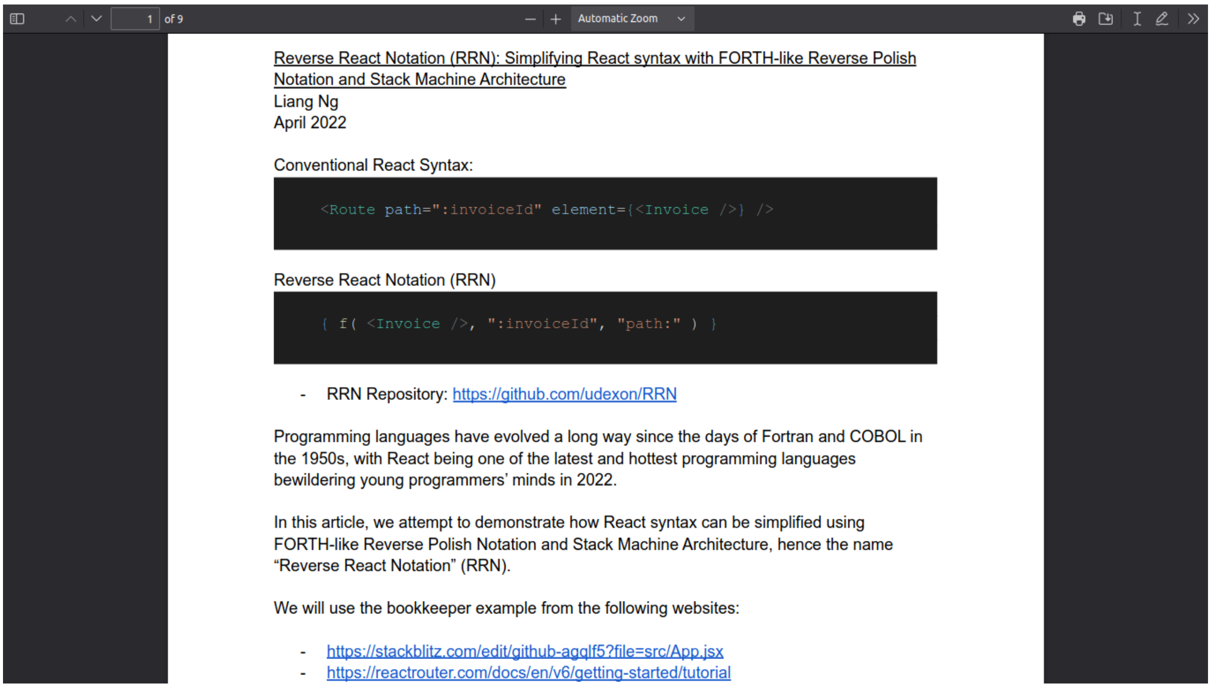
https://github.com/udexon/RRN

https://stackblitz.com/edit/github-agqlf5?file=src%2FApp.jsx



```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import App from './App';
import Expenses from './routes/expenses';
import Invoices from './routes/invoices';
import Invoice from './routes/invoice';
import './main.css';

const rootElement = document.getElementById('app');
ReactDOM.render(
  <BrowserRouter>
    <Routes>
      <Route path="/" element={<App />}>
        <Route path="expenses" element={<Expenses />} />
        <Route path="invoices" element={<Invoices />}>
          <Route
            index
            element={
              <main style={{ padding: '1rem' }}>
                <p>Select an invoice</p>
              </main>
```

```
agqlf5/node_modules/.vite/react-dom.js?v=12e82528 as it exceeds the max of 500KB.
Sourcemap for "/home/projects/github-agqlf5/node_modules/.vite/react.js" points to mis
sing source files
Sourcemap for "/home/projects/github-agqlf5/node_modules/.vite/react-router-dom.js" po
ints to missing source files
Sourcemap for "/home/projects/github-agqlf5/node_modules/.vite/react_jsx-dev-runtime.j
s" points to missing source files
Sourcemap for "/home/projects/github-agqlf5/node_modules/.vite/react-dom.js" points to
 missing source files
```



Reverse React Notation (RRN): Simplifying React syntax with FORTH-like Reverse Polish Notation and Stack Machine Architecture
Liang Ng
April 2022

Conventional React Syntax:

```
<Route path=":invoiceId" element={<Invoice />} />
```

Reverse React Notation (RRN)

```
{ f( <Invoice />, ":invoiceId", "path:" ) }
```

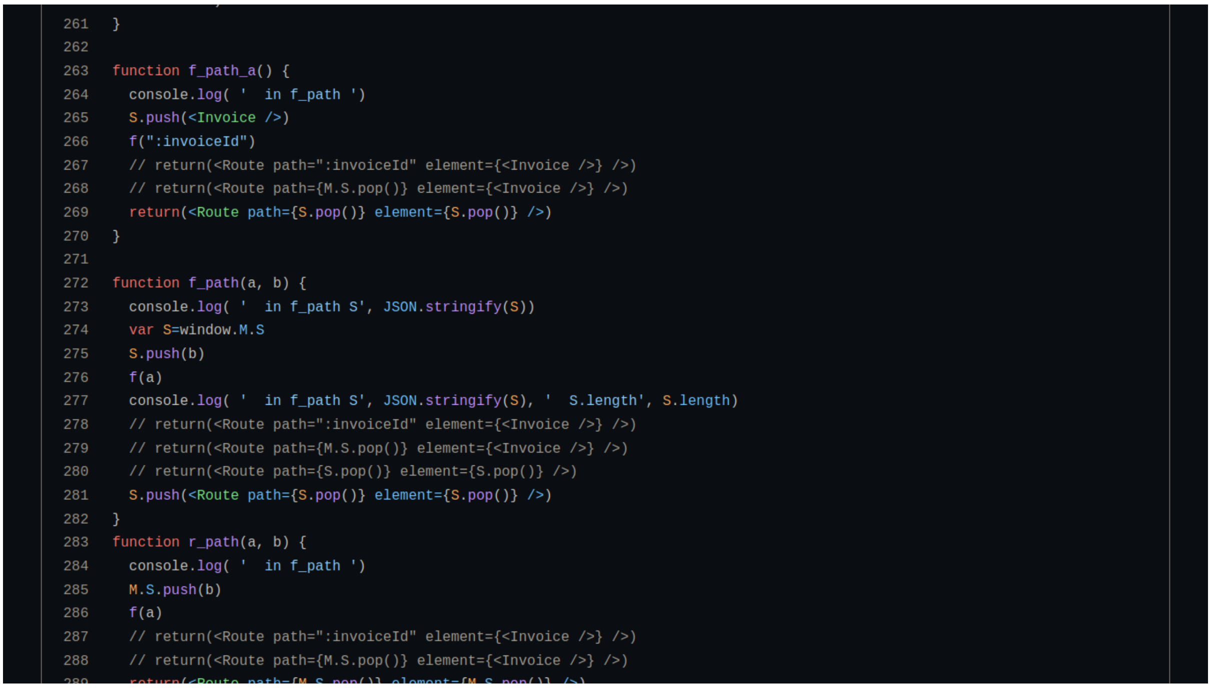
- RRN Repository: https://github.com/udexon/RRN

Programming languages have evolved a long way since the days of Fortran and COBOL in the 1950s, with React being one of the latest and hottest programming languages bewildering young programmers' minds in 2022.

