



Multi Code

マルチコード

maXbox Starter 155 - Code in Multicode.

Source:

[Split a character string based on change of character - Rosetta Code](#)

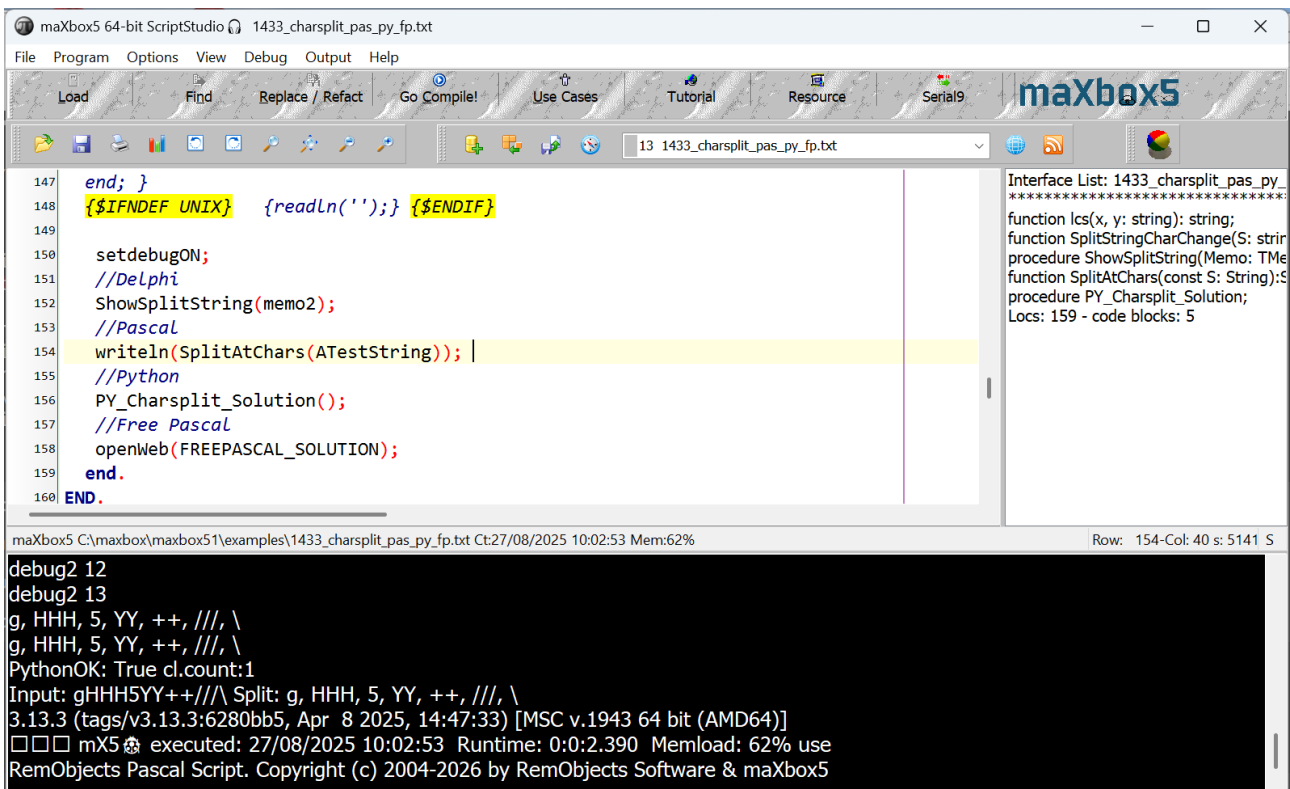
Script_: [Download 1433 charsplit_pas_py_fp_js_uc_test.txt \(maXbox5\)](#)

Multi-code programming refers to the practice of integrating multiple programming languages, paradigms, or models within a single system or application. This approach is particularly useful in scenarios where different languages or tools excel at specific tasks, allowing developers to leverage their strengths for optimal performance and flexibility.

Also Modern CLIs do more than run multi-commands. They:

- Save time with automation
- Integrate with cloud and AI tools
- Keep your focus in one window
- Work the same across Linux, macOS, and Windows

So CLI Tools still matter and I want to show that with one problem from rosettacode and the corresponding 5 solutions:



1433_mX5GuiScreenshot2025-08-27100346.png

The advantages of multi-code (or multi-core) are:

- **Optimized Performance:** Each language or tool is used where it performs best.
- **Flexibility:** Adapts to diverse project requirements.
- **Scalability:** Supports complex, large-scale systems.

