JavaScript
fwdays

Everywhere is undefined: using the “script” part of JS

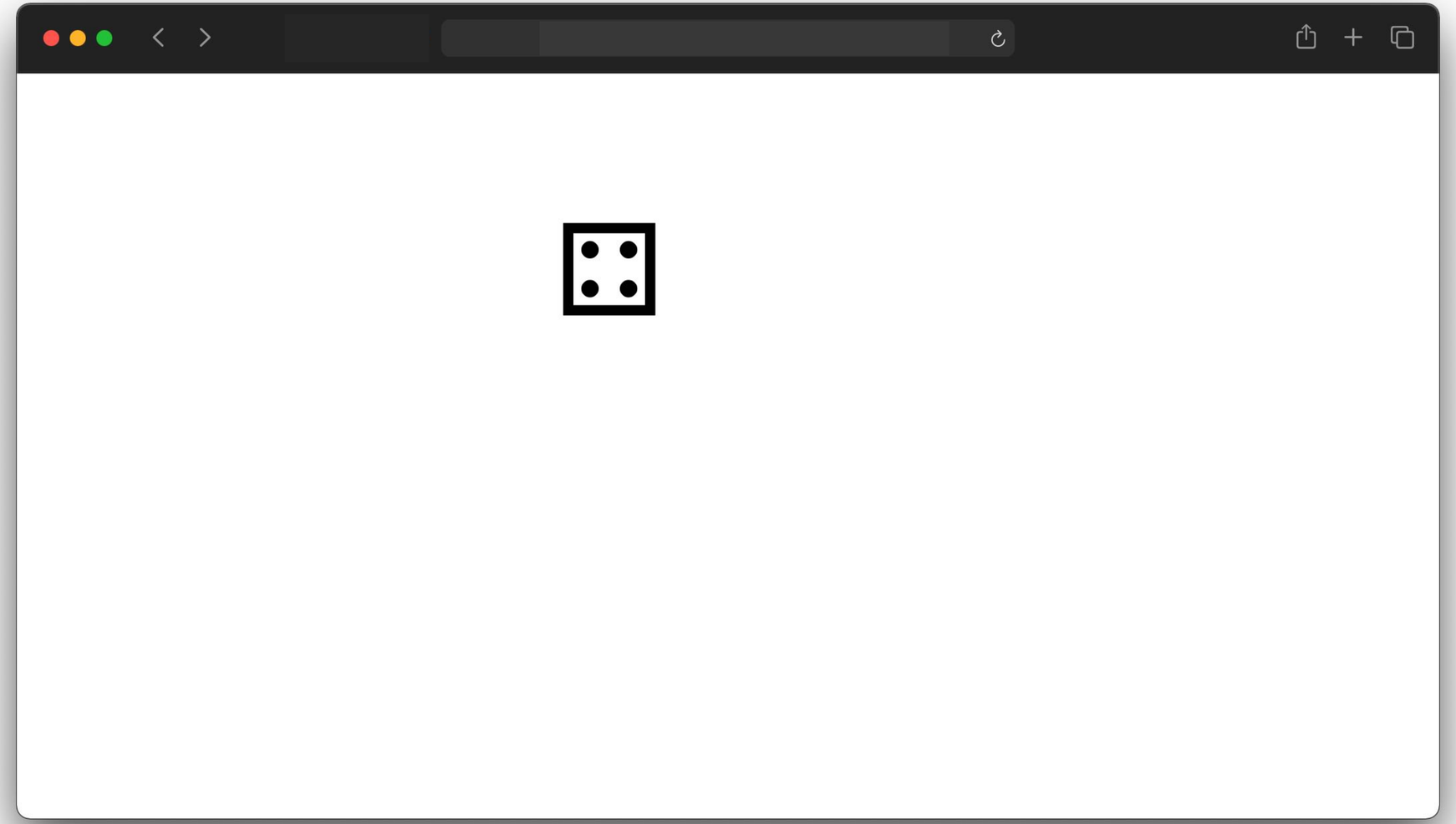
ANTON VERINOV

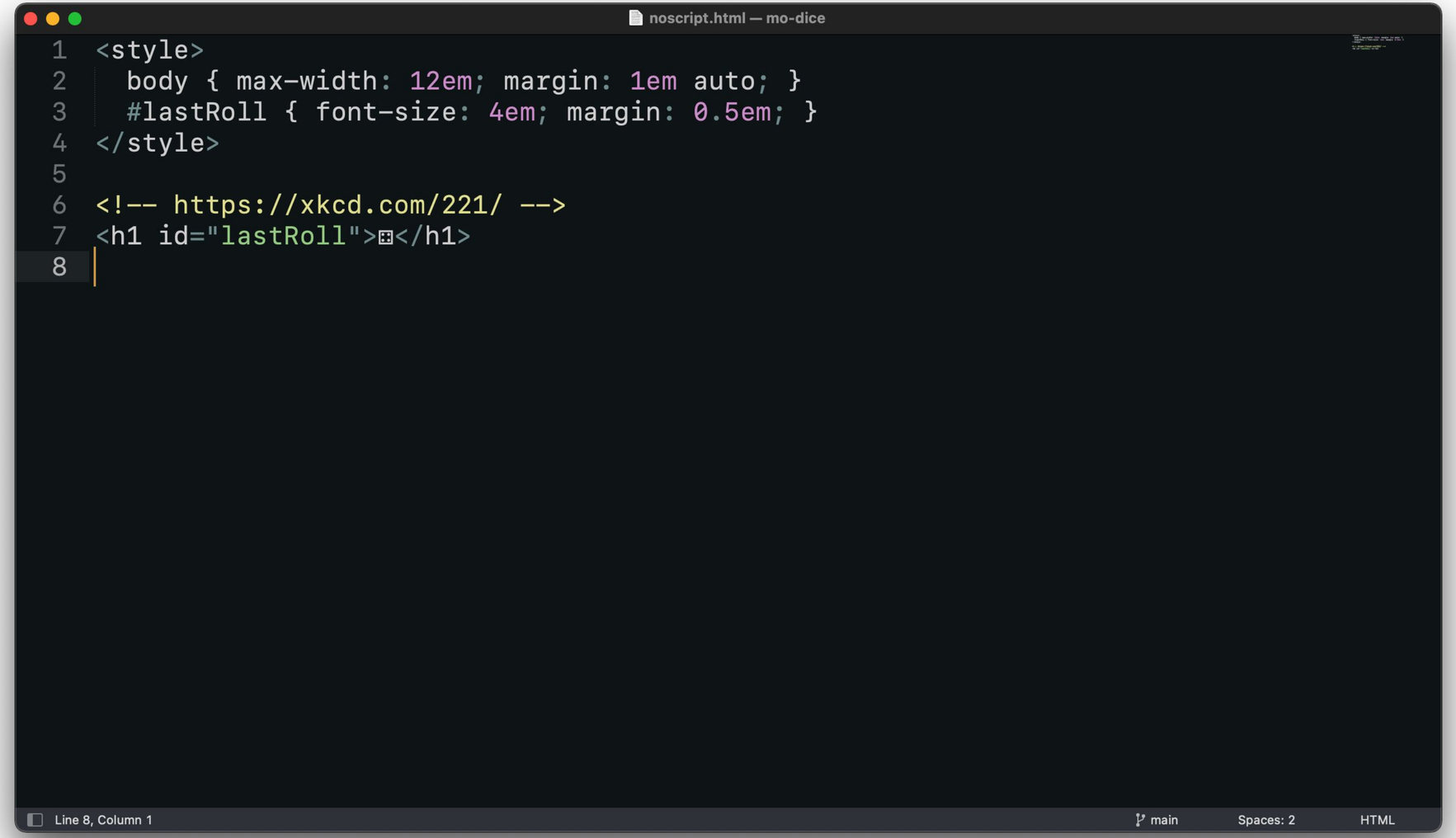
Wix Engineering

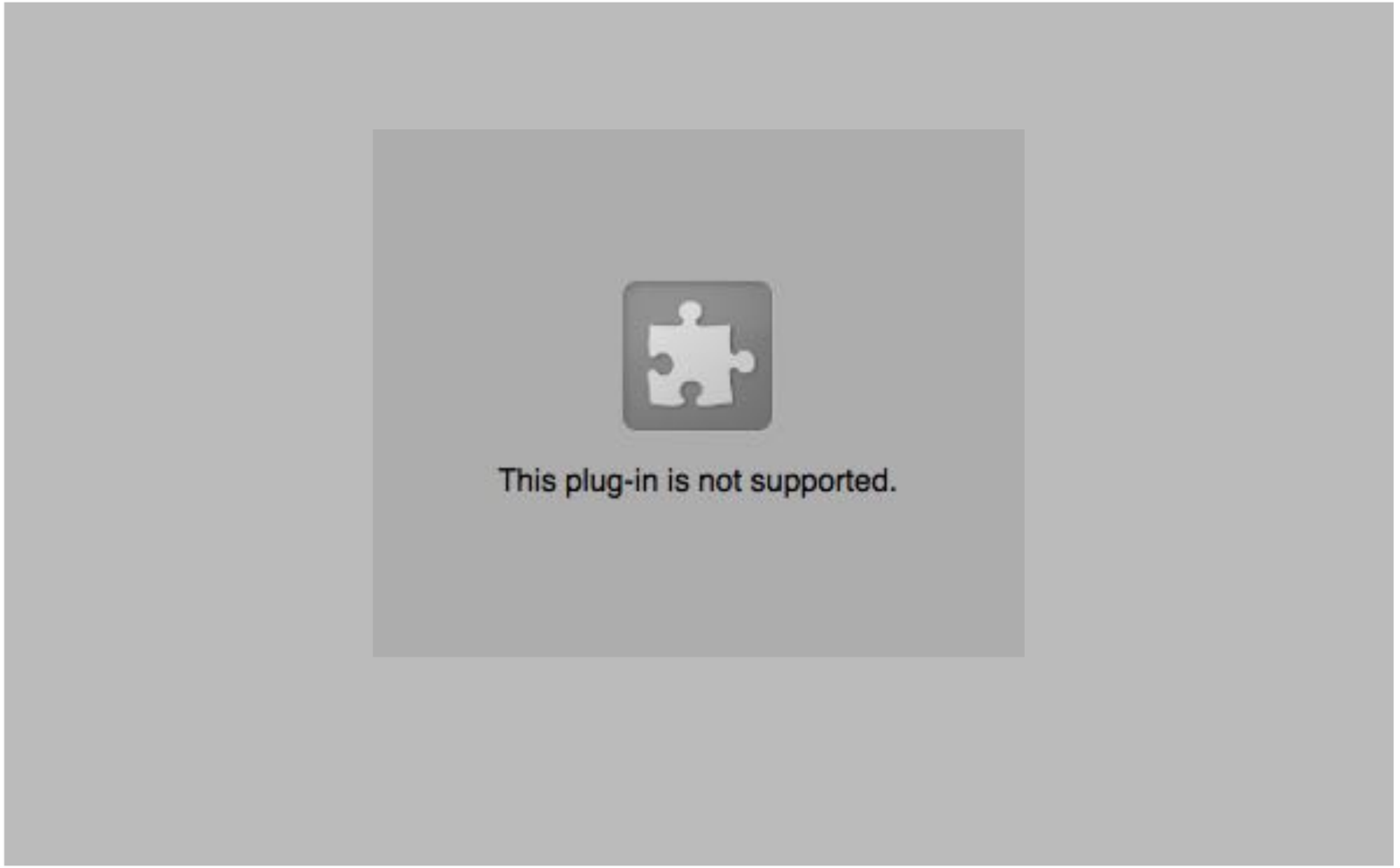
frontend →

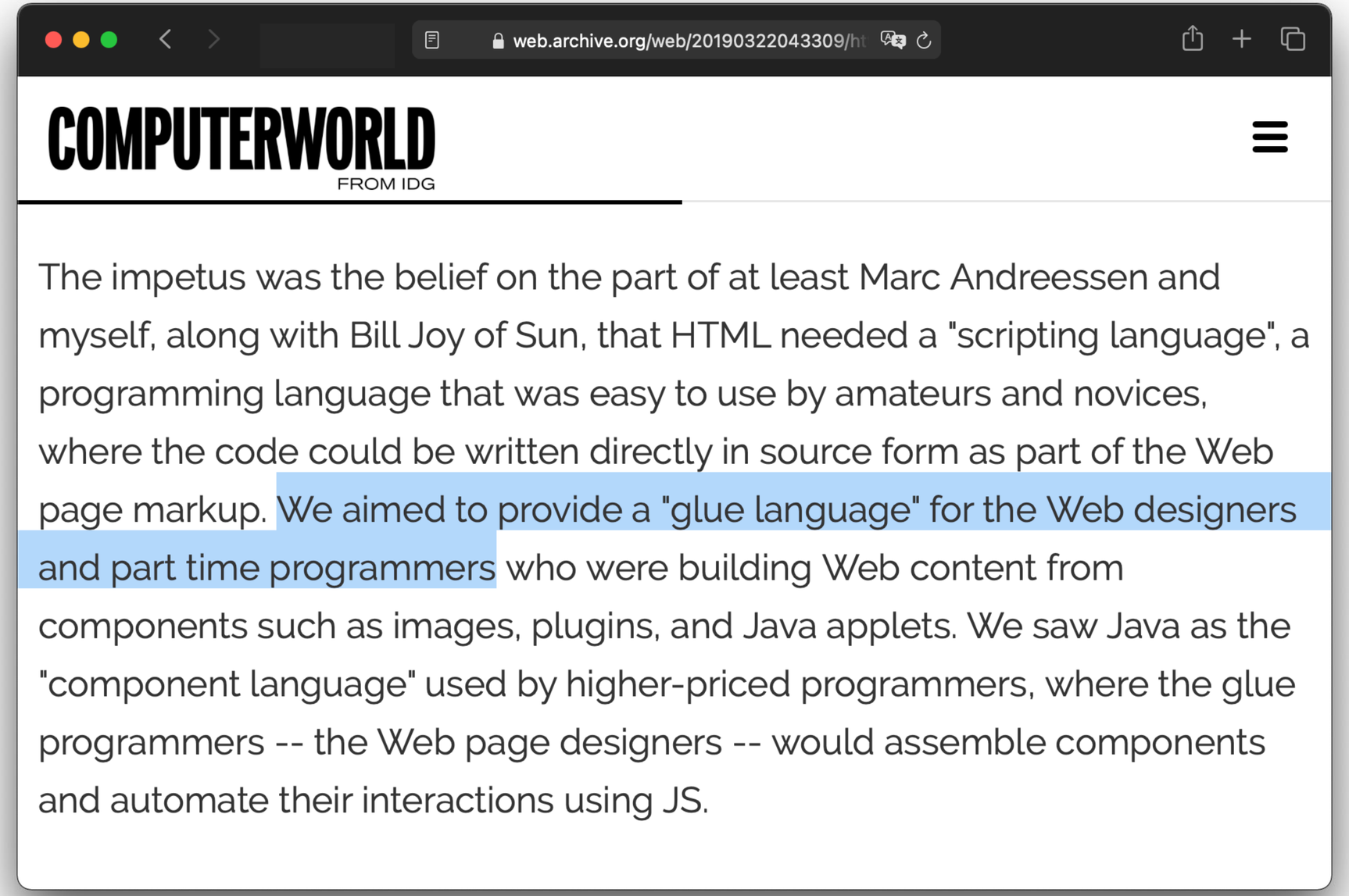
← @zemlanin

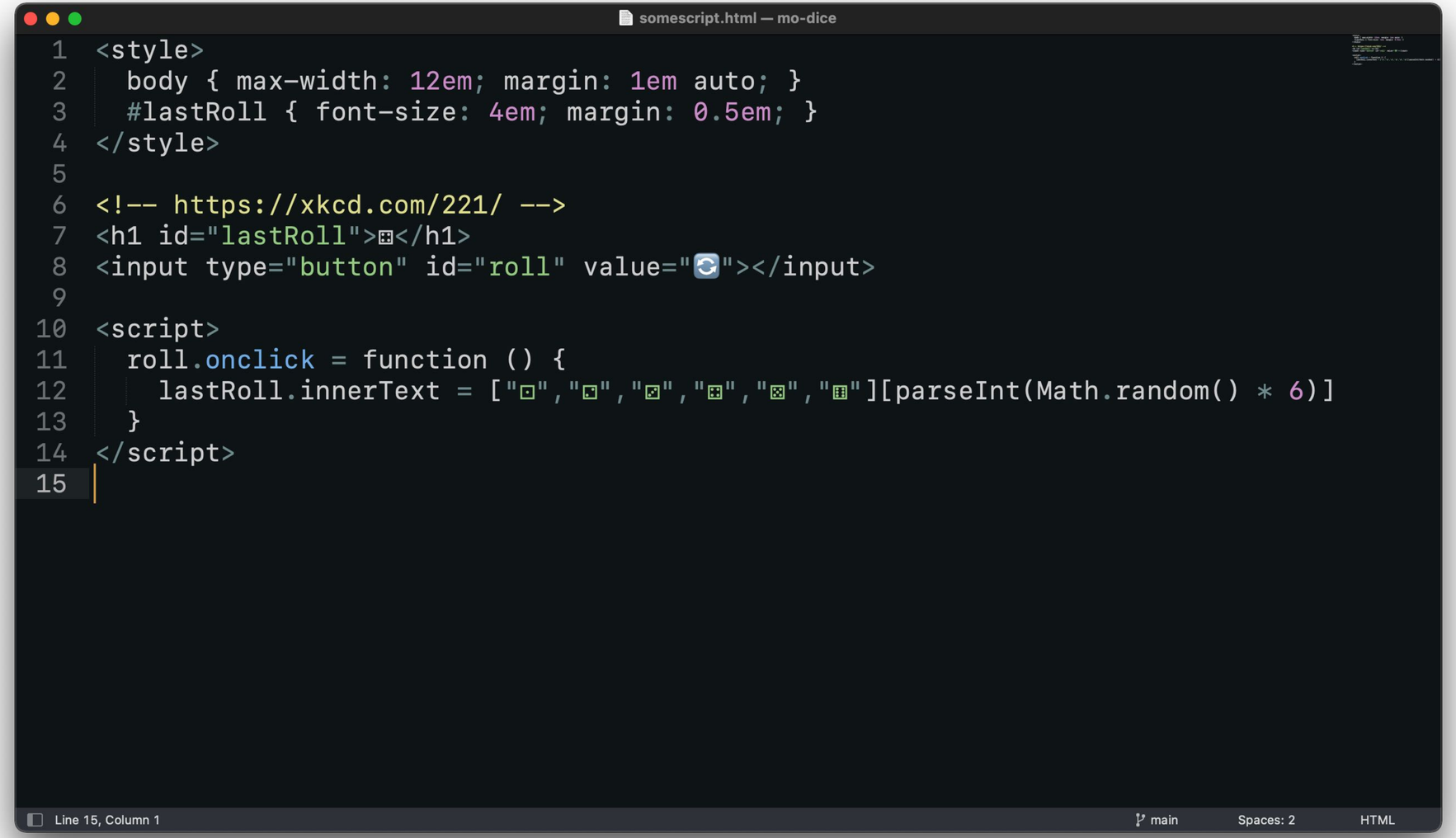
1995

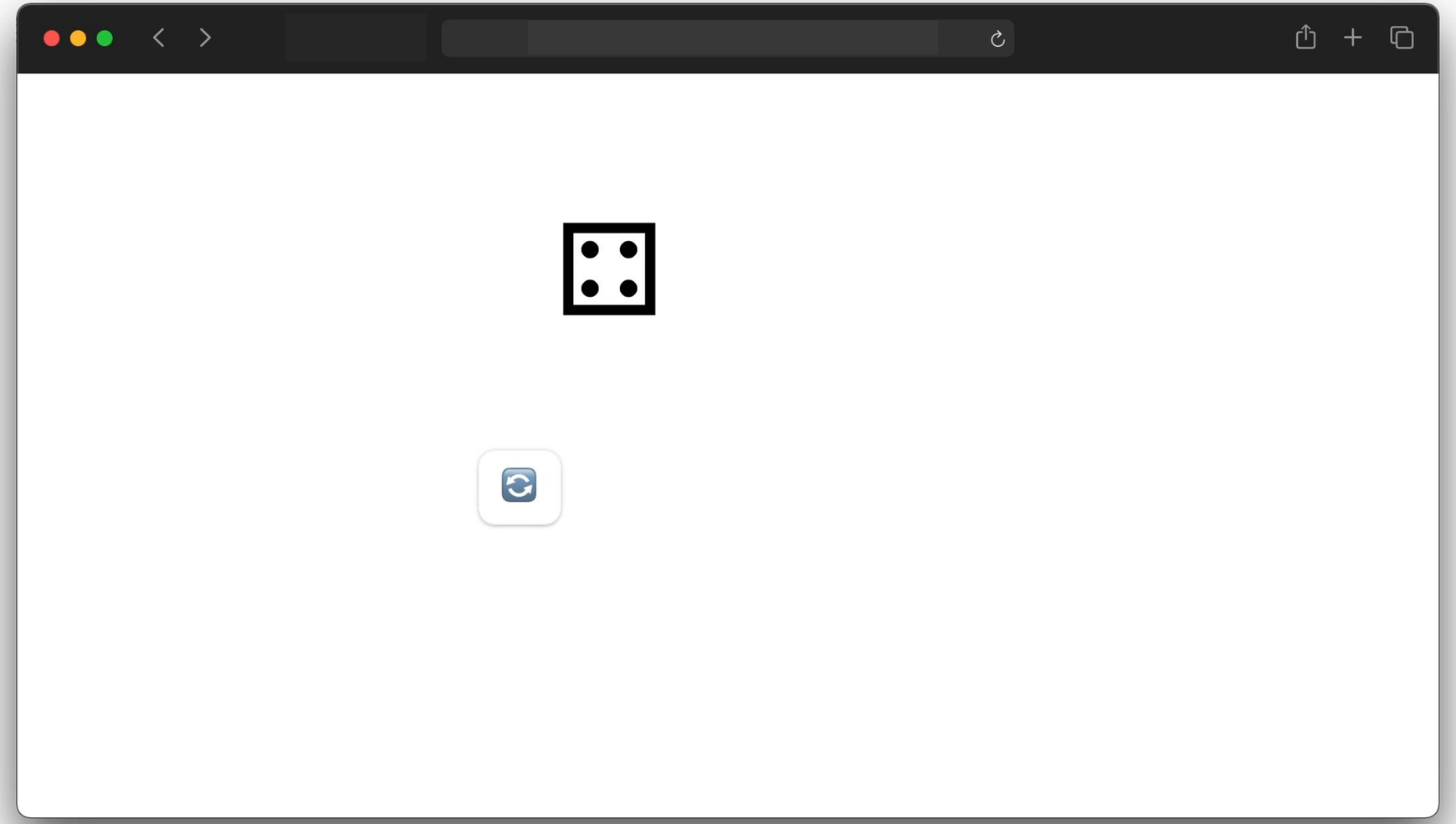



noscript.html — mo-dice
1 <style>
2 body { max-width: 12em; margin: 1em auto; }
3 #lastRoll { font-size: 4em; margin: 0.5em; }
4 </style>
5
6 <!-- https://xkcd.com/221/ -->
7 <h1 id="lastRoll">⚄</h1>
8
Line 8, Column 1
main
Spaces: 2
HTML

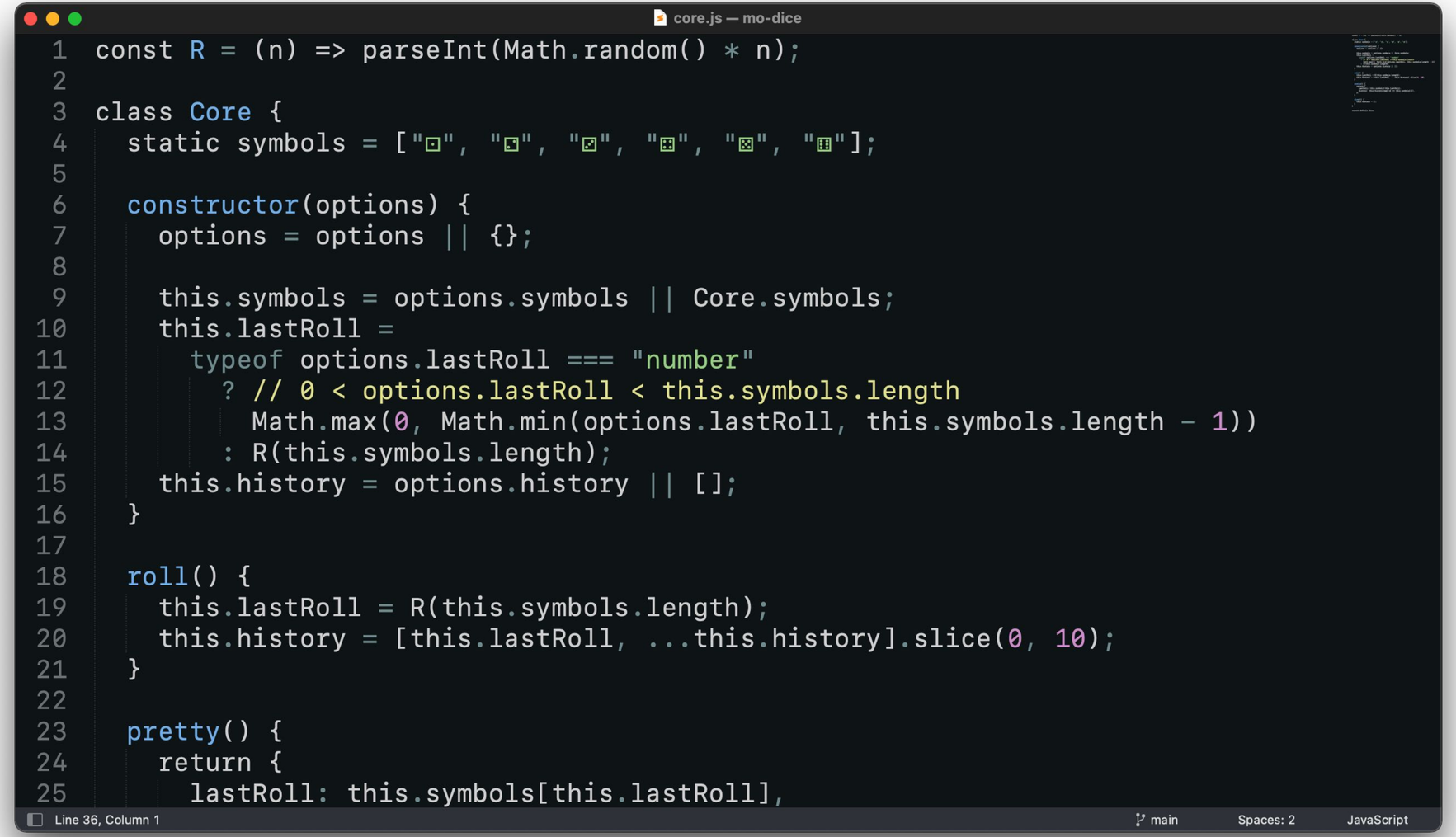

This plug-in is not supported.

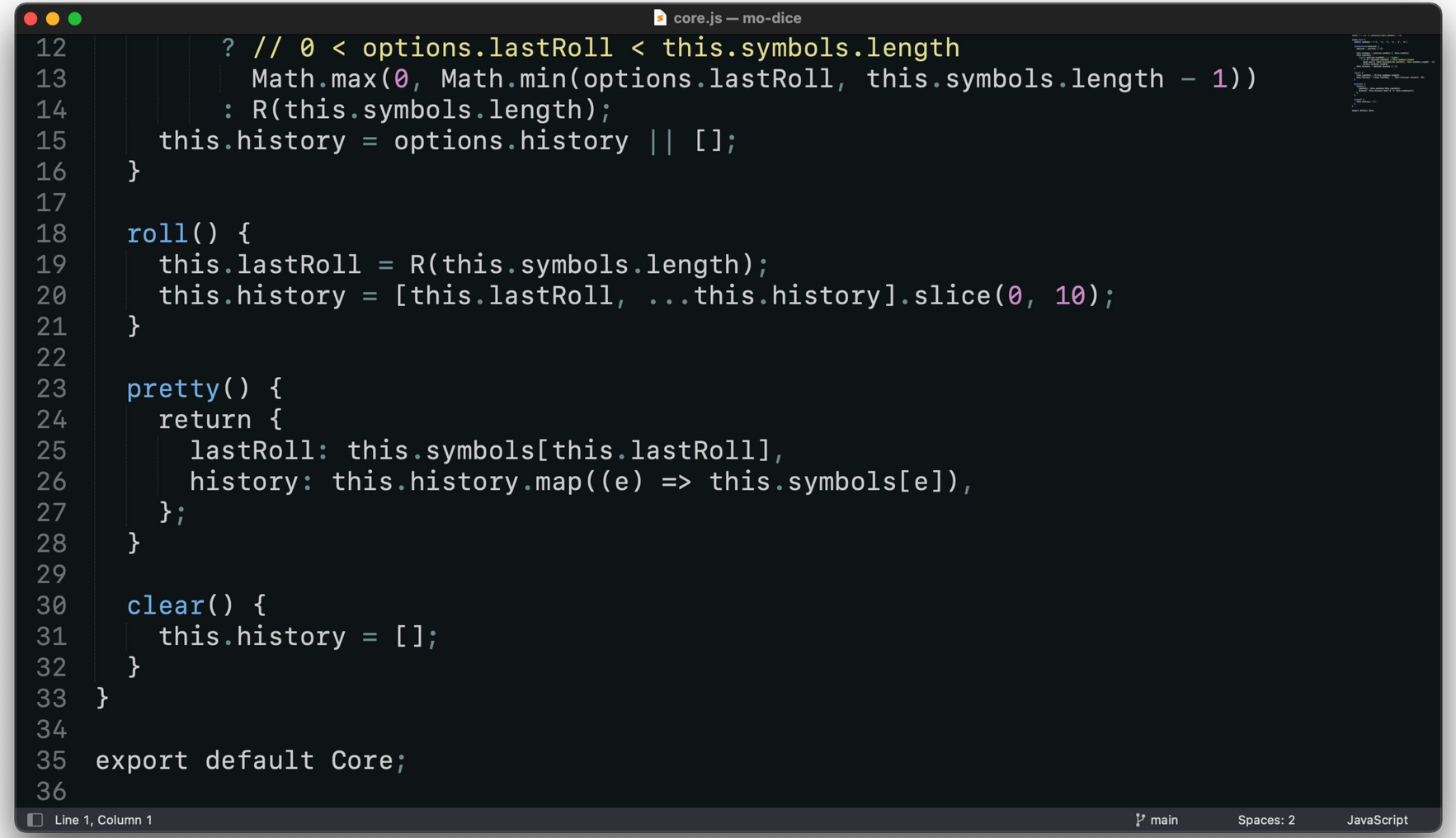

web.archive.org/web/20190322043309/ht
COMPUTERWORLD
FROM IDG
The impetus was the belief on the part of at least Marc Andreessen and myself, along with Bill Joy of Sun, that HTML needed a "scripting language", a programming language that was easy to use by amateurs and novices, where the code could be written directly in source form as part of the Web page markup. We aimed to provide a "glue language" for the Web designers and part time programmers who were building Web content from components such as images, plugins, and Java applets. We saw Java as the "component language" used by higher-priced programmers, where the glue programmers -- the Web page designers -- would assemble components and automate their interactions using JS.