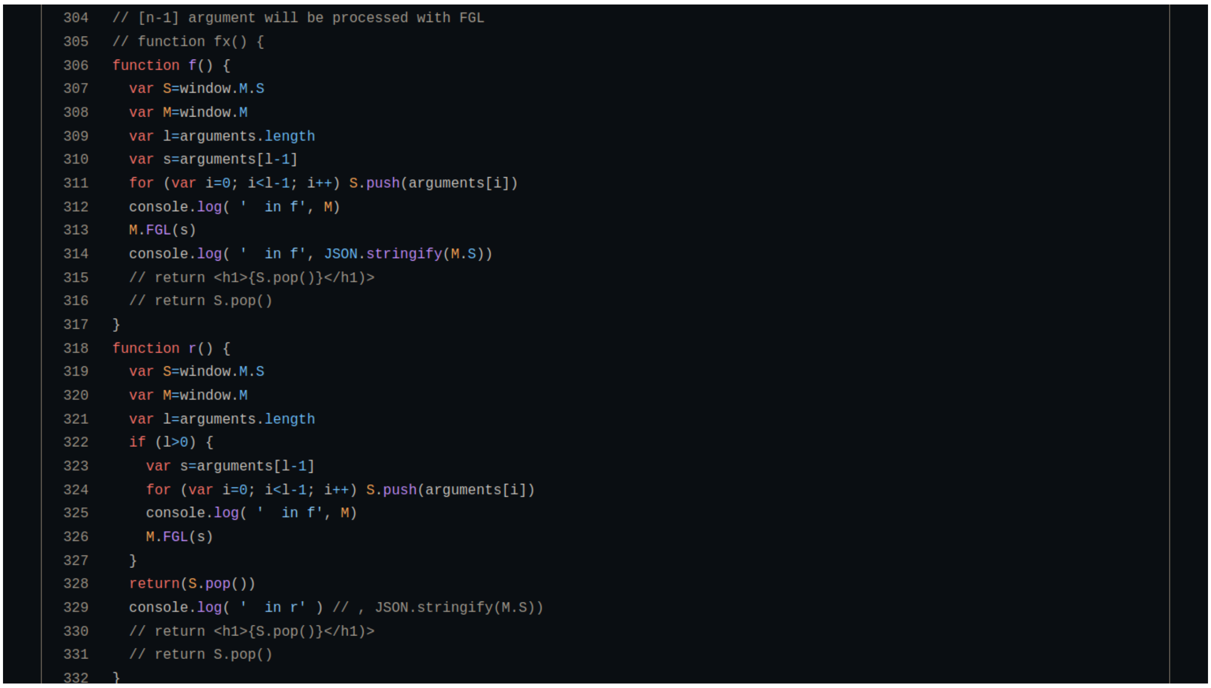
In this article, we attempt to demonstrate how React syntax can be simplified using FORTH-like Reverse Polish Notation and Stack Machine Architecture, hence the name "Reverse React Notation" (RRN).

We will use the bookkeeper example from the following websites:

- https://stackblitz.com/edit/github-agqlf5?file=src/App.jsx
- https://reactrouter.com/docs/en/v6/getting-started/tutorial

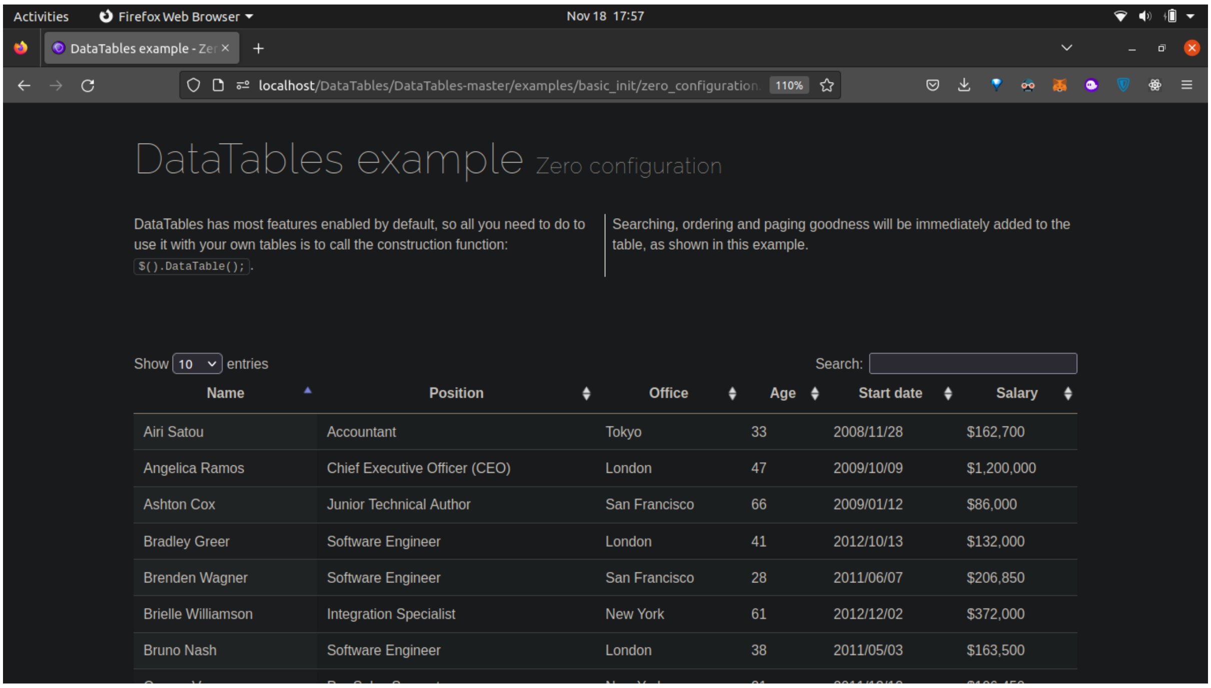


```
}

function f_path_a() {
  console.log( '  in f_path ')
  S.push(<Invoice />)
  f(":invoiceId")
  // return(<Route path=":invoiceId" element={<Invoice />} />)
  // return(<Route path={M.S.pop()} element={<Invoice />} />)
  return(<Route path={S.pop()} element={S.pop()} />)
}

function f_path(a, b) {
  console.log( '  in f_path S', JSON.stringify(S))
  var S=window.M.S
  S.push(b)
  f(a)
  console.log( '  in f_path S', JSON.stringify(S), '  S.length', S.length)
  // return(<Route path=":invoiceId" element={<Invoice />} />)
  // return(<Route path={M.S.pop()} element={<Invoice />} />)
  // return(<Route path={S.pop()} element={S.pop()} />)
  S.push(<Route path={S.pop()} element={S.pop()} />)
}
function r_path(a, b) {
  console.log( '  in f_path ')
  M.S.push(b)
  f(a)
  // return(<Route path=":invoiceId" element={<Invoice />} />)
  // return(<Route path={M.S.pop()} element={<Invoice />} />)
  return(<Route path={M.S.pop()} element={M.S.pop()} />)
```



```
// [n-1] argument will be processed with FGL
// function fx() {
function f() {
  var S=window.M.S
  var M=window.M
  var l=arguments.length
  var s=arguments[l-1]
  for (var i=0; i<l-1; i++) S.push(arguments[i])
  console.log( '  in f', M)
  M.FGL(s)
  console.log( '  in f', JSON.stringify(M.S))
  // return <h1>{S.pop()}</h1>
  // return S.pop()
}
function r() {
  var S=window.M.S
  var M=window.M
  var l=arguments.length
  if (l>0) {
    var s=arguments[l-1]
    for (var i=0; i<l-1; i++) S.push(arguments[i])
    console.log( '  in f', M)
    M.FGL(s)
  }
  return(S.pop())
  console.log( '  in r' ) // , JSON.stringify(M.S))
  // return <h1>{S.pop()}</h1>
  // return S.pop()
}
```