We combine languages like Python, Pascal, Delphi, Free Pascal and Javascript with the following problem:

"Split a (character) string into comma (plus a blank) delimited strings based on a change of character (left to right)".

Blanks should be treated as any other character (except they are problematic to display clearly). The same applies to commas. For instance, the string:

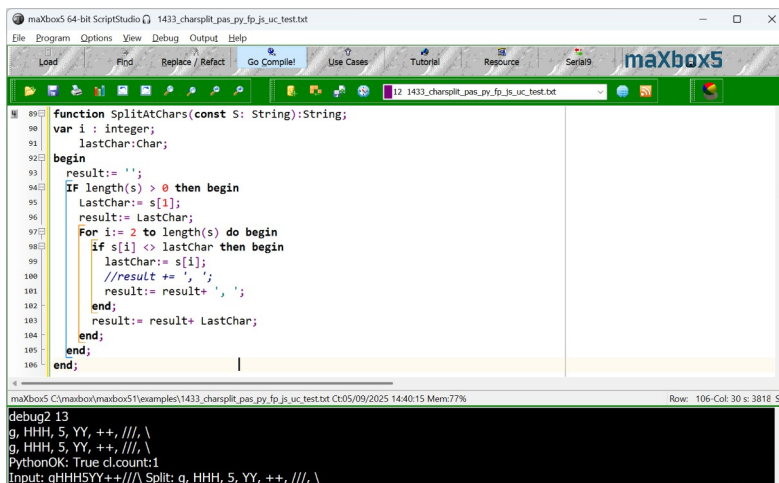
```
gHHH5YY+//\
```

should be split and show:

```
g, HHH, 5, YY, +, //, \
```

We start with the **Pascal** solution:

```
function SplitAtChars(const S: String):String;
var i : integer; lastChar:Char;
begin
  result:= '';
  IF length(s) > 0 then begin
    LastChar:= s[1];
    result:= LastChar;
    For i:= 2 to length(s) do begin
      if s[i] <> lastChar then begin
        lastChar:= s[i];
        //result += ', ';
        result:= result+ ', ';
      end;
      result:= result+ LastChar;
    end;
  end;
end;
```



The screenshot shows the ScriptStudio IDE interface. The main editor window displays the Pascal code for the `SplitAtChars` function. Below the editor, the output window shows the execution results:

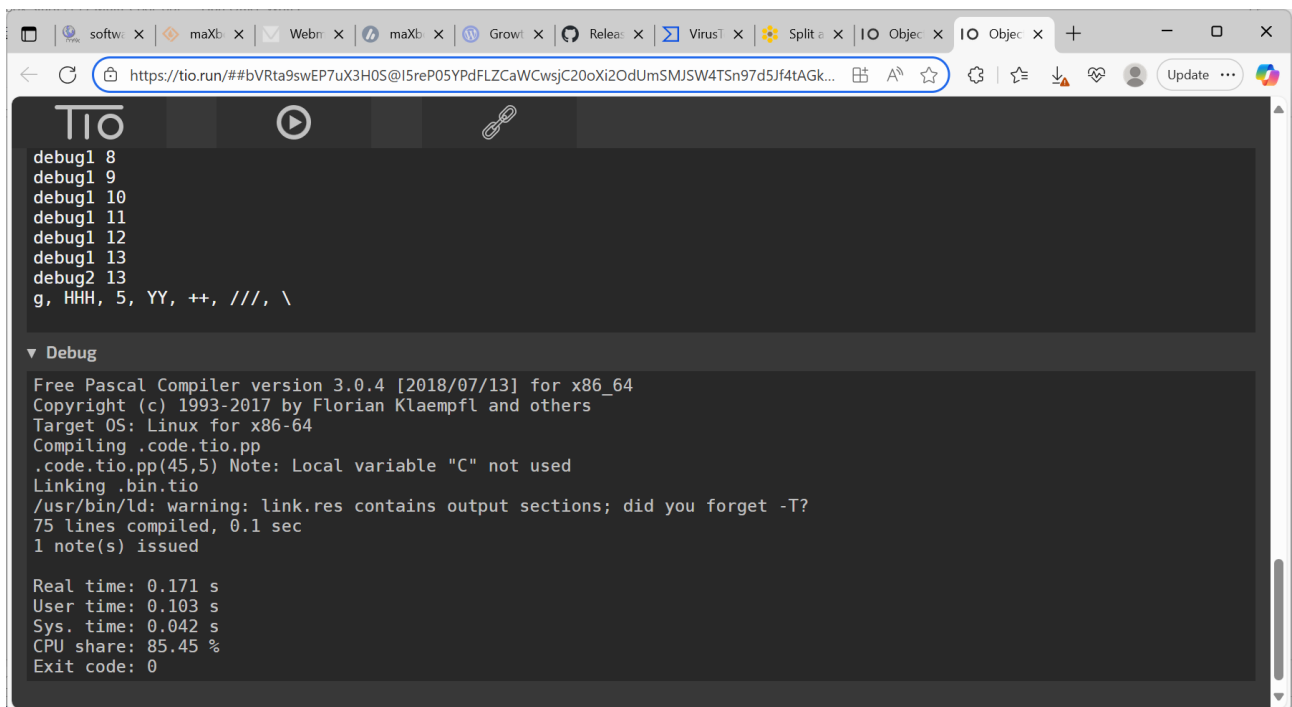
```
debug2 13
g, HHH, 5, YY, +, //, \
g, HHH, 5, YY, +, //, \
PythonOK: True cl.count:1
Input: gHHH5YY+//\ Split: g, HHH, 5, YY, +, //, \
```

