

Beyond JavaScript

Adding ooRexx and other JSR-223 Scripting Languages to the JavaFX WebView Control.

Maximilian Wannemacher

Why?

“ “ The by-design purpose of JavaScript was to make the monkey dance when you moused over it. Scripts were often a single line. We considered ten line scripts to be pretty normal, hundred line scripts to be huge, and thousand line scripts were unheard of.

- Eric Lippert

Why?



Concentrated power/responsibility



Different languages for different tasks



Personal Preference



Libraries

How?

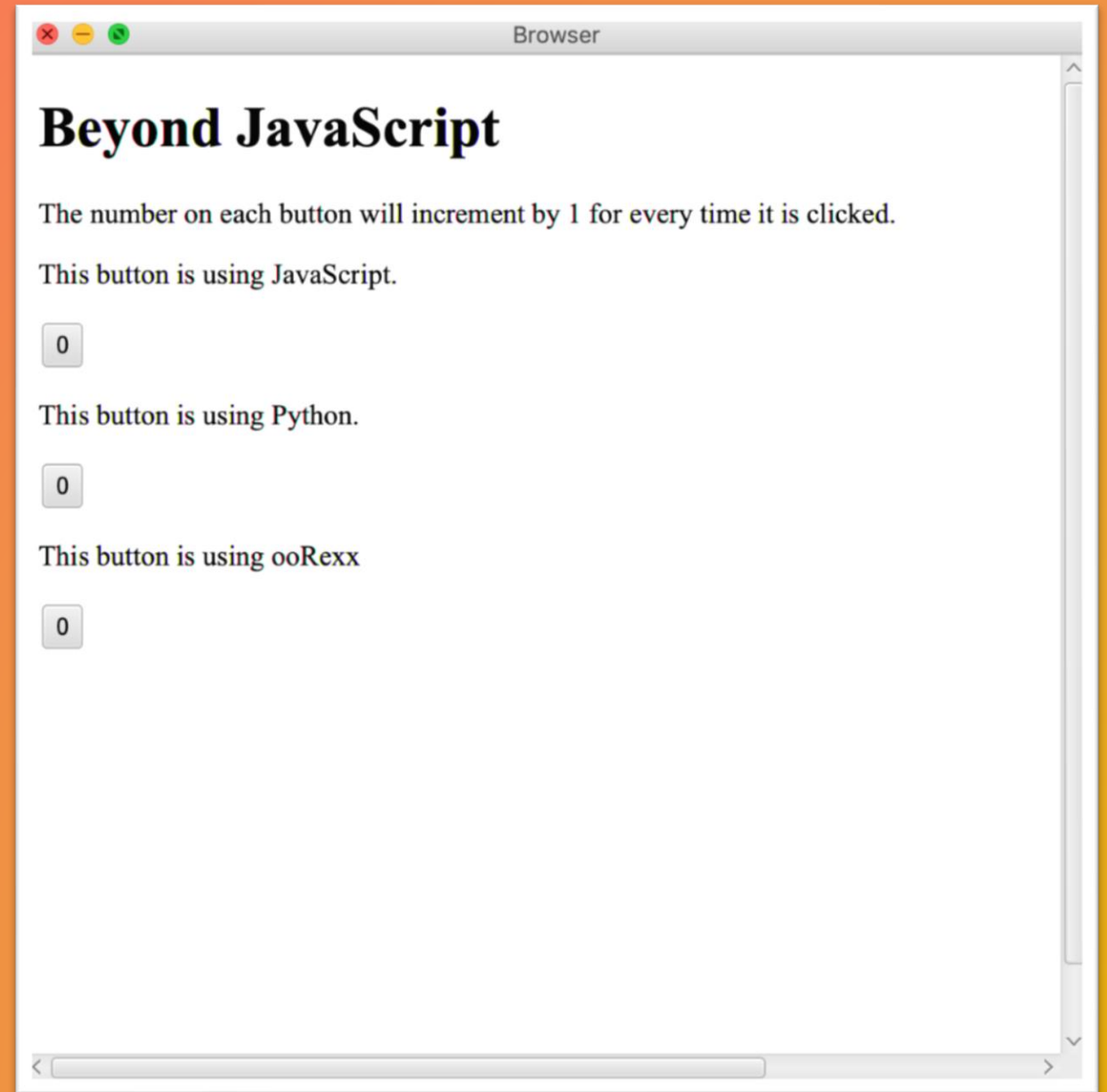
JavaFX Webview

Java Scripting
Framework

ScriptEngines

Result

A minimal Java-based browser application capable of displaying websites and evaluating scripts of multiple languages.



Things that work

- ✓ Displaying .html pages
- ✓ Handling external files
- ✓ Accessing DOM
- ✓ Evaluating scripts
- ✓ Invoking functions
- ~ Handling DOM events

```
<script type="text/javascript">  
  print("Hello world from JavaScript!");  
  var clickCount = 0;  
  function testFunctionJS(arg){  
    clickCount += 1;  
    arg.textContent = clickCount;  
  }  
</script>
```

testfile.html

```
<script type="text/python">
print("Hello world from Python!")
clickCount = 0
def testFunctionPy(arg):
    global clickCount
    clickCount += 1
    arg.textContent = str(clickCount)
</script>
```

testfile.html


```
<script type="text/ooress">  
say "Hello world from ooRexx!"  
.local~clickCount = 0  
exit  
::ROUTINE testFunctionRexx public  
  button = arg(1)  
  .local~clickCount += 1  
  button~textContent = .local~clickCount  
  return  
</script>
```

testfile.html

Things that don't work (yet)

- Mixing and matching languages
- DOM events other than onclick
- Some DOM event oddities
- Browser replacement

Conclusion

Java can be used to create a platform for language independent web scripting.

While the current implementation is not perfect, it can serve as a stepping stone for future improvement.