VI. Freddiscript (Find-Regular-Expression-Edit Script)


Activities
Firefox Web Browser
Nov 18 17:57
DataTables example - Ze
localhost/DataTables/DataTables-master/examples/basic_init/zero_configuration
110%

DataTables example Zero configuration

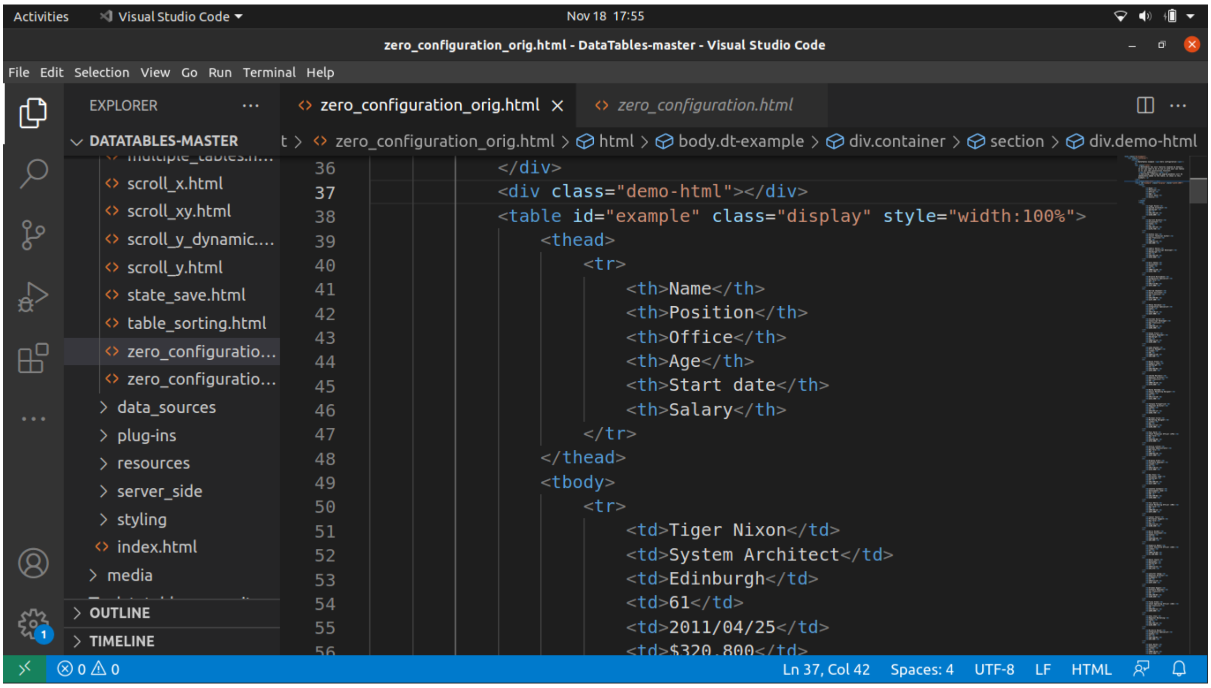
DataTables has most features enabled by default, so all you need to do to use it with your own tables is to call the construction function: `$().DataTable();`.

Searching, ordering and paging goodness will be immediately added to the table, as shown in this example.

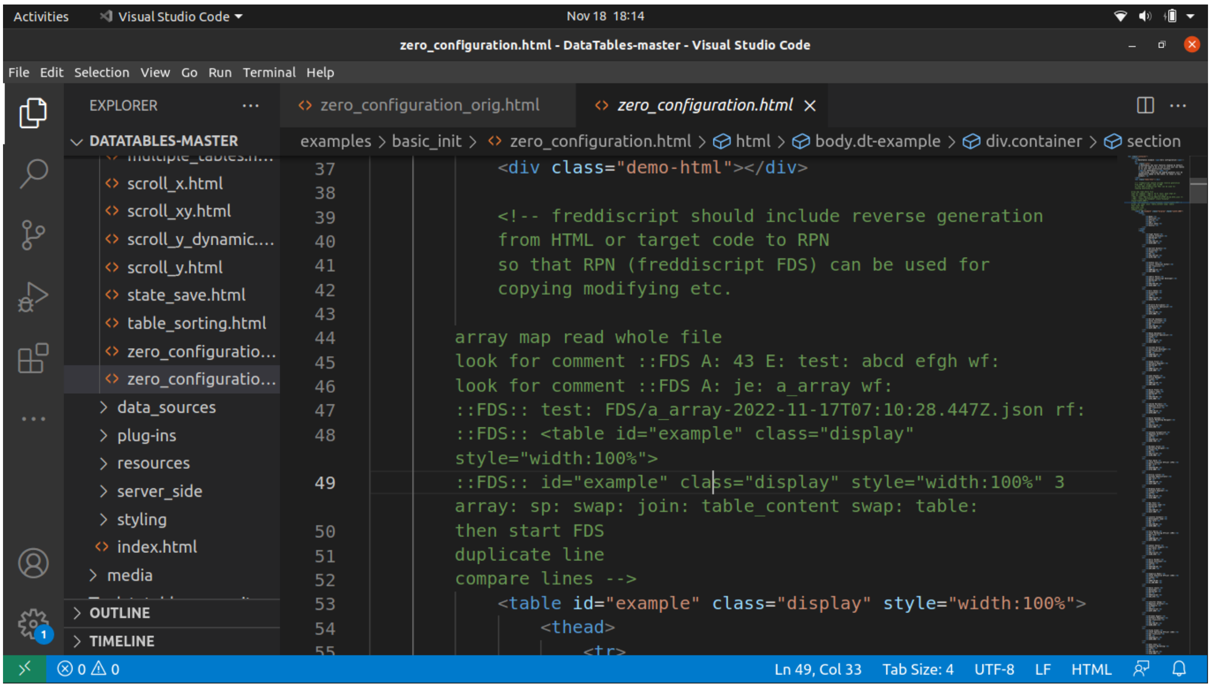
Show 10 entries

Search:

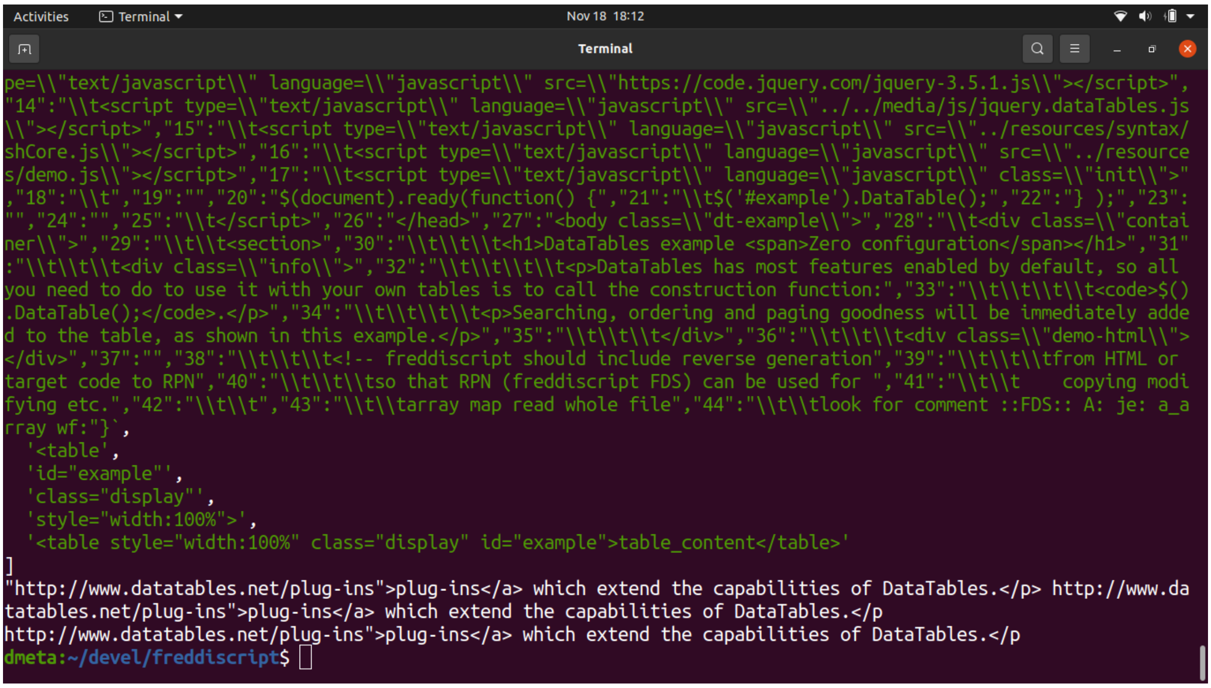
Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	$163,500