The second, the **Delphi** one has a form to configure the output:

```
const S1 = 'gHHH5YY++///\';
procedure ShowSplitString(Memo: TMemo);
var S2: string;
begin
    Memo.Lines.Add(S1);
    S2:= SplitStringCharChange(S1);
    Memo.Lines.Add(S2);
end;
```

It works with: Delphi version 6.0 and needs the Library SysUtils and StdCtrls, in maXbox there are precompiled on board.

The third one is **Free Pascal** and runs on the internet with the help of TIO: The web server of and the arenas (where user code is executed) are currently run on three separate servers. **TIO** is getting more and more traffic, so additional arenas will be required. Also, server-side permalinks will eventually require a separate storage. With your help, I hope to ensure a smooth operation of all TIO services.



```
TIO
debug1 8
debug1 9
debug1 10
debug1 11
debug1 12
debug1 13
debug2 13
g, HHH, 5, YY, ++, ///, \

▼ Debug
Free Pascal Compiler version 3.0.4 [2018/07/13] for x86_64
Copyright (c) 1993-2017 by Florian Klaempfl and others
Target OS: Linux for x86-64
Compiling .code.tio.pp
.code.tio.pp(45,5) Note: Local variable "C" not used
Linking .bin.tio
/usr/bin/ld: warning: link.res contains output sections; did you forget -T?
75 lines compiled, 0.1 sec
1 note(s) issued

Real time: 0.171 s
User time: 0.103 s
Sys. time: 0.042 s
CPU share: 85.45 %
Exit code: 0
```

TIO is powered by Digital Ocean.

The next is Java Script and will also run in a browser but a local based one. With WebView2 in Microsoft Edge, developers can now embed web content directly within native Windows applications. WebView2 offers a simple way to display web-based content using the Chromium engine inside your apps or scripts.