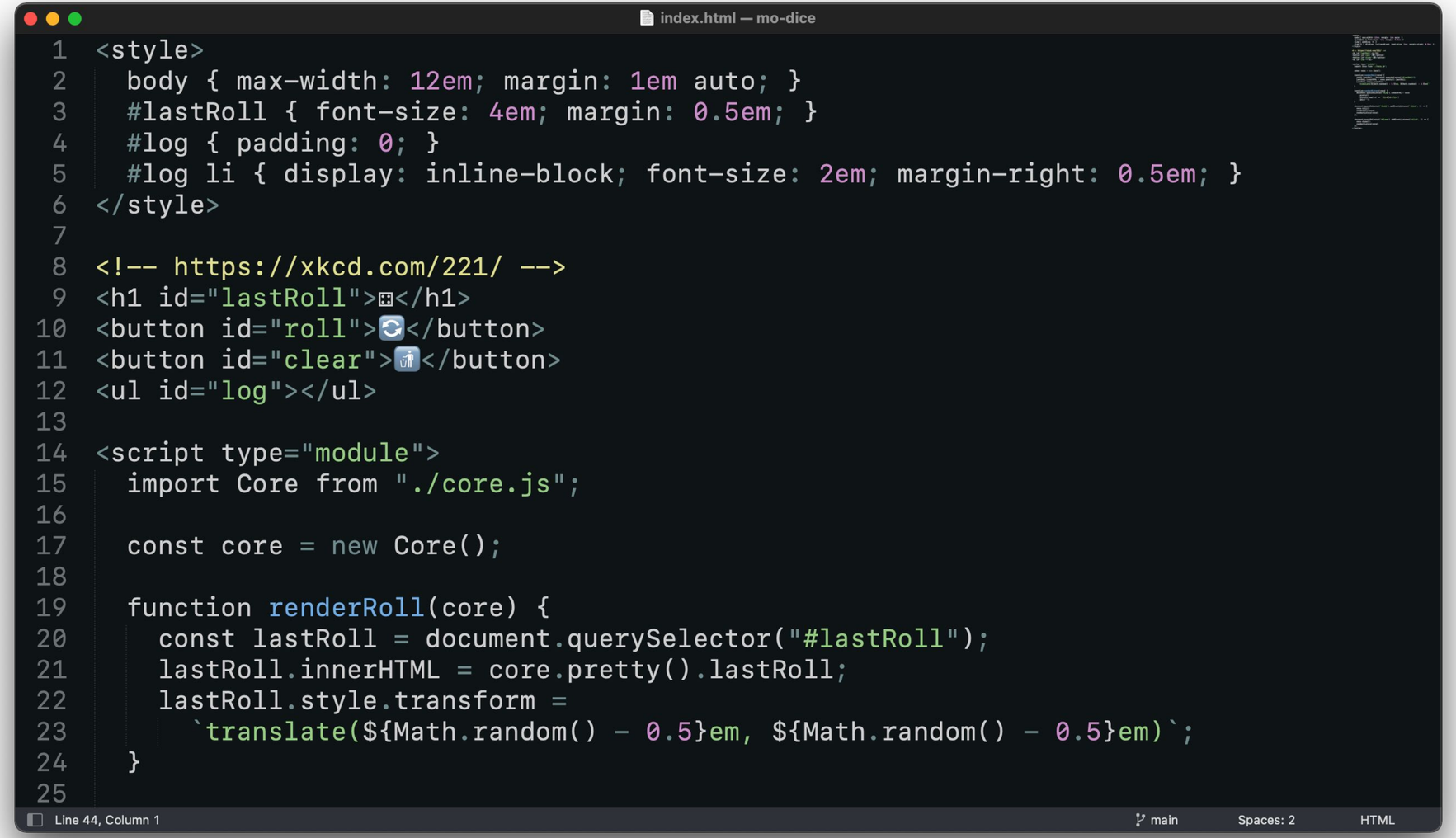

somescript.html — mo-dice
1 <style>
2 body { max-width: 12em; margin: 1em auto; }
3 #lastRoll { font-size: 4em; margin: 0.5em; }
4 </style>
5
6 <!-- https://xkcd.com/221/ -->
7 <h1 id="lastRoll">⚅</h1>
8 <input type="button" id="roll" value="🔄"></input>
9
10 <script>
11 roll.onclick = function () {
12 lastRoll.innerText = ["⚀","⚁","⚂","⚃","⚄","⚅"][parseInt(Math.random() * 6)]
13 }
14 </script>
15
Line 15, Column 1
main
Spaces: 2
HTML

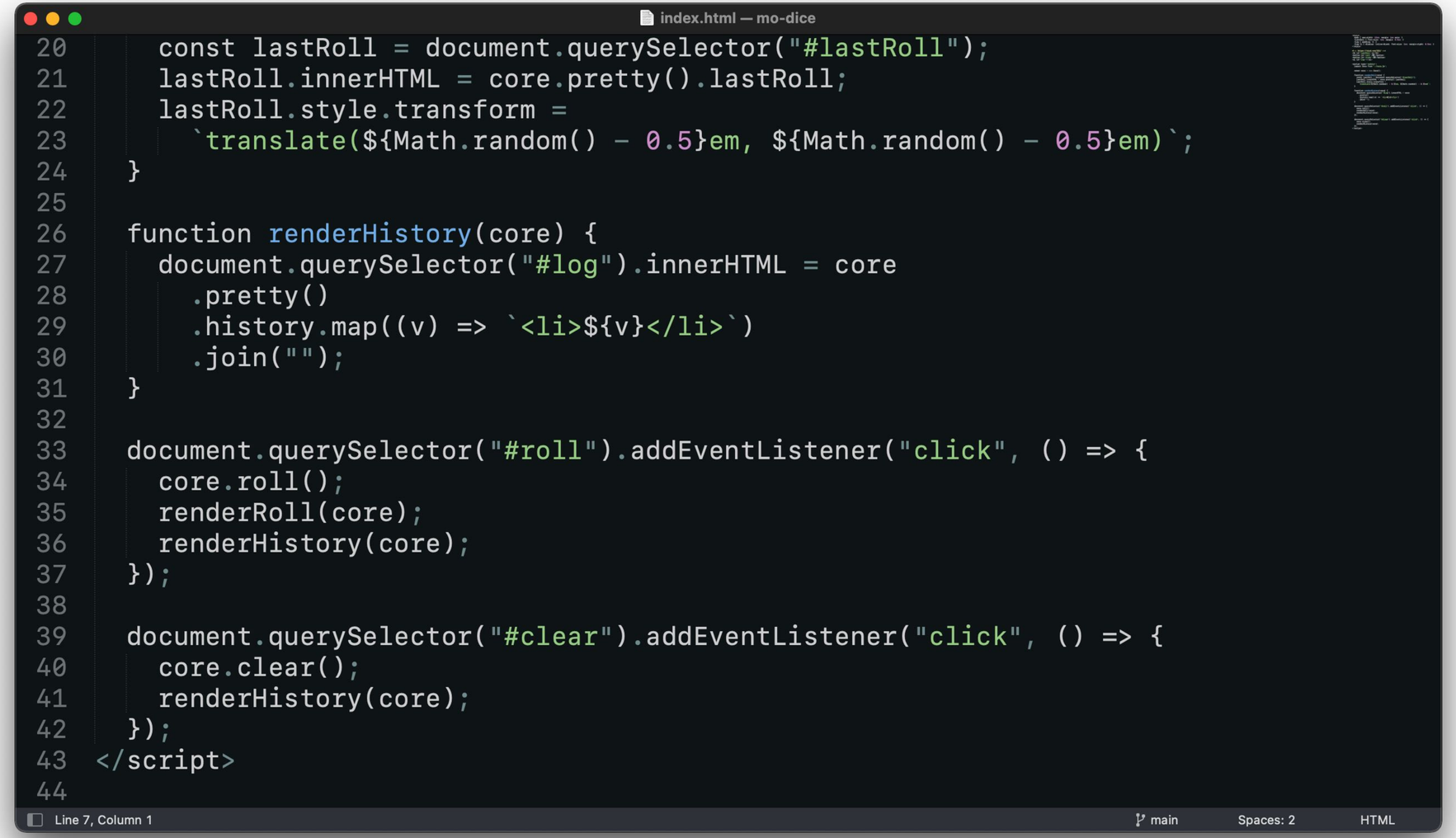


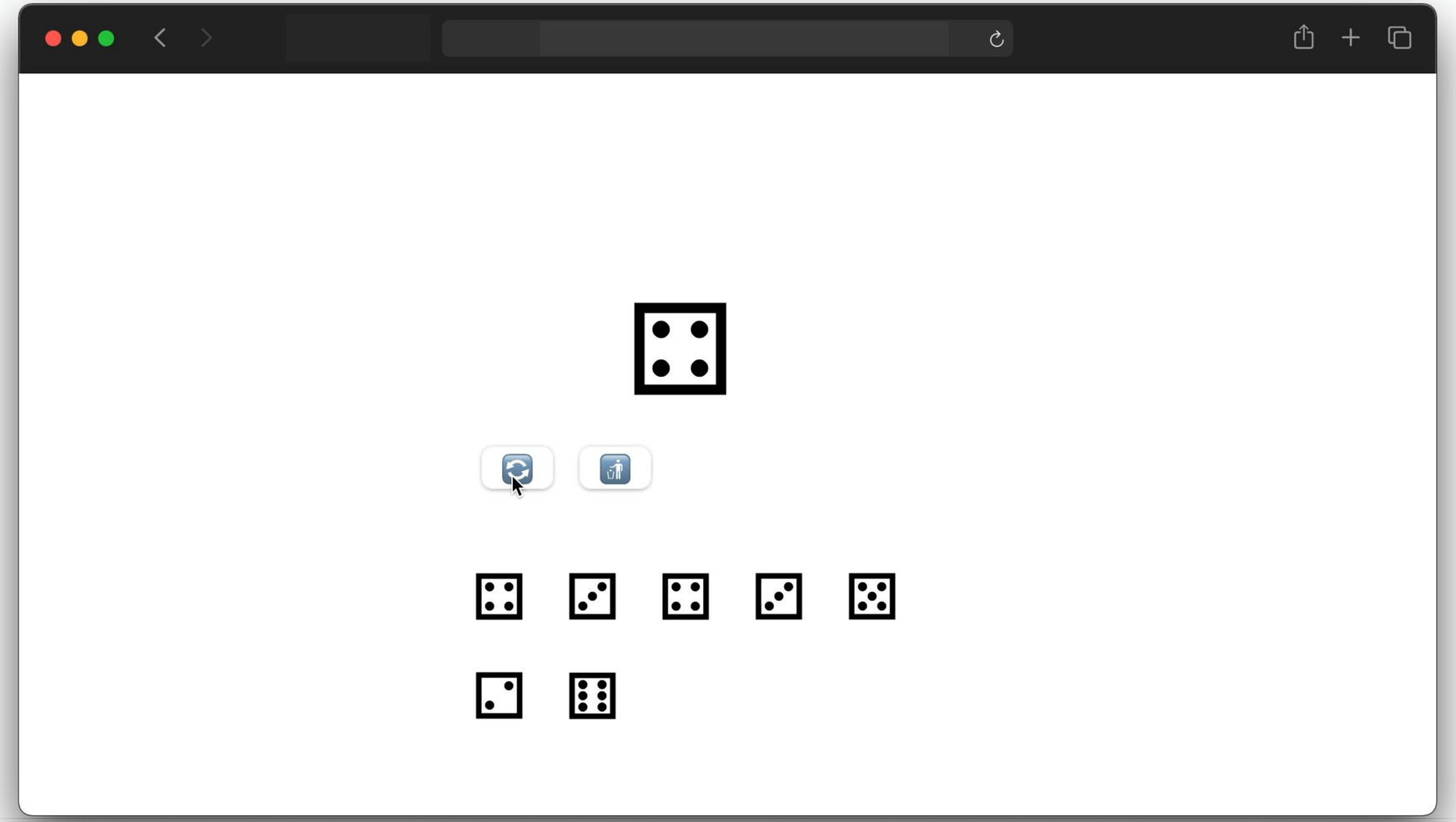
20XX

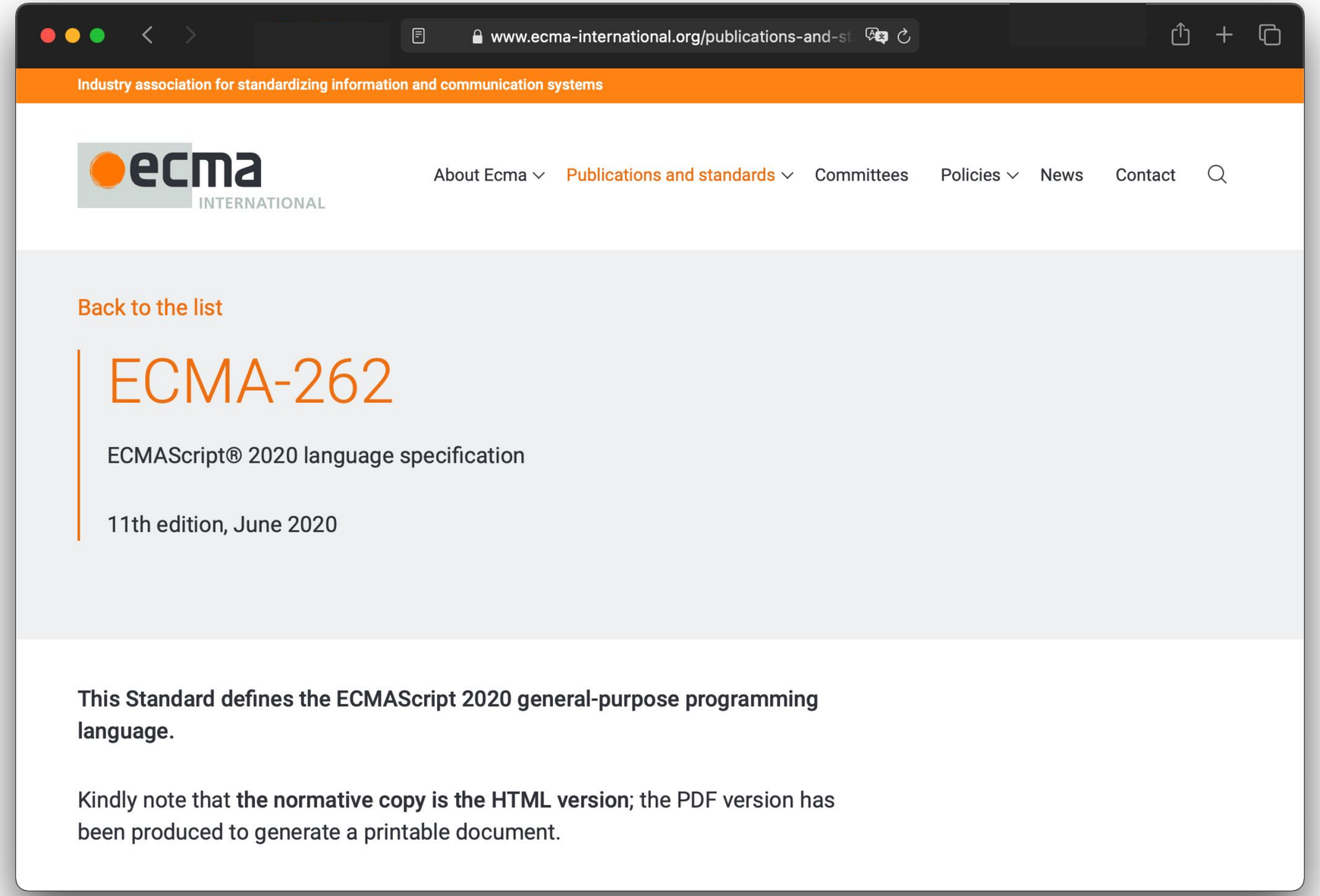

core.js — mo-dice
const R = (n) => parseInt(Math.random() * n);
class Core {
static symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
constructor(options) {
options = options || {};
this.symbols = options.symbols || Core.symbols;
this.lastRoll =
typeof options.lastRoll === "number"
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
Line 36, Column 1
main
Spaces: 2
JavaScript