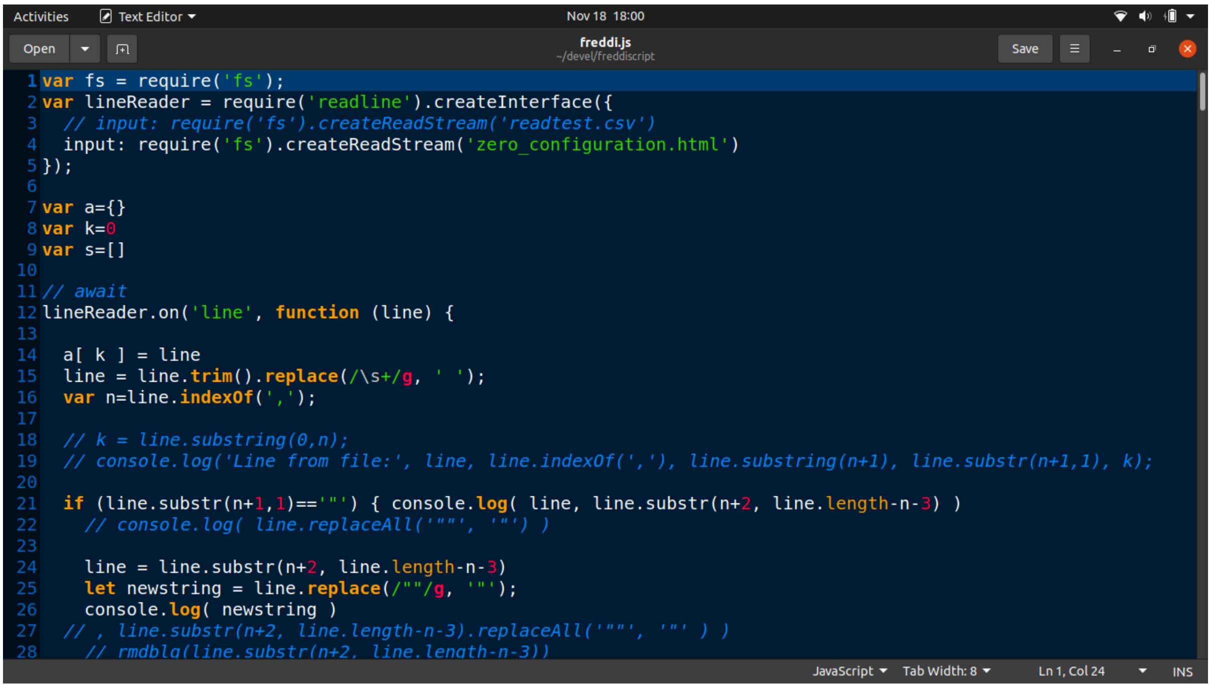
```
            </div>
            <div class="demo-html"></div>
            <table id="example" class="display" style="width:100%">
                <thead>
                    <tr>
                        <th>Name</th>
                        <th>Position</th>
                        <th>Office</th>
                        <th>Age</th>
                        <th>Start date</th>
                        <th>Salary</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>Tiger Nixon</td>
                        <td>System Architect</td>
                        <td>Edinburgh</td>
                        <td>61</td>
                        <td>2011/04/25</td>
                        <td>$320,800</td>
```



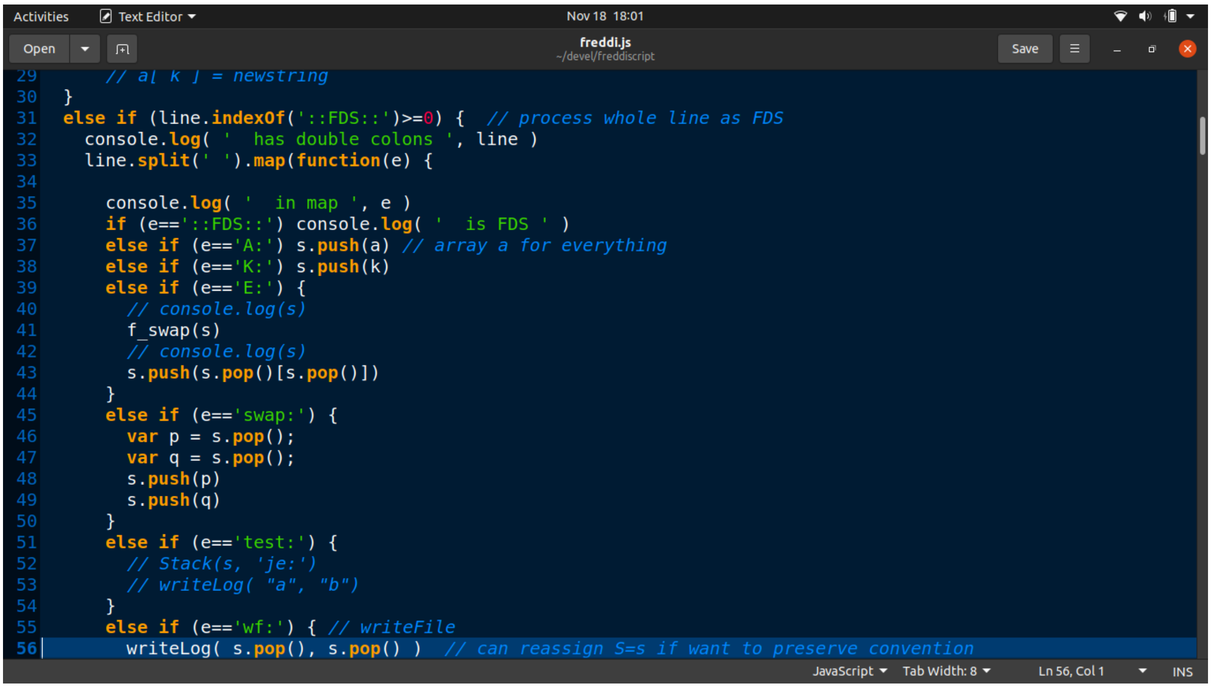
```
            <div class="demo-html"></div>

            <!-- freddiscript should include reverse generation
            from HTML or target code to RPN
            so that RPN (freddiscript FDS) can be used for
            copying modifying etc.

        array map read whole file
        look for comment ::FDS A: 43 E: test: abcd efgh wf:
        look for comment ::FDS A: je: a_array wf:
        ::FDS:: test: FDS/a_array-2022-11-17T07:10:28.447Z.json rf:
        ::FDS:: <table id="example" class="display"
        style="width:100%">
        ::FDS:: id="example" class="display" style="width:100%" 3
        array: sp: swap: join: table_content swap: table:
        then start FDS
        duplicate line
        compare lines -->
            <table id="example" class="display" style="width:100%">
                <thead>
                    <tr>
```