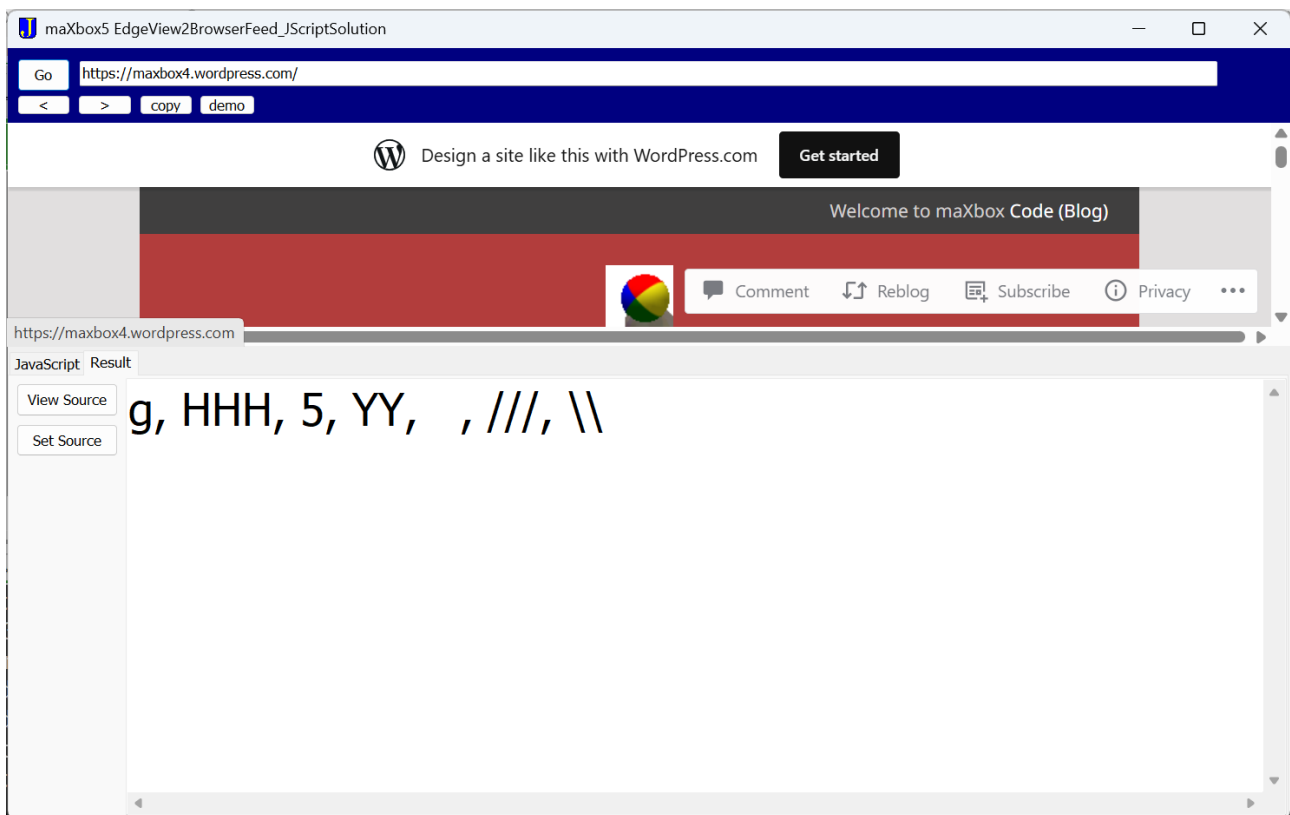
Specifically, WebView2 can run HTML, CSS, and JavaScript inside a standard non-web program, like something written in Visual C++, Delphi or .NET apps.

Support for Edge WebView2 on Windows 7, 8, Server 2012 and 2012R2, ended at Edge WebView2 version 107 in October of 2022 because those operating systems are no longer supported.

```

procedure JS_Solution;
begin
  with TEdgeViewForm.create(self) do begin
    width:= 1500;
    pagecontrol1.height:= 540;
    icon.loadfromresourcename(hinstance,'XJICON');
    Caption:= 'maXbox5 EdgeView2BrowserFeed_JSscriptSolution' ;
    panell1.ParentBackground:= false;
    panell1.color:= clnavy;
    sleep(500);
    navigate('https://maxbox4.wordpress.com/2025/07/01/ekon-29/');
    memoJavaScript.text:= JSCRIPT;//JS_InitFunc;
    PageControl1.ActivePageIndex := 1;
    memoHTML.font.size:= 26;
    memoJavaScript.wordwrap:= true;
    memoJavaScript.lines.linebreak:= lf;
    tabsheet3.caption:= 'Result';
    panel3.caption:= 'Source: Result';
    sleep(300)
    ExecuteScript(JSCRIPT);
    sleep(1600);
    writeln('result of js webview2: '+memoHTML.text);
    writeln('scriptresult of js webview2: '+getScriptresult);
    showmodal; //save resources
    free;
  end;
end;

```



pic: `with TEdgeViewForm.create(self) do begin`

The different sleep() is a bit tricky cause you need them to control the async return, otherwise you cant catch the result at the time you expect

it. The destination folder can contain the script, or in the script itself as a **const** or you can invoke scripts as an **URL** from a server with:

```
Navigate('https://maxbox4.wordpress.com/2025/07/01/ekon-29/jsdemo.js');
```

```
const JSCRIPT =
'(() => {                                     '+lf+
'    "use strict";                             '+lf+
'                                               '+lf+
'    // ----- SPLIT ON CHARACTER CHANGES ----- '+lf+
'    const main = () =>                         '+lf+
'        group("gHHH5YY+//\\")                 '+lf+
'        .map(x => x.join(""))                 '+lf+
'        .join(", ");                           '+lf+
```

The above is the basic usage of the WebView2 handling. You can customize the parameters and error handling based on specific needs. There's a tutorial about the topic: Tutorial 129 Using WebView2, October 2024.