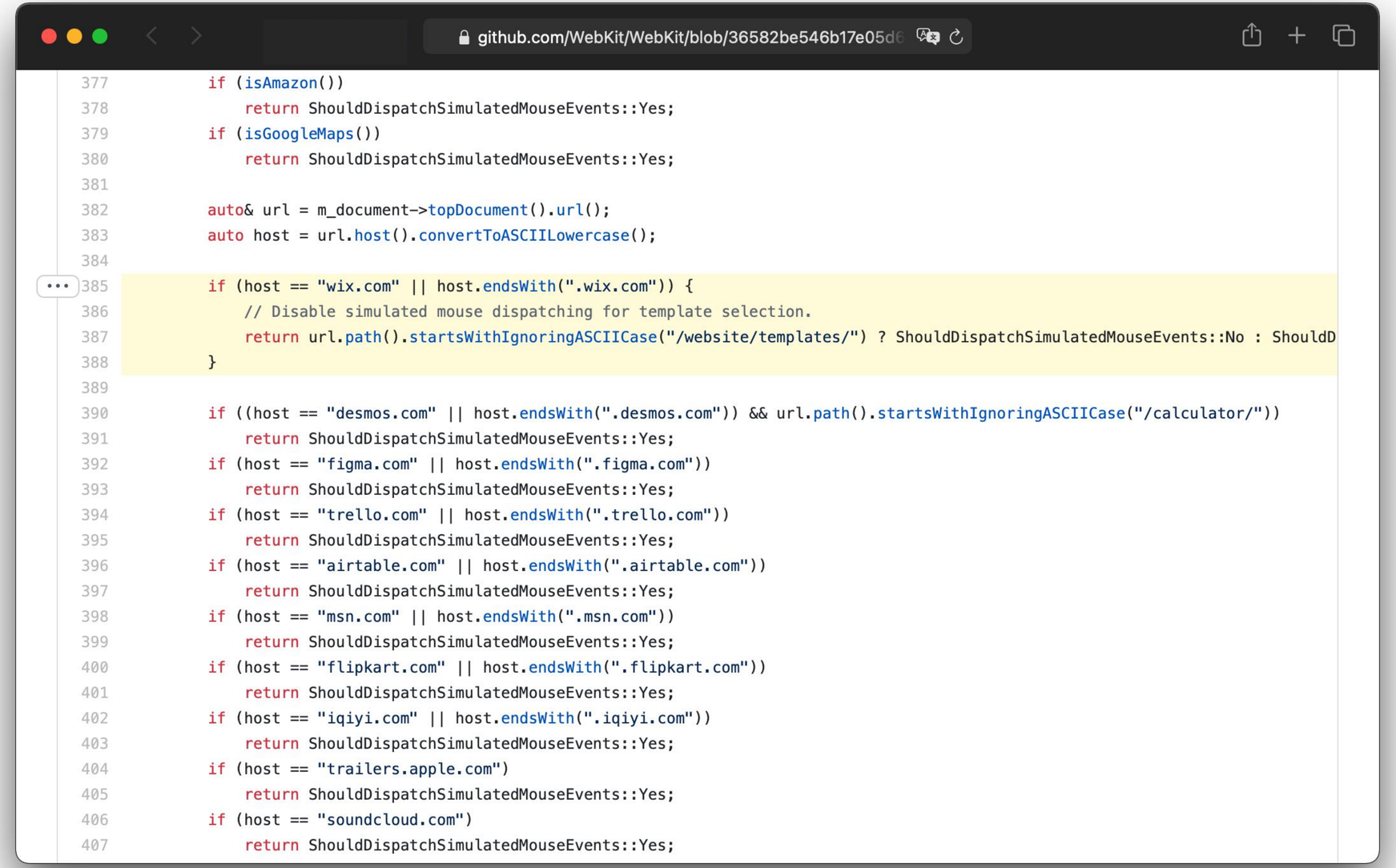

core.js — mo-dice
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
history: this.history.map((e) => this.symbols[e]),
};
}
clear() {
this.history = [];
}
}
export default Core;
Line 1, Column 1
main
Spaces: 2
JavaScript


index.html — mo-dice
1 <style>
2 body { max-width: 12em; margin: 1em auto; }
3 #lastRoll { font-size: 4em; margin: 0.5em; }
4 #log { padding: 0; }
5 #log li { display: inline-block; font-size: 2em; margin-right: 0.5em; }
6 </style>
7
8 <!-- https://xkcd.com/221/ -->
9 <h1 id="lastRoll">⚄</h1>
10 <button id="roll">🔄</button>
11 <button id="clear">🚮</button>
12 <ul id="log">
13
14 <script type="module">
15 import Core from "./core.js";
16
17 const core = new Core();
18
19 function renderRoll(core) {
20 const lastRoll = document.querySelector("#lastRoll");
21 lastRoll.innerHTML = core.pretty().lastRoll;
22 lastRoll.style.transform =
23 `translate(${Math.random() - 0.5}em, ${Math.random() - 0.5}em)`;
24 }
25
Line 44, Column 1
main
Spaces: 2
HTML


index.html — mo-dice
20 const lastRoll = document.querySelector("#lastRoll");
21 lastRoll.innerHTML = core.pretty().lastRoll;
22 lastRoll.style.transform =
23 `translate(${Math.random() - 0.5}em, ${Math.random() - 0.5}em)`;
24 }
25
26 function renderHistory(core) {
27 document.querySelector("#log").innerHTML = core
28 .pretty()
29 .history.map((v) => `<li>${v}</li>`)
30 .join("");
31 }
32
33 document.querySelector("#roll").addEventListener("click", () => {
34 core.roll();
35 renderRoll(core);
36 renderHistory(core);
37 });
38
39 document.querySelector("#clear").addEventListener("click", () => {
40 core.clear();
41 renderHistory(core);
42 });
43 </script>
44
Line 7, Column 1
main
Spaces: 2
HTML




www.ecma-international.org/publications-and-st
Industry association for standardizing information and communication systems
ecma INTERNATIONAL
About Ecma
Publications and standards
Committees
Policies
News
Contact
Back to the list
ECMA-262
ECMAScript® 2020 language specification
11th edition, June 2020
This Standard defines the ECMAScript 2020 general-purpose programming language.
Kindly note that the normative copy is the HTML version; the PDF version has been produced to generate a printable document.

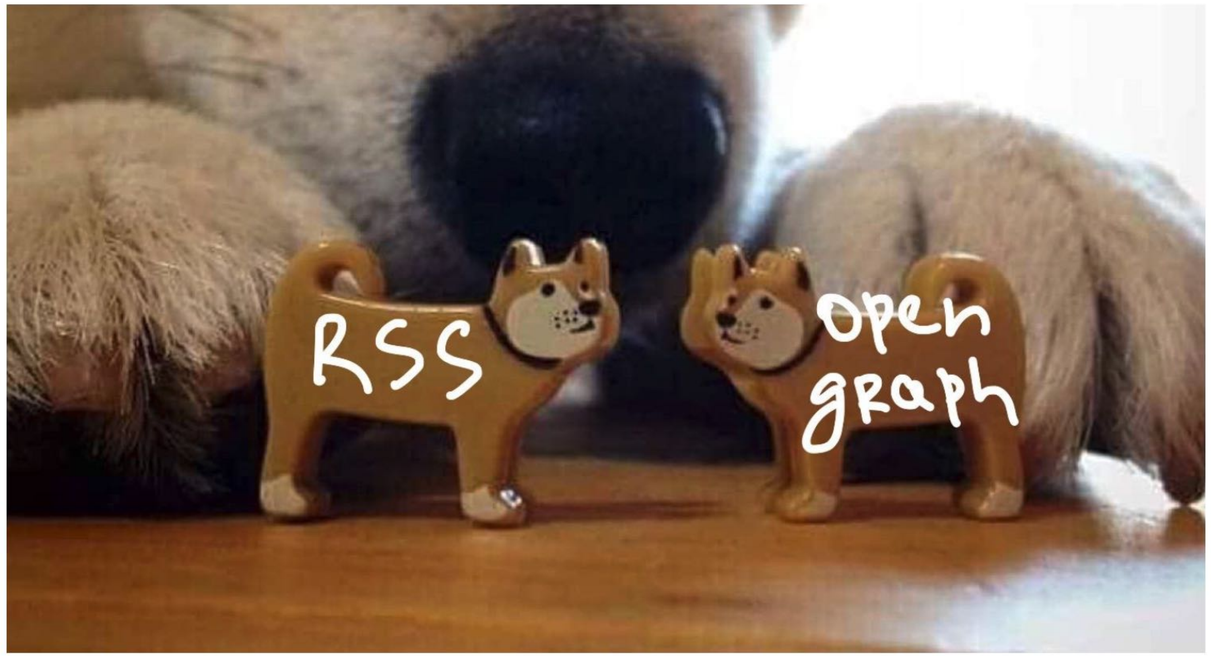



```
377         if (isAmazon())
378             return ShouldDispatchSimulatedMouseEvents::Yes;
379         if (isGoogleMaps())
380             return ShouldDispatchSimulatedMouseEvents::Yes;
381
382         auto& url = m_document->topDocument().url();
383         auto host = url.host().convertToASCIILowercase();
384
385         if (host == "wix.com" || host.endsWith(".wix.com")) {
386             // Disable simulated mouse dispatching for template selection.
387             return url.path().startsWithIgnoringASCIICase("/website/templates/") ? ShouldDispatchSimulatedMouseEvents::No : ShouldD
388         }
389
390         if ((host == "desmos.com" || host.endsWith(".desmos.com")) && url.path().startsWithIgnoringASCIICase("/calculator/"))
391             return ShouldDispatchSimulatedMouseEvents::Yes;
392         if (host == "figma.com" || host.endsWith(".figma.com"))
393             return ShouldDispatchSimulatedMouseEvents::Yes;
394         if (host == "trello.com" || host.endsWith(".trello.com"))
395             return ShouldDispatchSimulatedMouseEvents::Yes;
396         if (host == "airtable.com" || host.endsWith(".airtable.com"))
397             return ShouldDispatchSimulatedMouseEvents::Yes;
398         if (host == "msn.com" || host.endsWith(".msn.com"))
399             return ShouldDispatchSimulatedMouseEvents::Yes;
400         if (host == "flipkart.com" || host.endsWith(".flipkart.com"))
401             return ShouldDispatchSimulatedMouseEvents::Yes;
402         if (host == "iqiyi.com" || host.endsWith(".iqiyi.com"))
403             return ShouldDispatchSimulatedMouseEvents::Yes;
404         if (host == "trailers.apple.com")
405             return ShouldDispatchSimulatedMouseEvents::Yes;
406         if (host == "soundcloud.com")
407             return ShouldDispatchSimulatedMouseEvents::Yes;
```