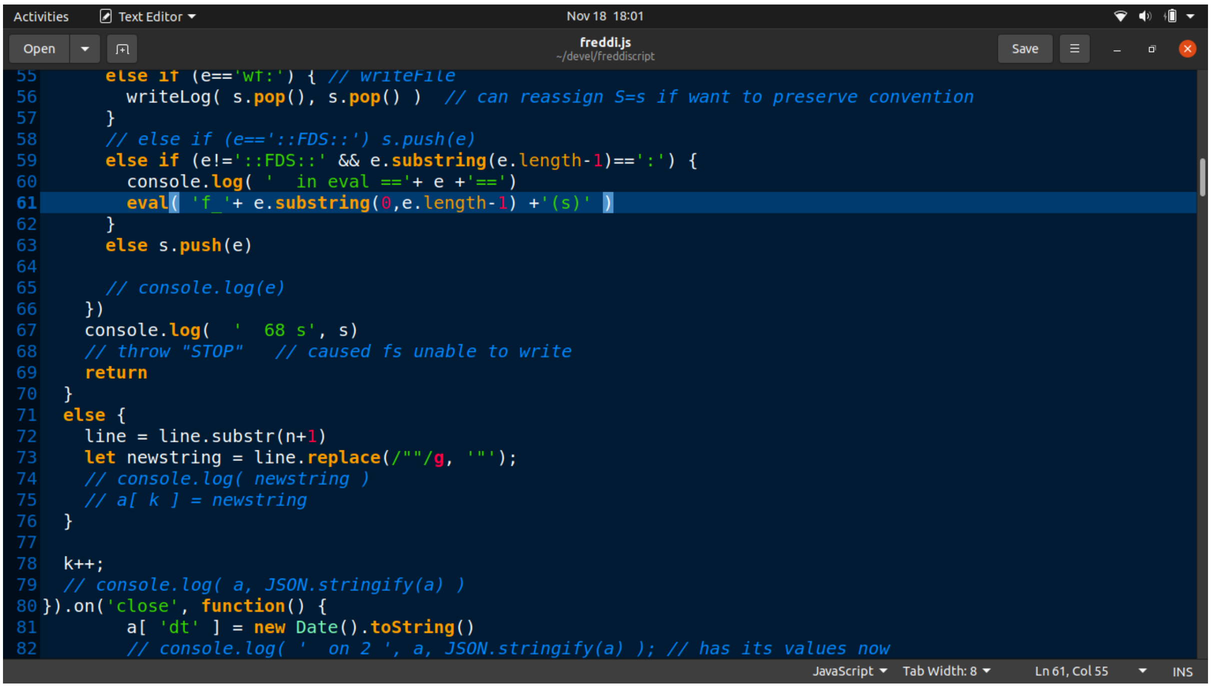
```
pe=\\"text/javascript\\" language=\\"javascript\\" src=\\"https://code.jquery.com/jquery-3.5.1.js\\"></script>",
"14":"\\t<script type=\\"text/javascript\\" language=\\"javascript\\" src=\\"../../media/js/jquery.dataTables.js
\\"></script>","15":"\\t<script type=\\"text/javascript\\" language=\\"javascript\\" src=\\"../resources/syntax/
shCore.js\\"></script>","16":"\\t<script type=\\"text/javascript\\" language=\\"javascript\\" src=\\"../resource
s/demo.js\\"></script>","17":"\\t<script type=\\"text/javascript\\" language=\\"javascript\\" class=\\"init\\">"
,"18":"\\t","19":"","20":"$(document).ready(function() {","21":"\\t$('#example').DataTable();","22":"} );","23":
"","24":"","25":"\\t</script>","26":"</head>","27":"<body class=\\"dt-example\\">","28":"\\t<div class=\\"contai
ner\\">","29":"\\t\\t<section>","30":"\\t\\t\\t<h1>DataTables example <span>Zero configuration</span></h1>","31"
:"\\t\\t\\t<div class=\\"info\\">","32":"\\t\\t\\t\\t<p>DataTables has most features enabled by default, so all
you need to do to use it with your own tables is to call the construction function:","33":"\\t\\t\\t\\t<code>$()
.DataTable();</code>.</p>","34":"\\t\\t\\t\\t<p>Searching, ordering and paging goodness will be immediately adde
d to the table, as shown in this example.</p>","35":"\\t\\t\\t</div>","36":"\\t\\t\\t<div class=\\"demo-html\\">
</div>","37":"","38":"\\t\\t\\t<!-- freddiscript should include reverse generation","39":"\\t\\t\\tfrom HTML or
target code to RPN","40":"\\t\\t\\tso that RPN (freddiscript FDS) can be used for ","41":"\\t\\t    copying modi
fying etc.","42":"\\t\\t","43":"\\t\\tarray map read whole file","44":"\\t\\tlook for comment ::FDS:: A: je: a_a
rray wf:"}`,
  '<table',
  'id="example"',
  'class="display"',
  'style="width:100%">',
  '<table style="width:100%" class="display" id="example">table_content</table>'
]
"http://www.datatables.net/plug-ins">plug-ins</a> which extend the capabilities of DataTables.</p> http://www.da
tatables.net/plug-ins">plug-ins</a> which extend the capabilities of DataTables.</p
http://www.datatables.net/plug-ins">plug-ins</a> which extend the capabilities of DataTables.</p
dmeta:~/devel/freddiscript$ 
```