The last solution is the Python one procedure:

```
Const PYFUNC =
' def splitter(text): '+lf+
'     return ", ".join("".join(group) for key, group in groupby(text)) ';
```

```
procedure PY_Charsplit_Solution;
begin
  with TPythonEngine.Create(Nil) do begin
    autofinalize:= false;
    loadDLL;
    try
      ExecString('import sys; from itertools import groupby');
      ExecString(PYFUNC);
      addclient(TEngineClient.create(nil));
      execstr('txt = 'gHHH5YY+//\\'
# Note backslash is Python escape char. ');
      println(evalstr('f'Input: {txt}\n Split: {splitter(txt)}'));
      println(evalstr('sys.version'));
    except
      raiseError;
    finally
      unloadDll;
      Free;
    end;
  end;
end;
```

Conclusion

Multi-code programming is a powerful approach for modern software development, especially in domains requiring high efficiency and adaptability.

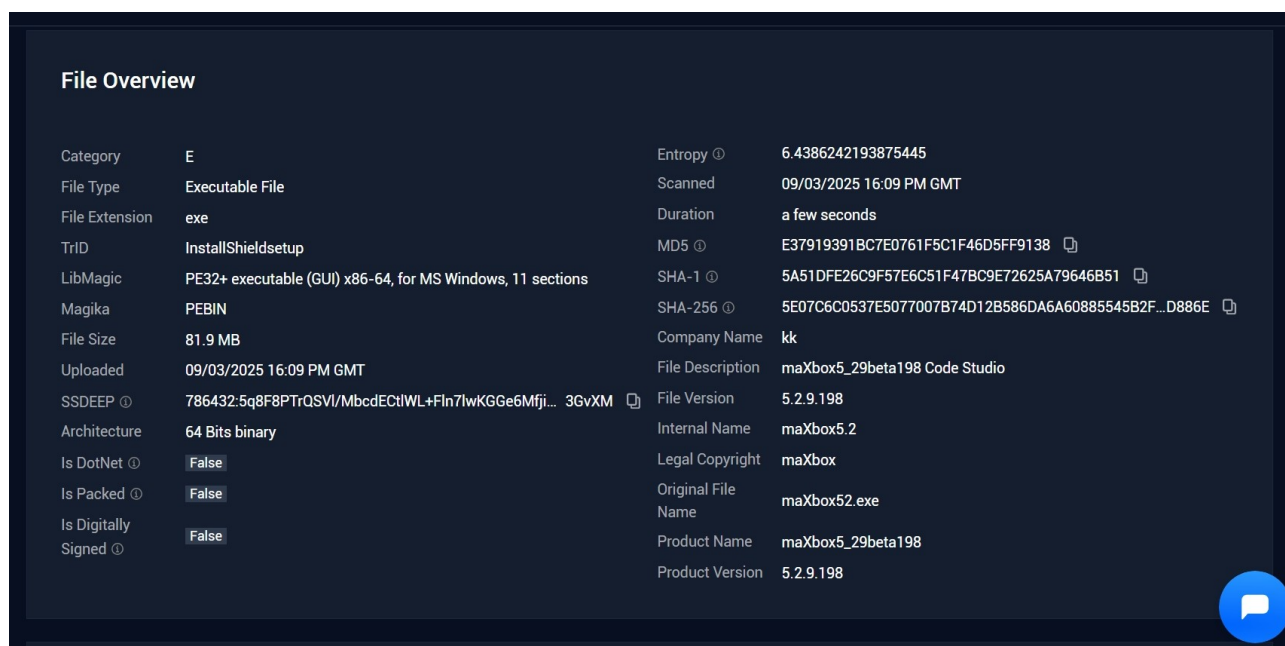
Put simply Microsoft EdgeWebView2 is a stripped down version of the Microsoft Edge browser- based on Google Chrome that developers can use to

run web code inside of applications you are using. By leveraging the web platform inside desktop scripts, developers can create rich user experiences, improve performance, and utilize modern web capabilities.

The challenges are:

- Complexity: Managing multiple languages can increase development and debugging effort.
- Interoperability Issues: Ensuring smooth communication between components.
- Learning Curve: Requires expertise in multiple programming paradigms.

Facilitates communication between different languages or frameworks, often through APIs, shared memory, or intermediate representations.



References:

[How to Use WebView2 in Microsoft Edge Browser - GeeksforGeeks Try It Online](#)

[maxkleiner/P4D-Data-Sciences: A collection of lightweight Python wrappers based on Python4Delphi simplifying Data Sciences development with Delphi](#)

[How to use SHFileOperation - Blog - Silicon Cloud](#)

Doc and Tool: [maXbox5 - Manage Files at SourceForge.net](#)

[Release maXbox V5.2.9 · maxkleiner/maXbox5](#)

Max Kleiner 05/09/2025