2015

zemlan.in

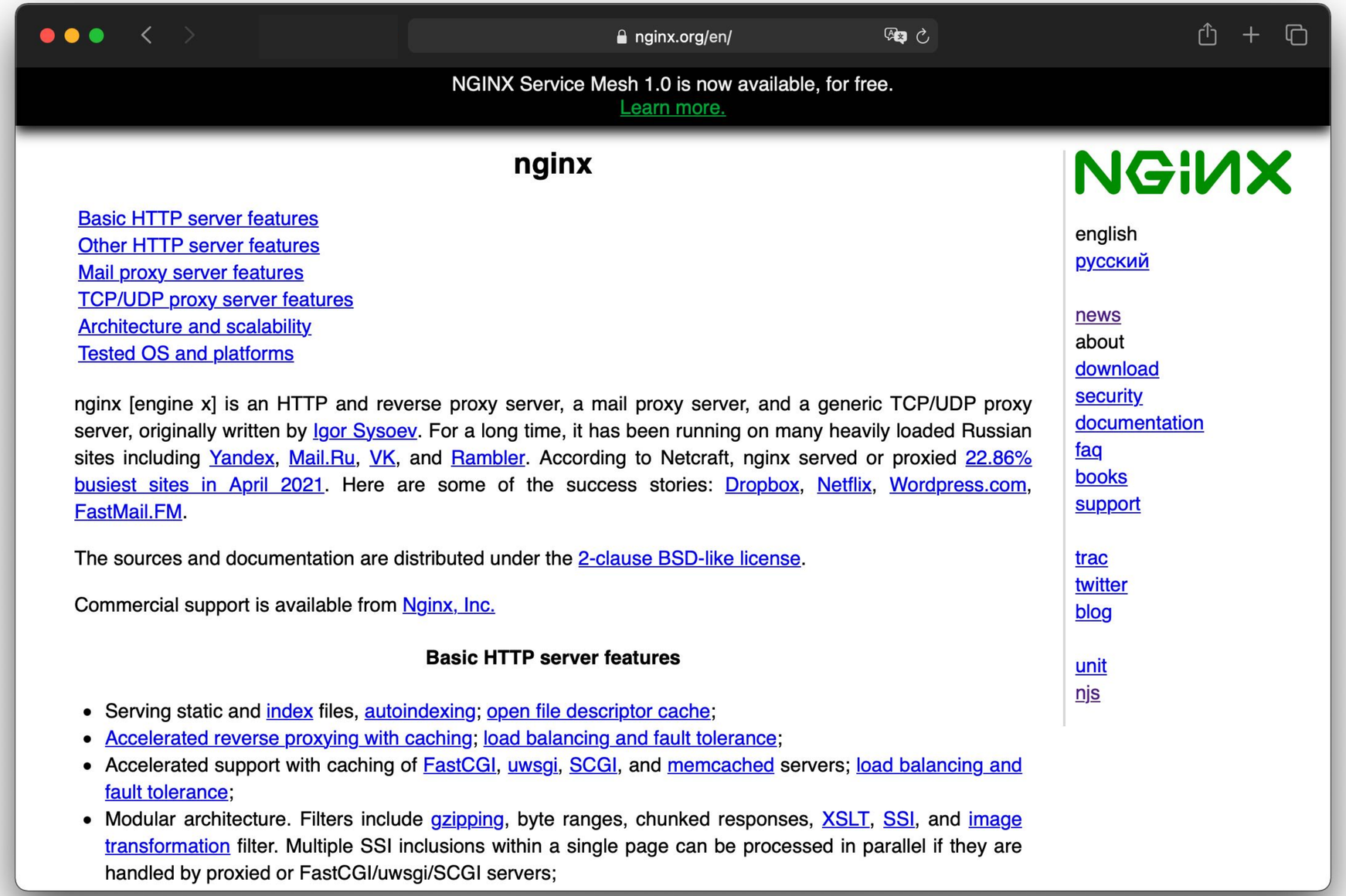
Склепал линкблог



...из скуки и открытых стандартов

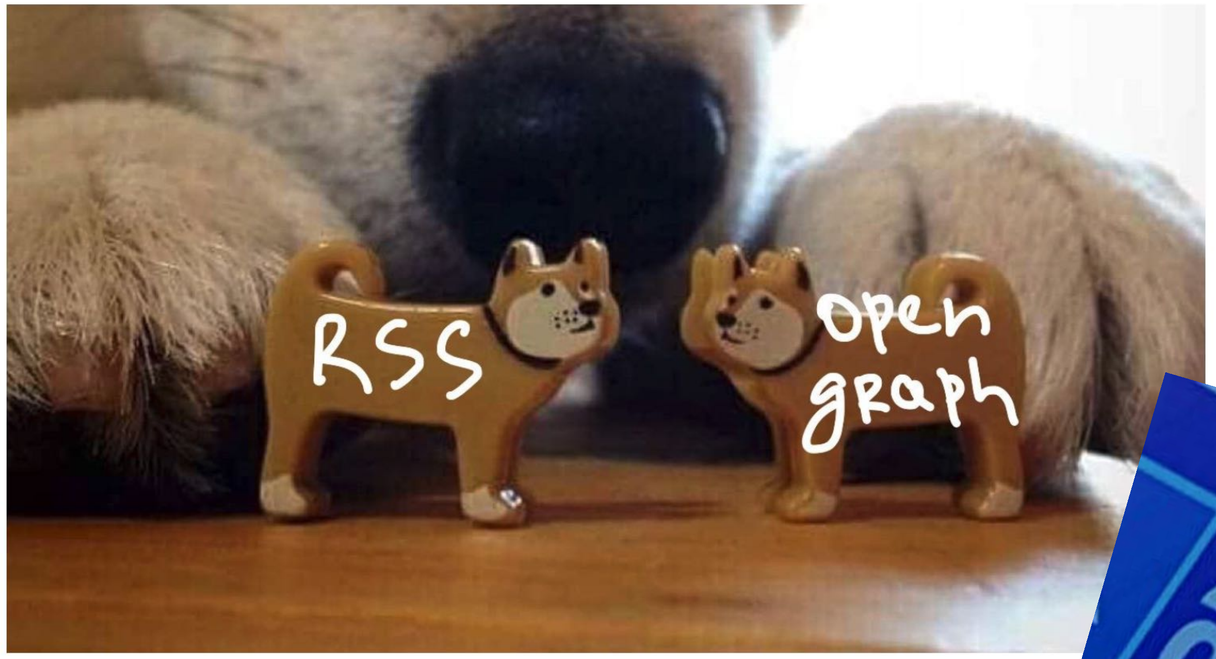
816 слов →

2015-04-19


nginx.org/en/
NGINX Service Mesh 1.0 is now available, for free.
Learn more.
nginx
Basic HTTP server features
Other HTTP server features
Mail proxy server features
TCP/UDP proxy server features
Architecture and scalability
Tested OS and platforms
nginx [engine x] is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server, originally written by Igor Sysoev. For a long time, it has been running on many heavily loaded Russian sites including Yandex, Mail.Ru, VK, and Rambler. According to Netcraft, nginx served or proxied 22.86% busiest sites in April 2021. Here are some of the success stories: Dropbox, Netflix, Wordpress.com, FastMail.FM.
The sources and documentation are distributed under the 2-clause BSD-like license.
Commercial support is available from Nginx, Inc.
Basic HTTP server features
Serving static and index files, autoindexing; open file descriptor cache;
Accelerated reverse proxying with caching; load balancing and fault tolerance;
Accelerated support with caching of FastCGI, uwsgi, SCGI, and memcached servers; load balancing and fault tolerance;
Modular architecture. Filters include gzipping, byte ranges, chunked responses, XSLT, SSI, and image transformation filter. Multiple SSI inclusions within a single page can be processed in parallel if they are handled by proxied or FastCGI/uwsgi/SCGI servers;
NGINX
english
русский
news
about
download
security
documentation
faq
books
support
trac
twitter
blog
unit
njs

zemlan.in

Склепал линкблог

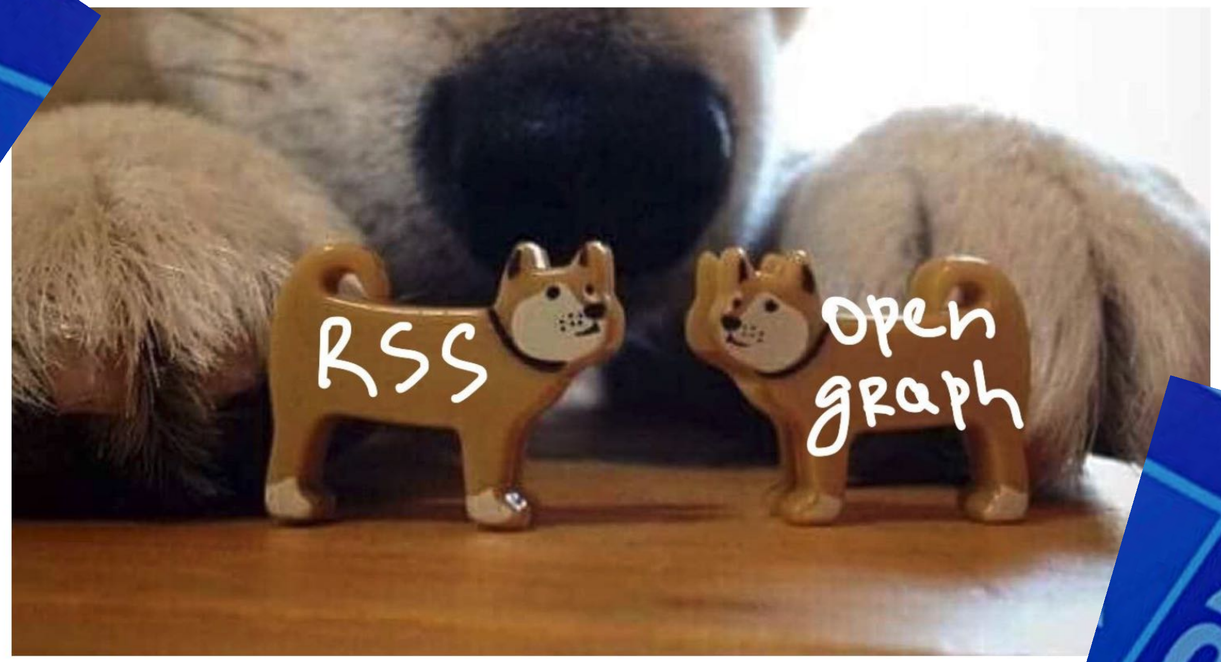


...из скуки и открытых стандартов

816 слов →

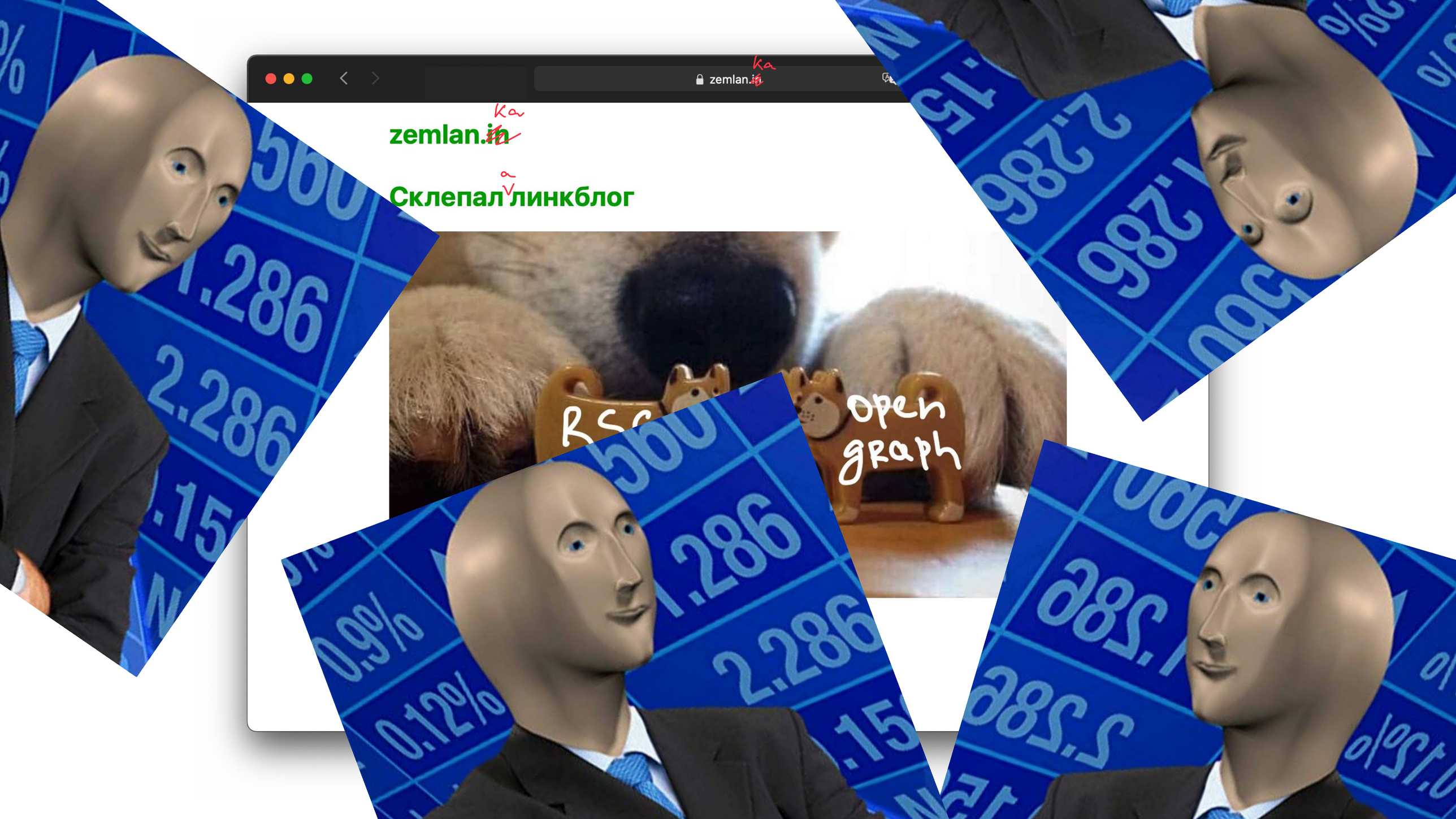
zemlan.in

Склепал линкблог



...из скуки и открытых стандартов

816 слов →


zemlan.in
zemlan.in
Склепал линкблог
RSS
open graph

About

português · news · showcase · uses · quotes · press · authors · contact · license · versions · thanks · getting started

❖ What is Lua?

Lua is a powerful, efficient, lightweight, embeddable scripting language. It supports procedural programming, object-oriented programming, functional programming, data-driven programming, and data description.

Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode with a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

❖ Where does Lua come from?

Lua is designed, implemented, and maintained by a team at PUC-Rio, the Pontifical Catholic University of Rio de Janeiro in Brazil. Lua was born and raised in Tecgraf, formerly the Computer Graphics Technology Group of PUC-Rio. Lua is now housed at LabLua, a laboratory of the Department of Computer Science of PUC-Rio.

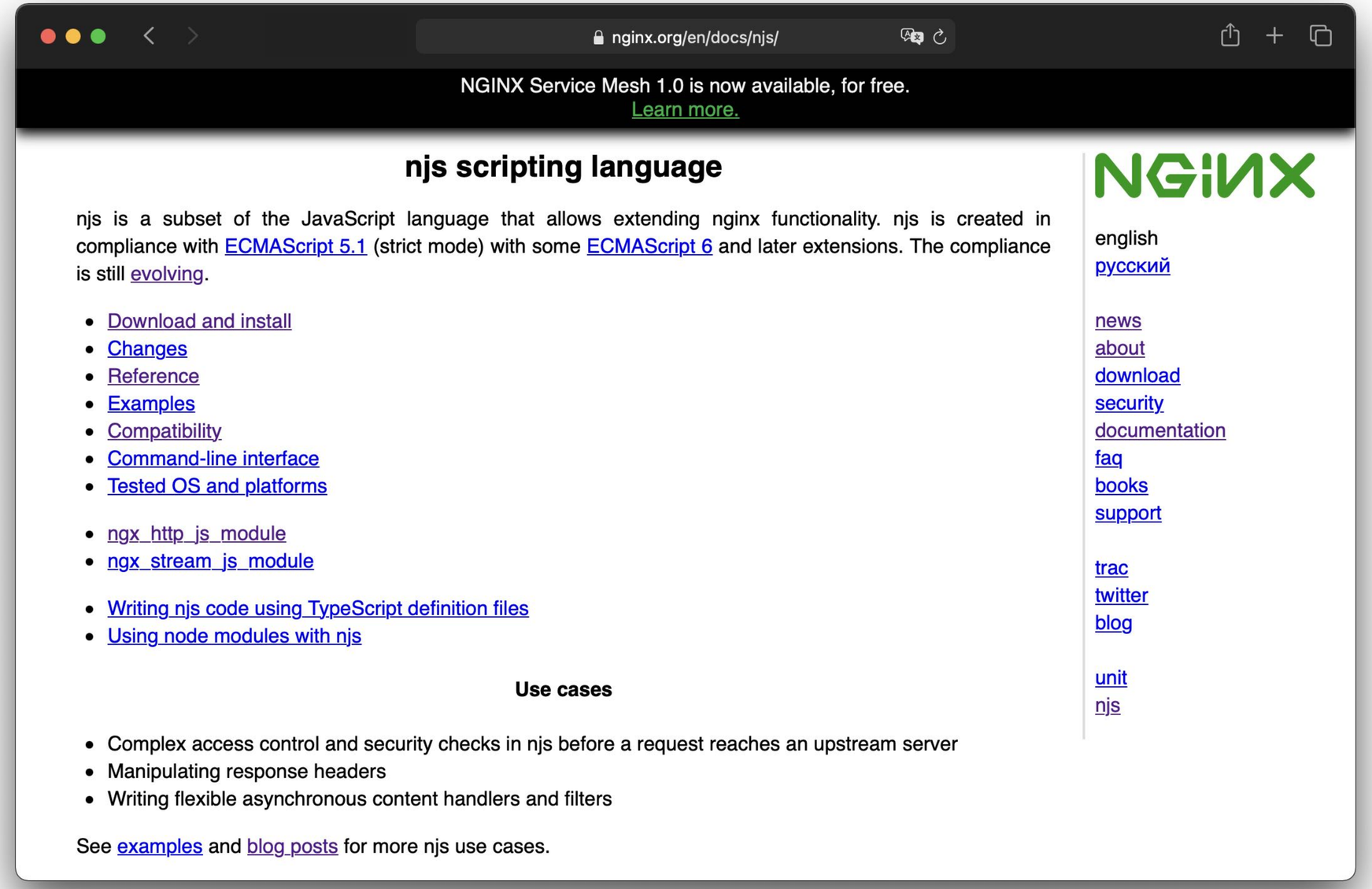
❖ Why choose Lua?