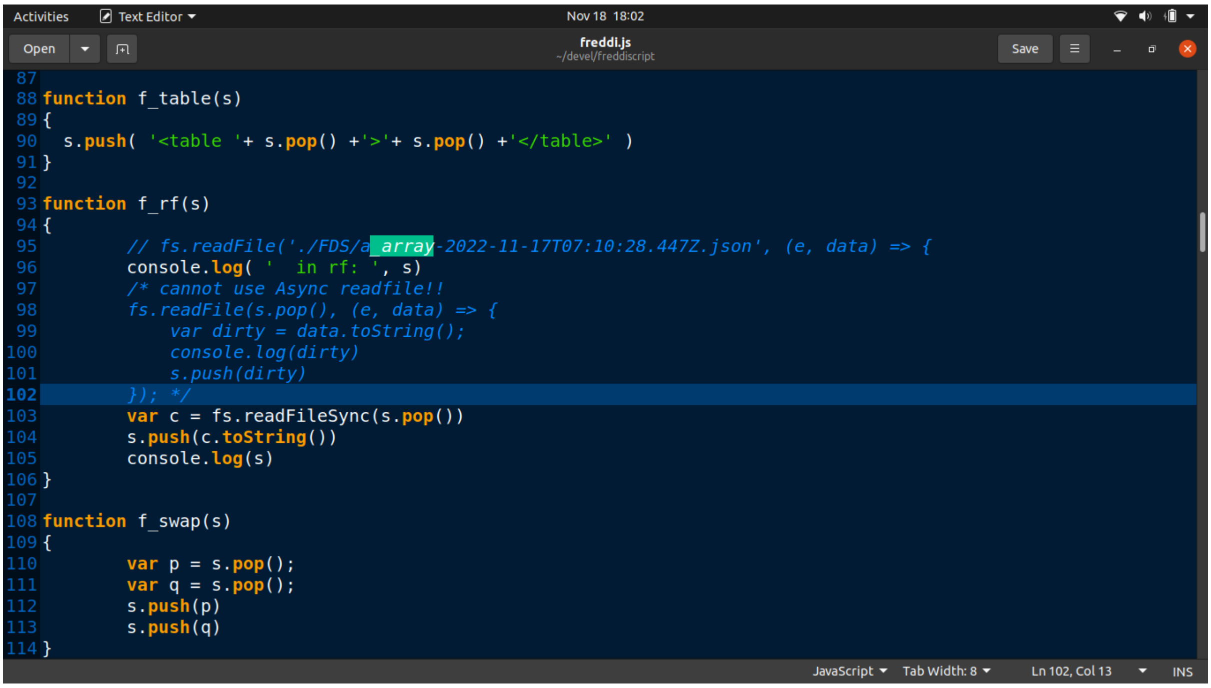
```
var fs = require('fs');
var lineReader = require('readline').createInterface({
  // input: require('fs').createReadStream('readtest.csv')
  input: require('fs').createReadStream('zero_configuration.html')
});

var a={}
var k=0
var s=[]

// await
lineReader.on('line', function (line) {

  a[ k ] = line
  line = line.trim().replace(/\s+/g, ' ');
  var n=line.indexOf(',');

  // k = line.substring(0,n);
  // console.log('Line from file:', line, line.indexOf(','), line.substring(n+1), line.substr(n+1,1), k);

  if (line.substr(n+1,1)=='"') { console.log( line, line.substr(n+2, line.length-n-3) )
    // console.log( line.replaceAll('""', '"') )

    line = line.substr(n+2, line.length-n-3)
    let newstring = line.replace(/""/g, '"');
    console.log( newstring )
  // , line.substr(n+2, line.length-n-3).replaceAll('""', '"' ) )
    // rmdblq(line.substr(n+2, line.length-n-3))
```



```
    // a[ k ] = newstring
  }
  else if (line.indexOf('::FDS::')>=0) {  // process whole line as FDS
    console.log( '  has double colons ', line )
    line.split(' ').map(function(e) {

      console.log( '  in map ', e )
      if (e=='::FDS::') console.log( '  is FDS ' )
      else if (e=='A:') s.push(a) // array a for everything
      else if (e=='K:') s.push(k)
      else if (e=='E:') {
        // console.log(s)
        f_swap(s)
        // console.log(s)
        s.push(s.pop()[s.pop()])
      }
      else if (e=='swap:') {
        var p = s.pop();
        var q = s.pop();
        s.push(p)
        s.push(q)
      }
      else if (e=='test:') {
        // Stack(s, 'je:')
        // writeLog( "a", "b")
      }
      else if (e=='wf:') { // writeFile
        writeLog( s.pop(), s.pop() )  // can reassign S=s if want to preserve convention
```



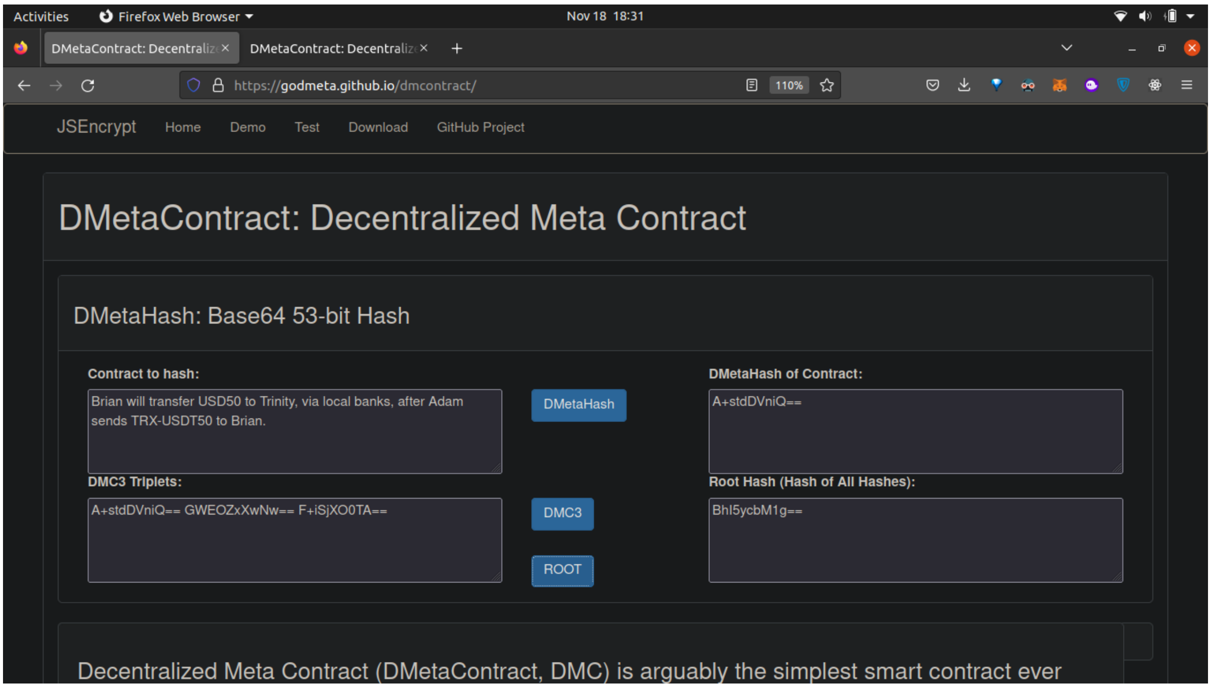
```
   else if (e=='wf:') { // writeFile
    writeLog( s.pop(), s.pop() )  // can reassign S=s if want to preserve convention
   }
   // else if (e=='::FDS::') s.push(e)
   else if (e!='::FDS::' && e.substring(e.length-1)==':') {
    console.log( '  in eval =='+ e +'==')
    eval( 'f_'+ e.substring(0,e.length-1) +'(s)' )
   }
   else s.push(e)

   // console.log(e)
  })
  console.log(  '  68 s', s)
  // throw "STOP"   // caused fs unable to write
  return
 }
 else {
  line = line.substr(n+1)
  let newstring = line.replace(/""/g, '"');
  // console.log( newstring )
  // a[ k ] = newstring
 }

 k++;
 // console.log( a, JSON.stringify(a) )
}).on('close', function() {
    a[ 'dt' ] = new Date().toString()
    // console.log( '  on 2 ', a, JSON.stringify(a) ); // has its values now
```



```

function f_table(s)
{
 s.push( '<table '+ s.pop() +'>'+ s.pop() +'</table>' )
}

function f_rf(s)
{
    // fs.readFile('./FDS/a_array-2022-11-17T07:10:28.447Z.json', (e, data) => {
    console.log( '  in rf: ', s)
    /* cannot use Async readfile!!
    fs.readFile(s.pop(), (e, data) => {
      var dirty = data.toString();
      console.log(dirty)
      s.push(dirty)
    }); */
    var c = fs.readFileSync(s.pop())
    s.push(c.toString())
    console.log(s)
}

function f_swap(s)
{
    var p = s.pop();
    var q = s.pop();
    s.push(p)
    s.push(q)
}
```

VII. DMeta Code Hash

https://godmeta.github.io/dmcontract/



DMetaContract: Decentralized Meta Contract

DMetaHash: Base64 53-bit Hash

Contract to hash:

Brian will transfer USD50 to Trinity, via local banks, after Adam sends TRX-USDT50 to Brian.

