Testing javascript in the browser without the browser

There are times that we want to test snippets of Javascript code in different browsers to make sure that it works as intended in all our target browsers. I've always fired the browsers and pasted the code into DevTools or Web Inspector to check if the results are the same.

Now there is a pair of applications that will automate this for you: jsvu and eshost.

In order to install these packages run the following command

```
npm i -g jsvu eshost
```

JSVU (JavaScript Version Updater) manages the installation and update for differnt Javascript engines avoiding the compilation process. Out of the box, it supports the following engines and OS combinations.

Vendor	JavaScript engine	Binary name	mac64	win32	win64	linux32	linux
Microsoft	Chakra	chakra or ch	✅	✅	✅	❌	✅
Facebook	Hermes	hermes & hermes-repl	✅	❌	✅	❌	✅
WebKit/ Apple	JavaScriptCore	javascriptcore or jsc	✅	✅*	✅*	✅	✅
Fabrice Bellard	QuickJS	quickjs	❌	❌	❌	❌	✅
Mozilla	SpiderMonkey	spidermonkey or sm	✅	✅	✅	✅	✅
Google	V8	v8	✅	✅	✅	✅	✅

Vendor	JavaScript engine	Binary name	mac64	win32	win64	linux32	linux
Google	V8 debug	v8-debug	✅	✅	✅	✅	✅
Moddable-OpenSource	XS	xs	✅ (32)	✅	✅ (32)	✅	✅

When you first run it it will prompt you to select the JS engines to install. After initial install running the `jsvu` commannd will update the engines as appropriate.

Once we have the engines that we want to work with we can configure `ESHost` to run the same command with these multiple js engines. In this example we're adding the major browser's JS engines to work with ESHost.

```
# Chakra is old Edge's JS engine
eshost --add 'Chakra' ch ~/.jsvu/chakra
'Chakra'# JSCore is Safari's JS engine--add 'JavaScriptCore' jsc ~/.jsvu/j
# Spider'JavaScriptCore'# Spider Monkey is Firefox JS enginekey' jsshell ~
# V8 is Chrome's JS engi' jsshell ~/.jsvu/spidermonkey
# V8 is Chrome'# V8 is Chrome's JS engine
eshost --add 'V8' d8 ~/.jsvu/v8
```

Once we have the engines set up, ESHost is ready to go.

We have multiple ways to run scripts in ESHost. We can run a short script in all configured browsers with the `-e` flag like this:

```
eshost -e "12*12"
```

You can also run complete scripts in the configured browsers with the following command:

```
eshost my-script.js
```

The only flags I will refer to are `-m` and `-s`.

The `-m` flag will treat the script as a module with the corresponding differences

in syntax.

The `-s` flag will consolidate results when different engines produce the same result. For example, the following command:

```
eshost -se "console.log(112*12)"
```

will produce the following result where only JavaScriptCore produces a different result. This will help in researching browser differences that need workarounds

```
eshost -se "console.log(122*12)"
#### Chakra, SpiderMonkey, V8
1464
undefined

#### JavaScriptCore

TypeError: undefined is not an object (evaluating 'console.log')
'console.log'
```

For more information use the following command:

```
eshost --help
```