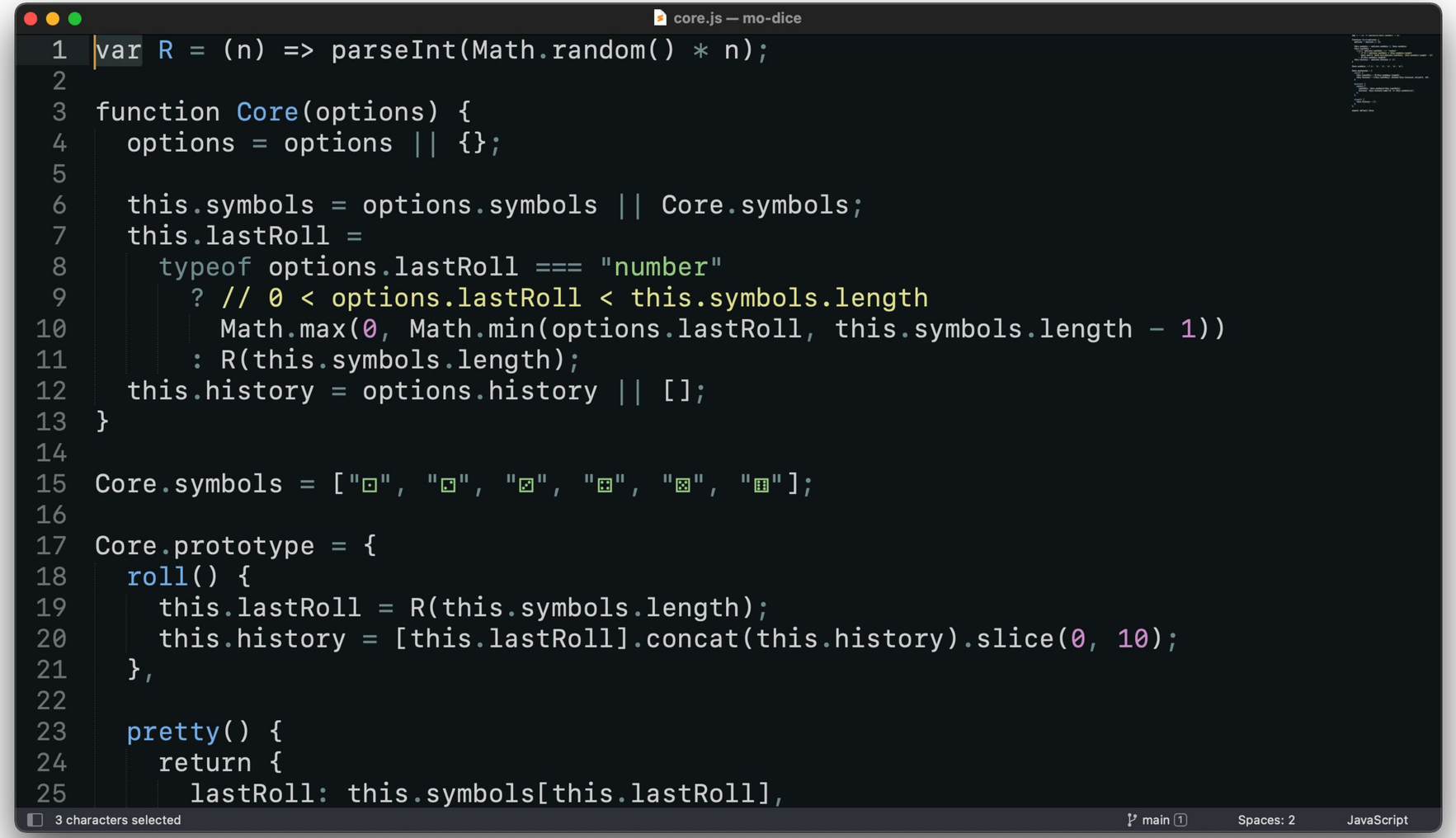
Lua is a proven, robust language

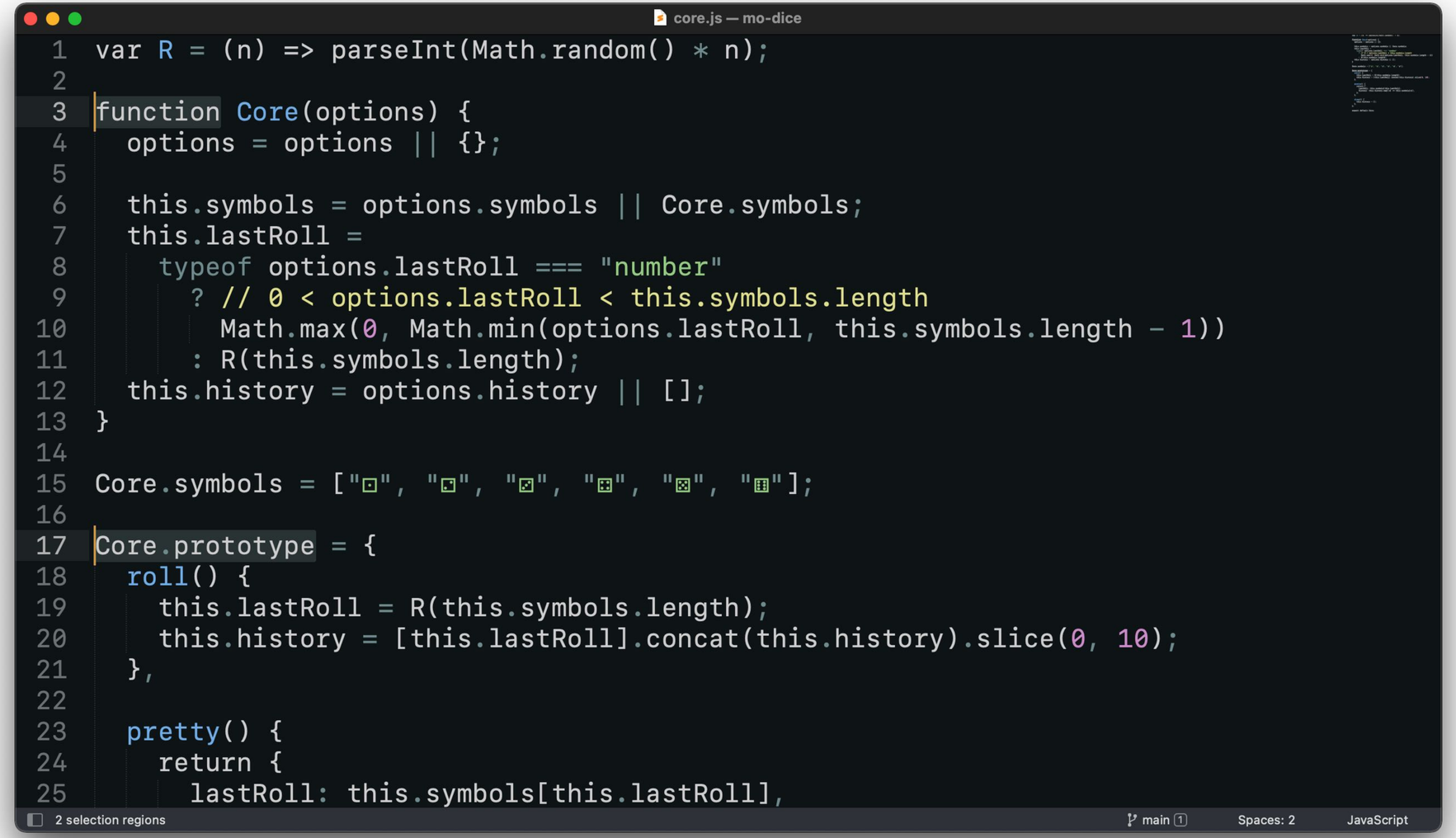
Lua has been used in many industrial applications (e.g., Adobe's Photoshop Lightroom), with an emphasis on embedded systems (e.g., the Ginga middleware for digital TV in Brazil) and games (e.g., World of Warcraft and Angry Birds). Lua is currently the leading scripting language in games. Lua has a solid reference manual and there are several books about it. Several versions of Lua have been released and used in real applications since its creation in 1993. Lua featured in HOPL III, the Third ACM SIGPLAN History of Programming Languages Conference, in 2007. Lua won the Front Line Award 2011 from the Game Developers Magazine.

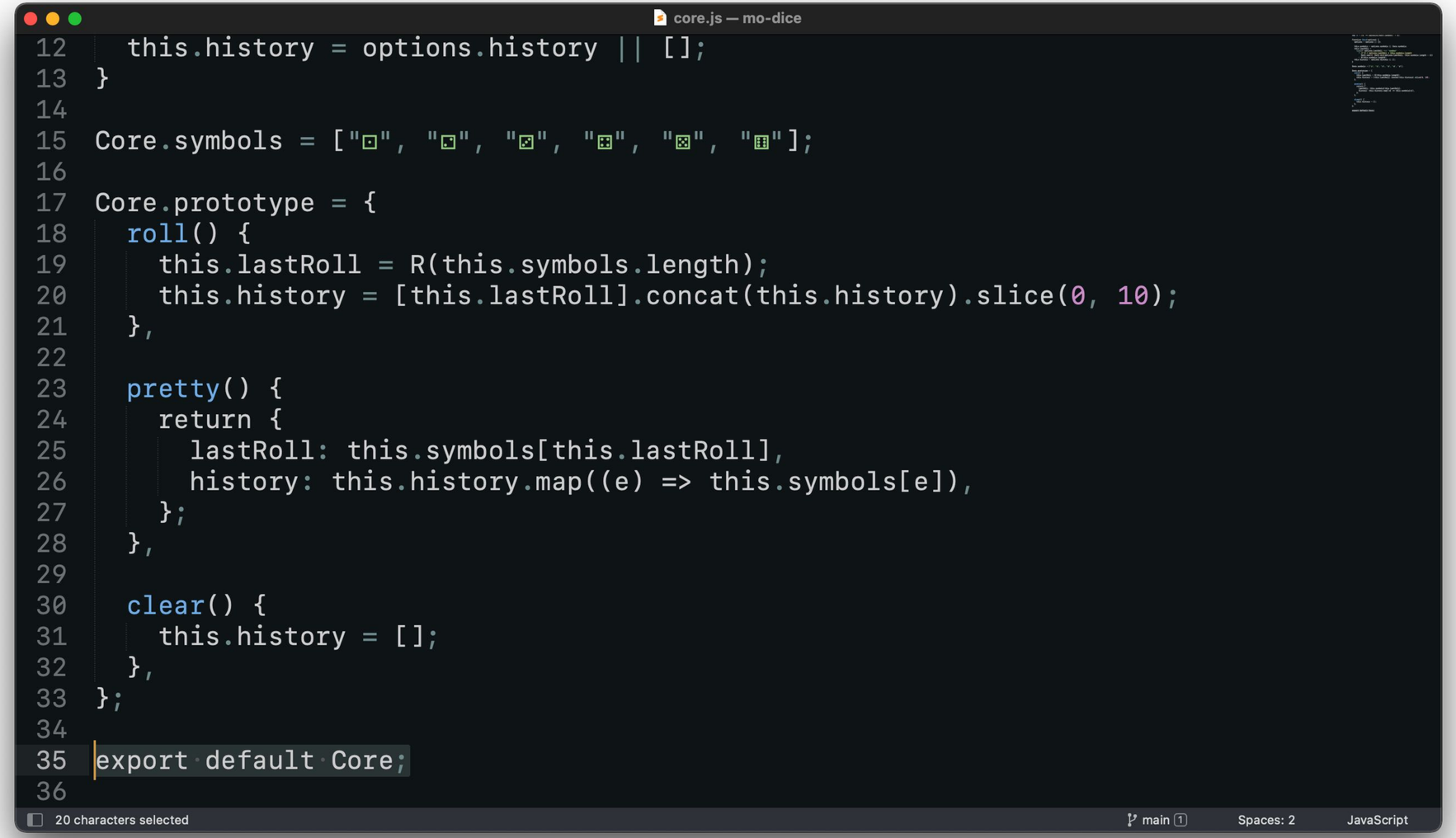
Lua is fast

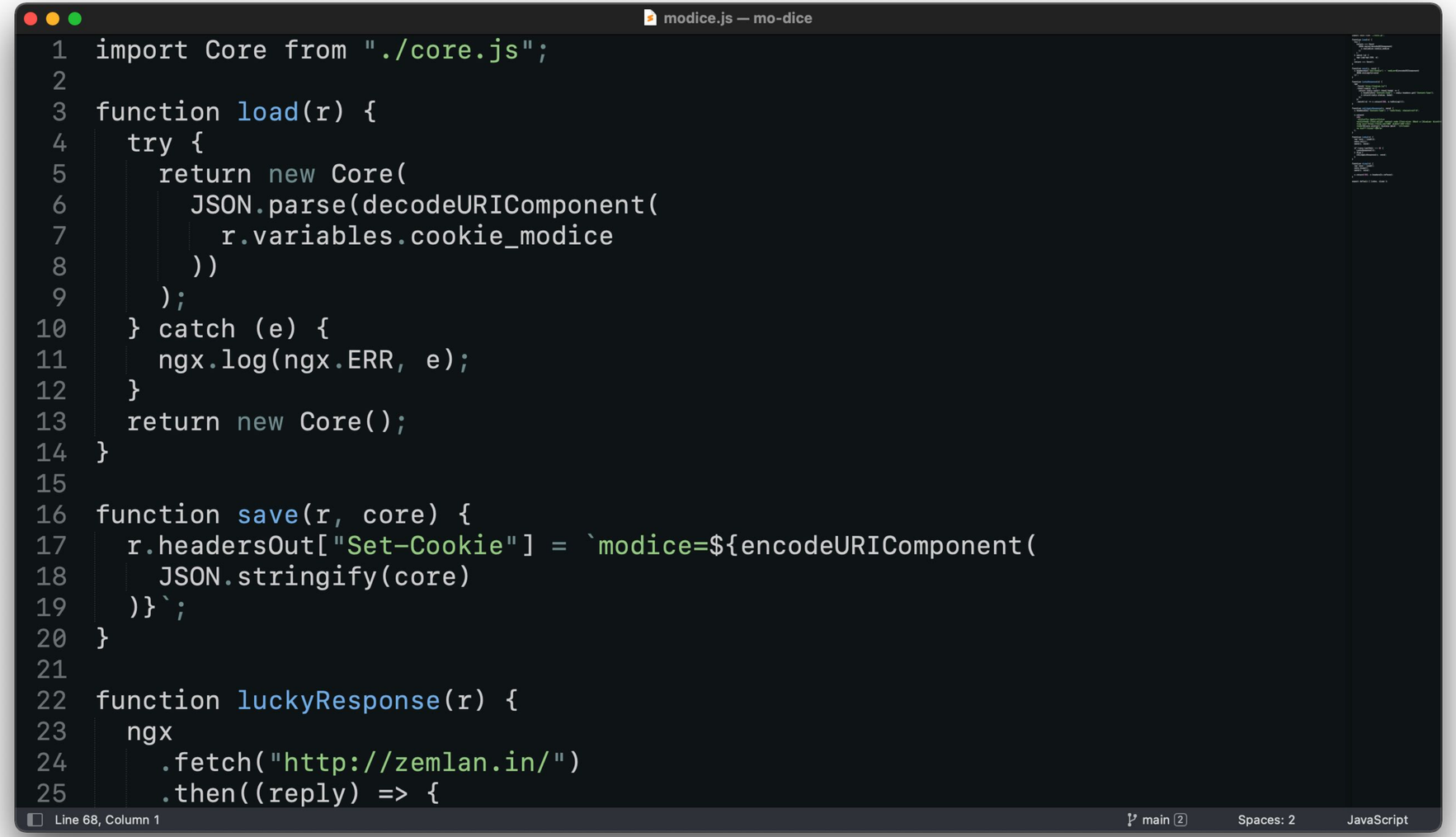
Lua has a deserved reputation for performance. To claim to be "as fast as Lua" is an aspiration of other scripting languages. Several benchmarks show Lua as the fastest language in the realm of interpreted scripting languages. Lua is fast not only in fine-tuned benchmark programs, but in real life too. Substantial fractions of large applications have been written in Lua.