DMetaHash of Contract:

A+stdDVniQ==

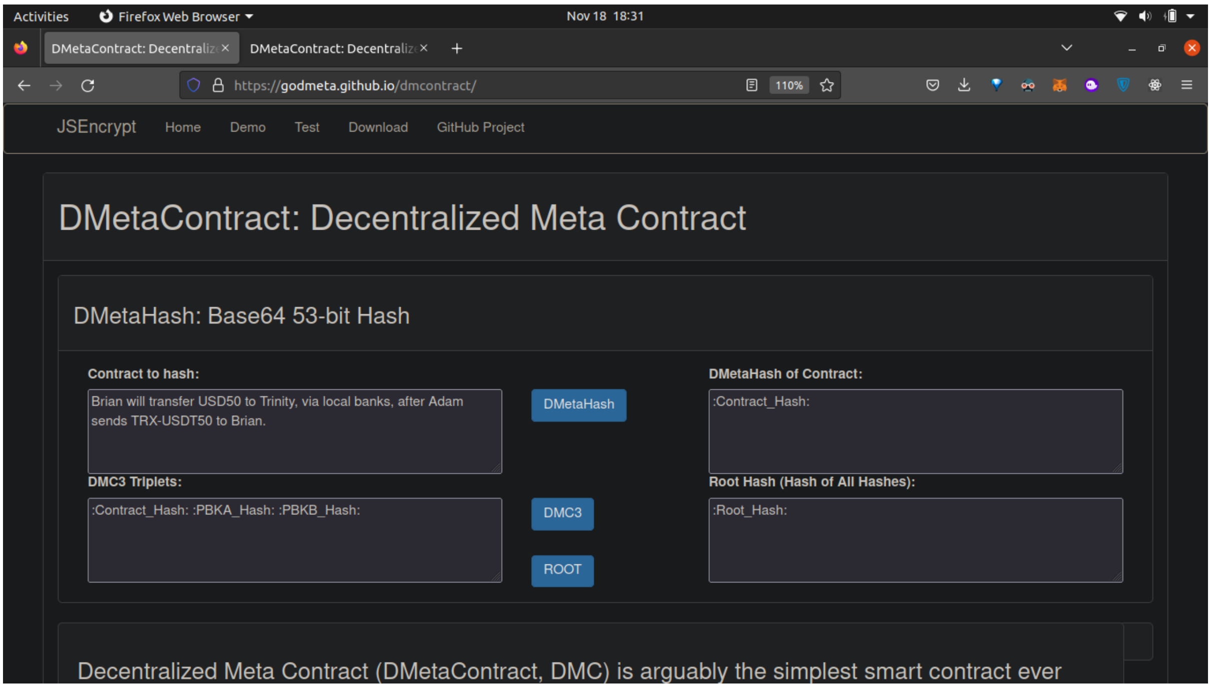
DMC3 Triplets:

A+stdDVniQ== GWEOZxXwNw== F+iSjXO0TA==

Root Hash (Hash of All Hashes):

Bhl5ycbM1g==

Decentralized Meta Contract (DMetaContract, DMC) is arguably the simplest smart contract ever



DMetaContract: Decentralized Meta Contract

DMetaHash: Base64 53-bit Hash

Contract to hash:

Brian will transfer USD50 to Trinity, via local banks, after Adam sends TRX-USDT50 to Brian.

DMetaHash of Contract:

:Contract_Hash:

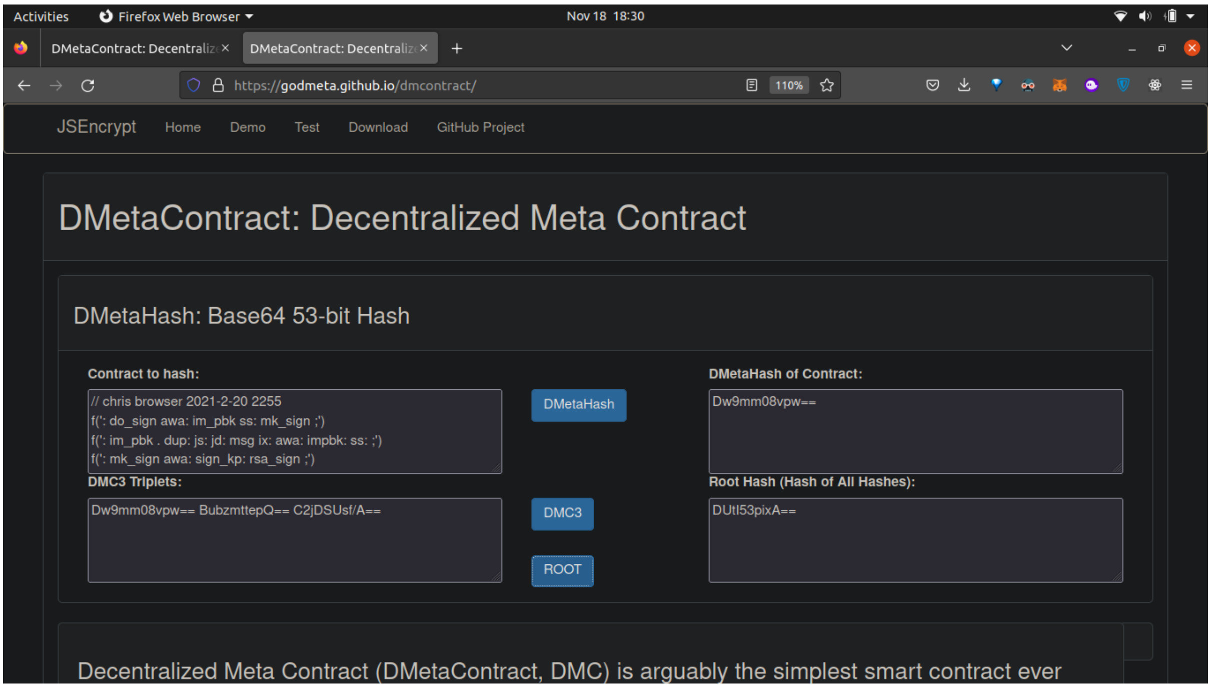
DMC3 Triplets:

:Contract_Hash: :PBKA_Hash: :PBKB_Hash:

Root Hash (Hash of All Hashes):

:Root_Hash:

Decentralized Meta Contract (DMetaContract, DMC) is arguably the simplest smart contract ever



DMetaContract: Decentralized Meta Contract

DMetaHash: Base64 53-bit Hash

Contract to hash:

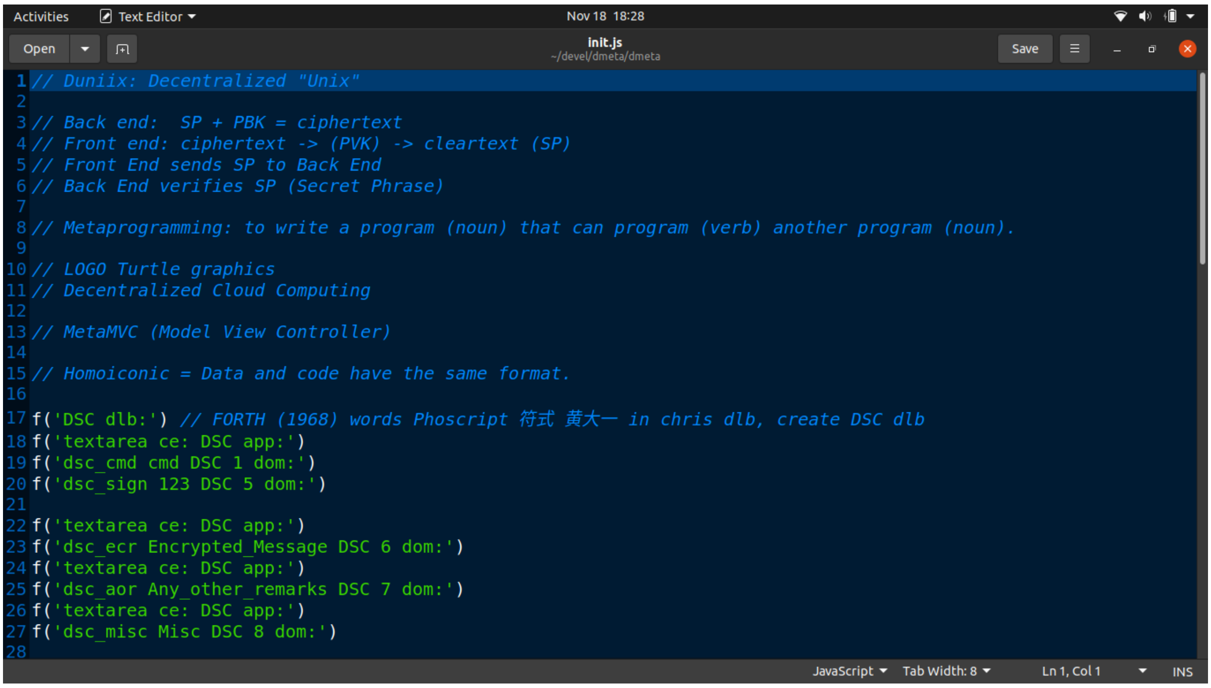
```
// chris browser 2021-2-20 2255
f(': do_sign awa: im_pbk ss: mk_sign ;')
f(': im_pbk . dup: js: jd: msg ix: awa: impbk: ss: ;')
f(': mk_sign awa: sign_kp: rsa_sign ;')
```

DMetaHash of Contract: Dw9mm08vpw==

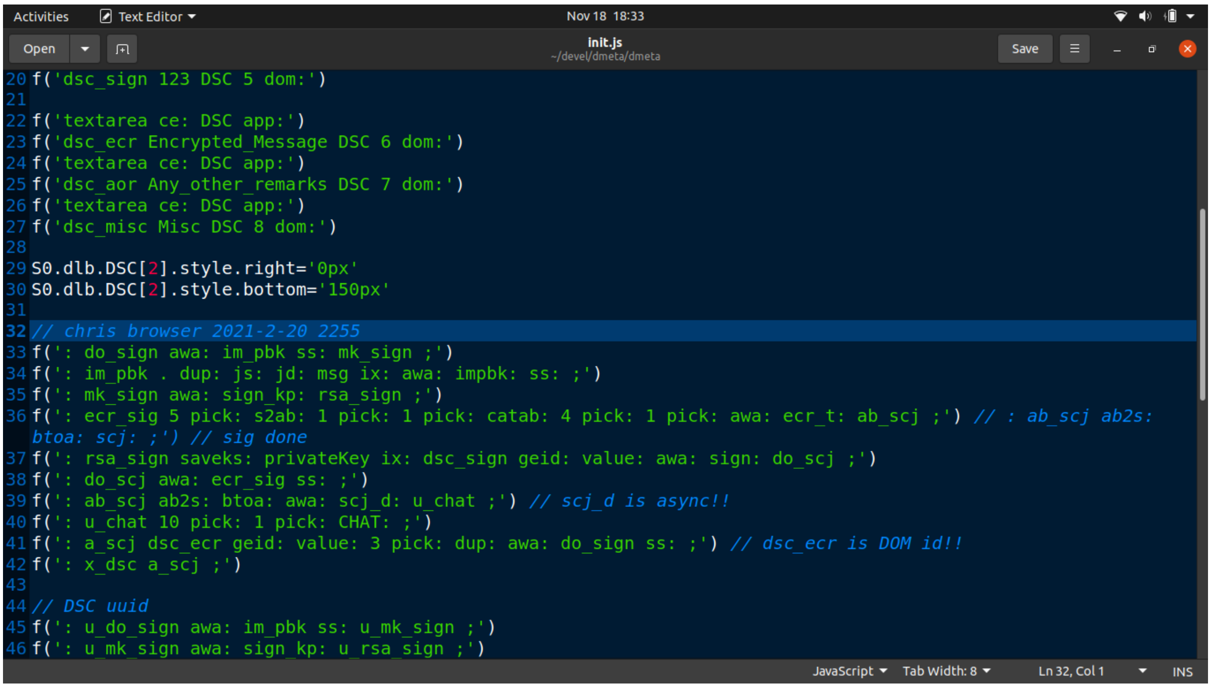
DMC3 Triplets: Dw9mm08vpw== BubzmttepQ== C2jDSUsf/A==

Root Hash (Hash of All Hashes): DUtI53pixA==

Decentralized Meta Contract (DMetaContract, DMC) is arguably the simplest smart contract ever



```
// Duniix: Decentralized "Unix"

// Back end:  SP + PBK = ciphertext
// Front end: ciphertext -> (PVK) -> cleartext (SP)
// Front End sends SP to Back End
// Back End verifies SP (Secret Phrase)

// Metaprogramming: to write a program (noun) that can program (verb) another program (noun).

// LOGO Turtle graphics
// Decentralized Cloud Computing

// MetaMVC (Model View Controller)

// Homoiconic = Data and code have the same format.

f('DSC dlb:') // FORTH (1968) words Phoscript 符式 黄大一 in chris dlb, create DSC dlb
f('textarea ce: DSC app:')
f('dsc_cmd cmd DSC 1 dom:')
f('dsc_sign 123 DSC 5 dom:')

f('textarea ce: DSC app:')
f('dsc_ecr Encrypted_Message DSC 6 dom:')
f('textarea ce: DSC app:')
f('dsc_aor Any_other_remarks DSC 7 dom:')
f('textarea ce: DSC app:')
f('dsc_misc Misc DSC 8 dom:')

```



```
f('dsc_sign 123 DSC 5 dom:')

f('textarea ce: DSC app:')
f('dsc_ecr Encrypted_Message DSC 6 dom:')
f('textarea ce: DSC app:')
f('dsc_aor Any_other_remarks DSC 7 dom:')
f('textarea ce: DSC app:')
f('dsc_misc Misc DSC 8 dom:')

S0.dlb.DSC[2].style.right='0px'
S0.dlb.DSC[2].style.bottom='150px'

// chris browser 2021-2-20 2255
f(': do_sign awa: im_pbk ss: mk_sign ;')
f(': im_pbk . dup: js: jd: msg ix: awa: impbk: ss: ;')
f(': mk_sign awa: sign_kp: rsa_sign ;')
f(': ecr_sig 5 pick: s2ab: 1 pick: 1 pick: catab: 4 pick: 1 pick: awa: ecr_t: ab_scj ;') // : ab_scj ab2s: btoa: scj: ;') // sig done
f(': rsa_sign saveks: privateKey ix: dsc_sign geid: value: awa: sign: do_scj ;')
f(': do_scj awa: ecr_sig ss: ;')
f(': ab_scj ab2s: btoa: awa: scj_d: u_chat ;') // scj_d is async!!
f(': u_chat 10 pick: 1 pick: CHAT: ;')
f(': a_scj dsc_ecr geid: value: 3 pick: dup: awa: do_sign ss: ;') // dsc_ecr is DOM id!!
f(': x_dsc a_scj ;')

// DSC uuid
f(': u_do_sign awa: im_pbk ss: u_mk_sign ;')
f(': u_mk_sign awa: sign_kp: u_rsa_sign ;')
```