nginx.org/en/docs/njs/
NGINX Service Mesh 1.0 is now available, for free.
Learn more.
njs scripting language
njs is a subset of the JavaScript language that allows extending nginx functionality. njs is created in compliance with ECMAScript 5.1 (strict mode) with some ECMAScript 6 and later extensions. The compliance is still evolving.
Download and install
Changes
Reference
Examples
Compatibility
Command-line interface
Tested OS and platforms
ngx_http_js_module
ngx_stream_js_module
Writing njs code using TypeScript definition files
Using node modules with njs
Use cases
Complex access control and security checks in njs before a request reaches an upstream server
Manipulating response headers
Writing flexible asynchronous content handlers and filters
See examples and blog posts for more njs use cases.
NGINX
english
русский
news
about
download
security
documentation
faq
books
support
trac
twitter
blog
unit
njs


core.js — mo-dice
var R = (n) => parseInt(Math.random() * n);
function Core(options) {
options = options || {};
this.symbols = options.symbols || Core.symbols;
this.lastRoll =
typeof options.lastRoll === "number"
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
Core.symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
Core.prototype = {
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll].concat(this.history).slice(0, 10);
},
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
3 characters selected
main 1
Spaces: 2
JavaScript


core.js — mo-dice
var R = (n) => parseInt(Math.random() * n);
function Core(options) {
options = options || {};
this.symbols = options.symbols || Core.symbols;
this.lastRoll =
typeof options.lastRoll === "number"
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
Core.symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
Core.prototype = {
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll].concat(this.history).slice(0, 10);
},
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
2 selection regions
main 1
Spaces: 2
JavaScript


core.js — mo-dice
this.history = options.history || [];
}
Core.symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
Core.prototype = {
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll].concat(this.history).slice(0, 10);
},
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
history: this.history.map((e) => this.symbols[e]),
};
},
clear() {
this.history = [];
},
};
export default Core;
20 characters selected
main 1
Spaces: 2
JavaScript


modice.js — mo-dice
import Core from "./core.js";
function load(r) {
try {
return new Core(
JSON.parse(decodeURIComponent(
r.variables.cookie_modice
))
);
} catch (e) {
ngx.log(ngx.ERR, e);
}
return new Core();
}
function save(r, core) {
r.headersOut["Set-Cookie"] = `modice=${encodeURIComponent(
JSON.stringify(core)
)}`;
}
function luckyResponse(r) {
ngx
.fetch("http://zemlan.in/")
.then((reply) => {
Line 68, Column 1
main 2
Spaces: 2
JavaScript


modice.js — mo-dice
44   );
45 }
46
47 function index(r) {
48   var core = load(r);
49   core.roll();
50   save(r, core);
51
52   if (core.lastRoll === 0) {
53   luckyResponse(r);
54   } else {
55   rollAgainResponse(r, core);
56   }
57 }
58
59 function clear(r) {
60   var core = load();
61   core.clear();
62   save(r, core);
63
64   r.return(303, r.headersIn.referer);
65 }
66
67 export default { index, clear };
68
Line 2, Column 1
main 2
Spaces: 2
JavaScript

modice.js — mo-dice

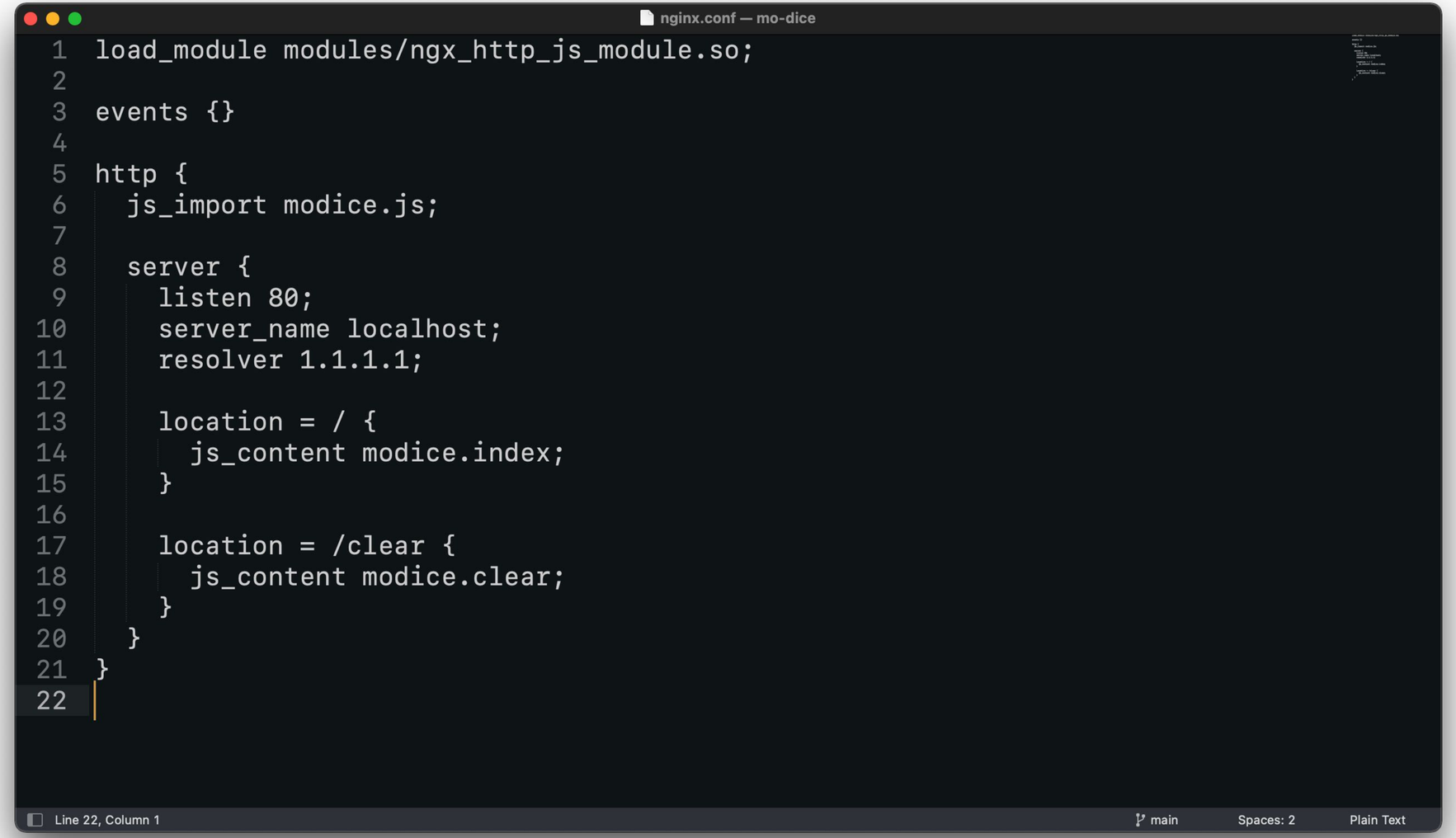
```
function luckyResponse(r) {
  ngx
    .fetch("http://zemlan.in/")
    .then((reply) => {
      return reply.text().then((body) => {
        r.headersOut["Content-Type"] = reply.headers.get("Content-Type");
        r.return(reply.status, body);
      });
    })
    .catch((e) => r.return(500, e.toString()));
}

function rollAgainResponse(r, core) {
  r.headersOut["Content-Type"] = "text/html; charset=utf-8";

  r.return(
    403,
    `<title>Try Again</title>
    <style>body {text-align: center} code {font-size: 80px} a {display: block}</
    <img src="https://http.cat/403" width="403"><hr>
    <code>${core.pretty().history.join(" ")}</code>
    <a href="/clear">🚮</a>`
  );
}

```

Line 68, Column 1 main 2 Spaces: 2 JavaScript


nginx.conf — mo-dice
load_module modules/ngx_http_js_module.so;
events {}
http {
js_import modice.js;
server {
listen 80;
server_name localhost;
resolver 1.1.1.1;
location = / {
js_content modice.index;
}
location = /clear {
js_content modice.clear;
}
}
}
Line 22, Column 1
main
Spaces: 2
Plain Text

zemlan.ka
i

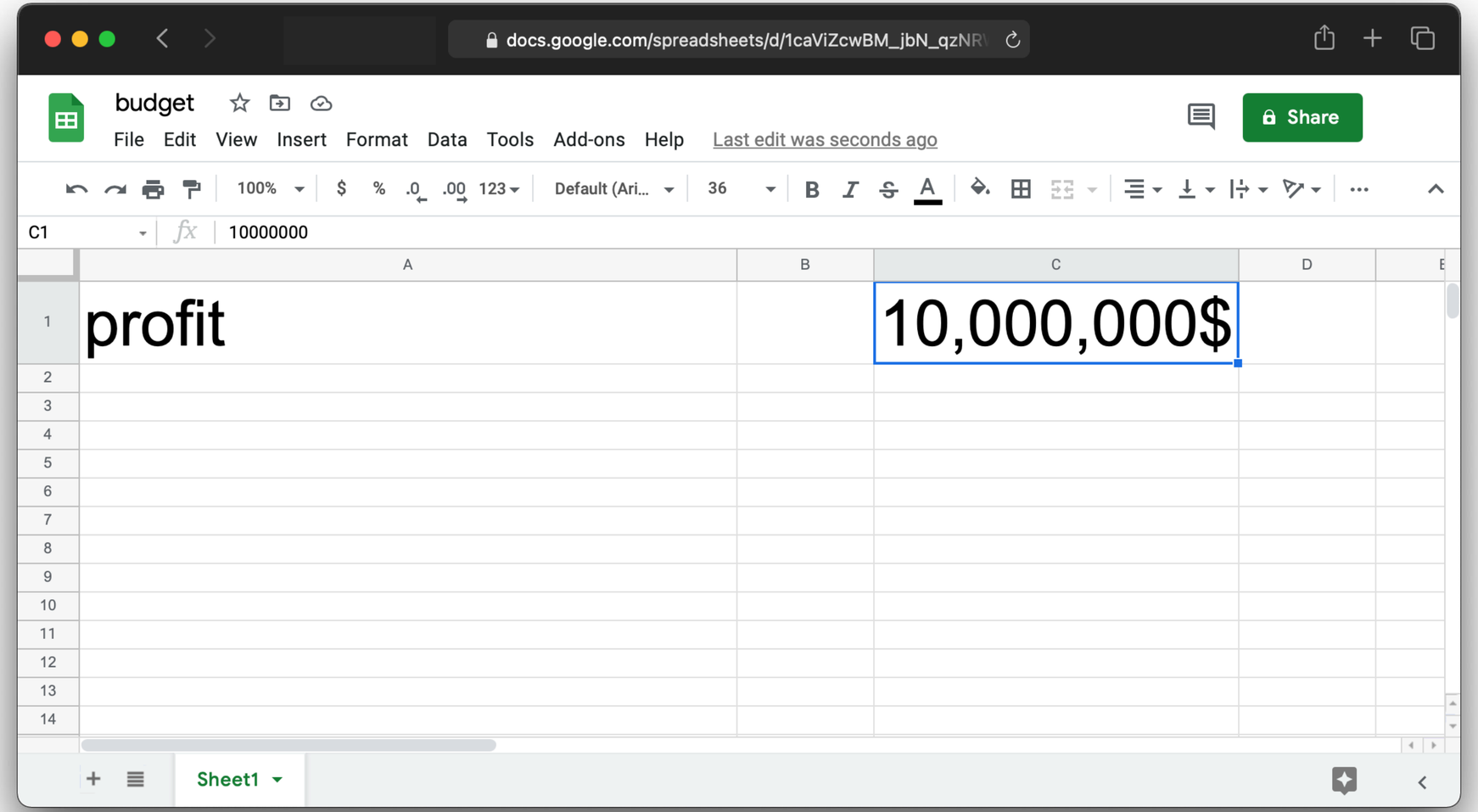
zemlan.in

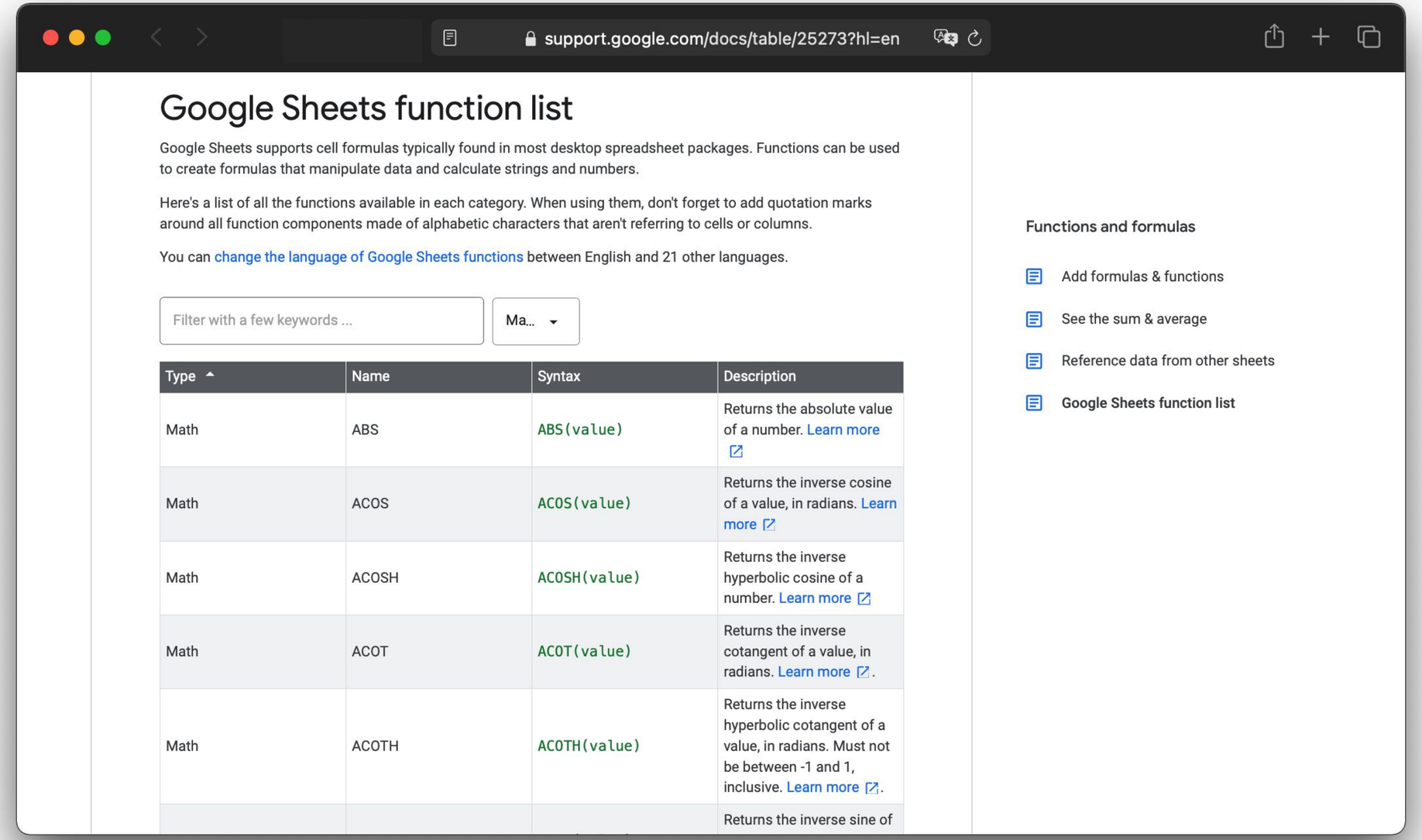
zemlan.in

Склепал линкблог

zemlan.

2017


docs.google.com/spreadsheets/d/1caViZcwBM_jbN_qzNR
budget
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100%
Default (Ari...
36
C1
10000000
A
B
C
D
profit
10,000,000$
Sheet1



Google Sheets function list

Google Sheets supports cell formulas typically found in most desktop spreadsheet packages. Functions can be used to create formulas that manipulate data and calculate strings and numbers.

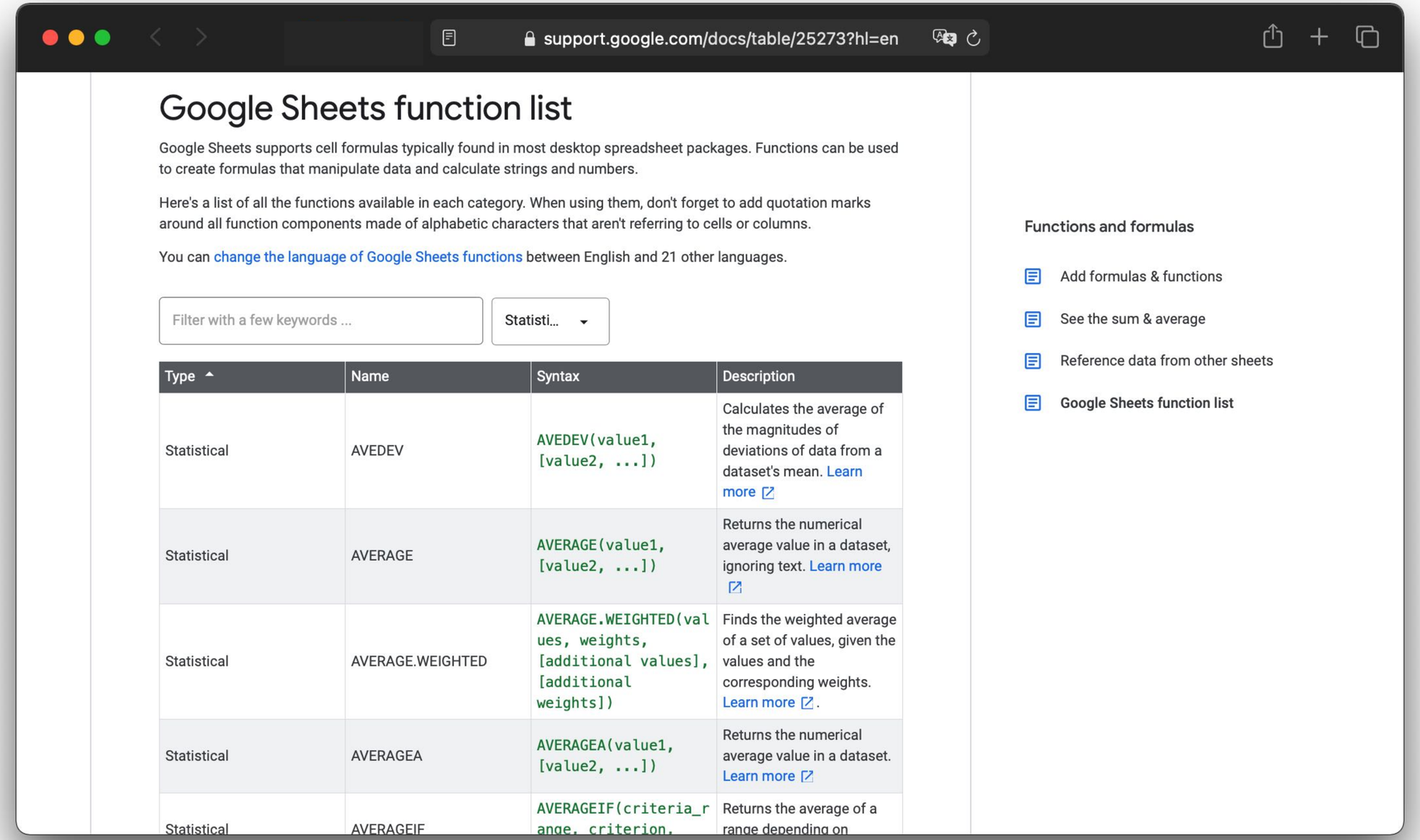
Here's a list of all the functions available in each category. When using them, don't forget to add quotation marks around all function components made of alphabetic characters that aren't referring to cells or columns.

You can change the language of Google Sheets functions between English and 21 other languages.

Filter with a few keywords ...

Ma...

Type	Name	Syntax	Description
Math	ABS	`ABS(value)`	Returns the absolute value of a number. Learn more
Math	ACOS	`ACOS(value)`	Returns the inverse cosine of a value, in radians. Learn more
Math	ACOSH	`ACOSH(value)`	Returns the inverse hyperbolic cosine of a number. Learn more
Math	ACOT	`ACOT(value)`	Returns the inverse cotangent of a value, in radians. Learn more.
Math	ACOTH	`ACOTH(value)`	Returns the inverse hyperbolic cotangent of a value, in radians. Must not be between -1 and 1, inclusive. Learn more.
			Returns the inverse sine of



Google Sheets function list

Google Sheets supports cell formulas typically found in most desktop spreadsheet packages. Functions can be used to create formulas that manipulate data and calculate strings and numbers.

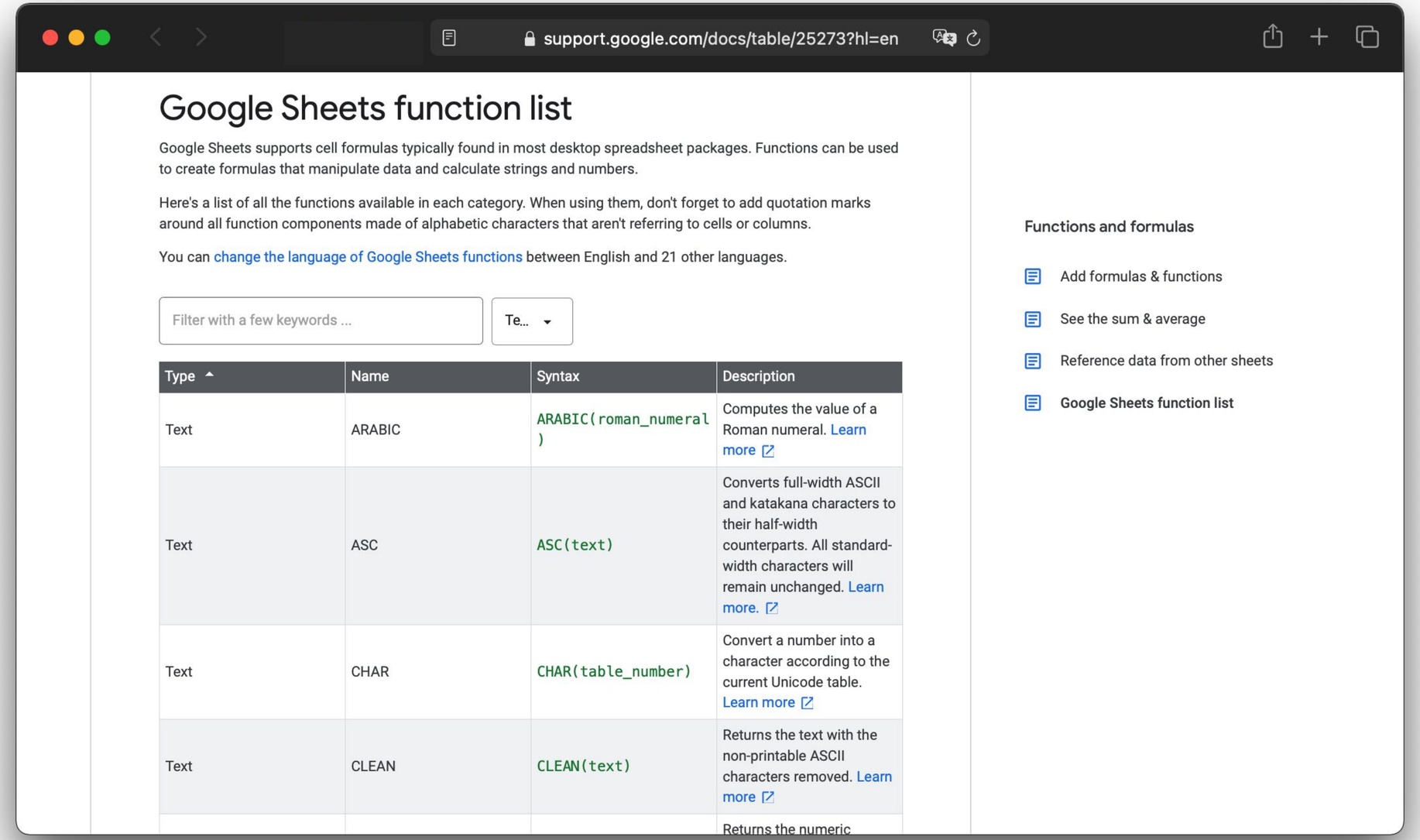
Here's a list of all the functions available in each category. When using them, don't forget to add quotation marks around all function components made of alphabetic characters that aren't referring to cells or columns.

You can change the language of Google Sheets functions between English and 21 other languages.

Filter with a few keywords ...

Statisti...

Type	Name	Syntax	Description
Statistical	AVEDEV	`AVEDEV(value1, [value2, ...])`	Calculates the average of the magnitudes of deviations of data from a dataset's mean. Learn more
Statistical	AVERAGE	`AVERAGE(value1, [value2, ...])`	Returns the numerical average value in a dataset, ignoring text. Learn more
Statistical	AVERAGE.WEIGHTED	`AVERAGE.WEIGHTED(values, weights, [additional values], [additional weights])`	Finds the weighted average of a set of values, given the values and the corresponding weights. Learn more.
Statistical	AVERAGEA	`AVERAGEA(value1, [value2, ...])`	Returns the numerical average value in a dataset. Learn more
Statistical	AVERAGEIF	`AVERAGEIF(criteria_range, criterion,`	Returns the average of a range depending on



Google Sheets function list

Google Sheets supports cell formulas typically found in most desktop spreadsheet packages. Functions can be used to create formulas that manipulate data and calculate strings and numbers.

Here's a list of all the functions available in each category. When using them, don't forget to add quotation marks around all function components made of alphabetic characters that aren't referring to cells or columns.

You can change the language of Google Sheets functions between English and 21 other languages.

Filter with a few keywords ...

Te...

Type	Name	Syntax	Description
Text	ARABIC	`ARABIC(roman_numeral)`	Computes the value of a Roman numeral. Learn more
Text	ASC	`ASC(text)`	Converts full-width ASCII and katakana characters to their half-width counterparts. All standard-width characters will remain unchanged. Learn more.
Text	CHAR	`CHAR(table_number)`	Convert a number into a character according to the current Unicode table. Learn more
Text	CLEAN	`CLEAN(text)`	Returns the text with the non-printable ASCII characters removed. Learn more
			Returns the numeric

Functions and formulas

- Add formulas & functions
- See the sum & average
- Reference data from other sheets
- **Google Sheets function list**

GOOGLEFINANCE

Fetches current or historical securities information from Google Finance.

Sample Usage

```
GOOGLEFINANCE("NASDAQ:GOOG", "price", DATE(2014,1,1), DATE(2014,12,31), "DAILY")
GOOGLEFINANCE("NASDAQ:GOOG","price",TODAY()-30,TODAY())
GOOGLEFINANCE(A2,A3)
```

Syntax

```
GOOGLEFINANCE(ticker, [attribute], [start_date], [end_date|num_days], [interval])
```

- `ticker` - The ticker symbol for the security to consider. It's **mandatory** to use **both** the exchange symbol and ticker symbol for accurate results and to avoid discrepancies. For example, use "NASDAQ:GOOG" instead of "GOOG."
  - If the exchange symbol is not specified, `GOOGLEFINANCE` will use its best judgement to choose one for you.
  - **Note:** Reuters Instrument Codes are no longer supported. For example, use TSE:123 or ASX:XYZ instead of ticker 123.TO or XYZ.AX.
- `attribute` - [OPTIONAL - `"price"` by default] - The attribute to fetch about `ticker` from Google Finance and is required if a date is specified.
  - `attribute` is one of the following for real-time data:
    - `"price"` - Real-time price quote, delayed by up to 20 minutes.
    - `"priceopen"` - The price as of market open.
    - `"high"` - The current day's high price.

IMPORTXML

Imports data from any of various structured data types including XML, HTML, CSV, TSV, and RSS and ATOM XML feeds.

Sample Usage

```
IMPORTXML("https://en.wikipedia.org/wiki/Moon_landing", "//a/@href")
```

```
IMPORTXML(A2,B2)
```

Syntax

```
IMPORTXML(url, xpath_query)
```

- `url` - The URL of the page to examine, including protocol (e.g. `http://`).
  - The value for `url` must either be enclosed in quotation marks or be a reference to a cell containing the appropriate text.
- `xpath_query` - The XPath query to run on the structured data.
  - For more information on XPath, see http://www.w3schools.com/xml/xpath_intro.asp.

See Also

`IMPORTRANGE`: Imports a range of cells from a specified spreadsheet.

`IMPORTHTML`: Imports data from a table or list within an HTML page.

`IMPORTFEED`: Imports a RSS or ATOM feed.

`IMPORTDATA`: Imports data at a given url in .csv (comma-separated value) or .tsv (tab-separated value) format.

Web

Google Sheets function list

ENCODEURL function

HYPERLINK

IMPORTDATA

IMPORTFEED

IMPORTHTML

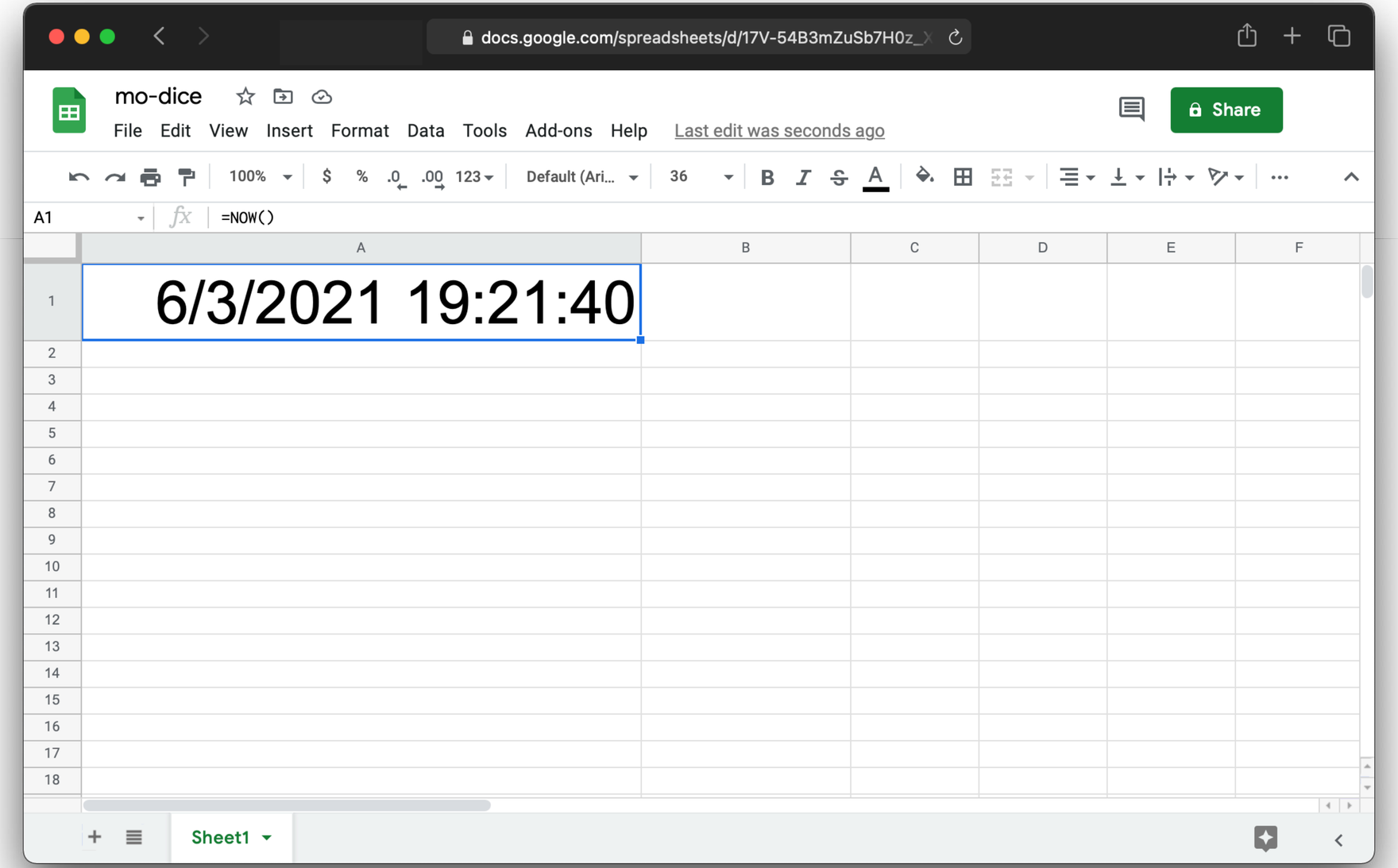
IMPORTRANGE

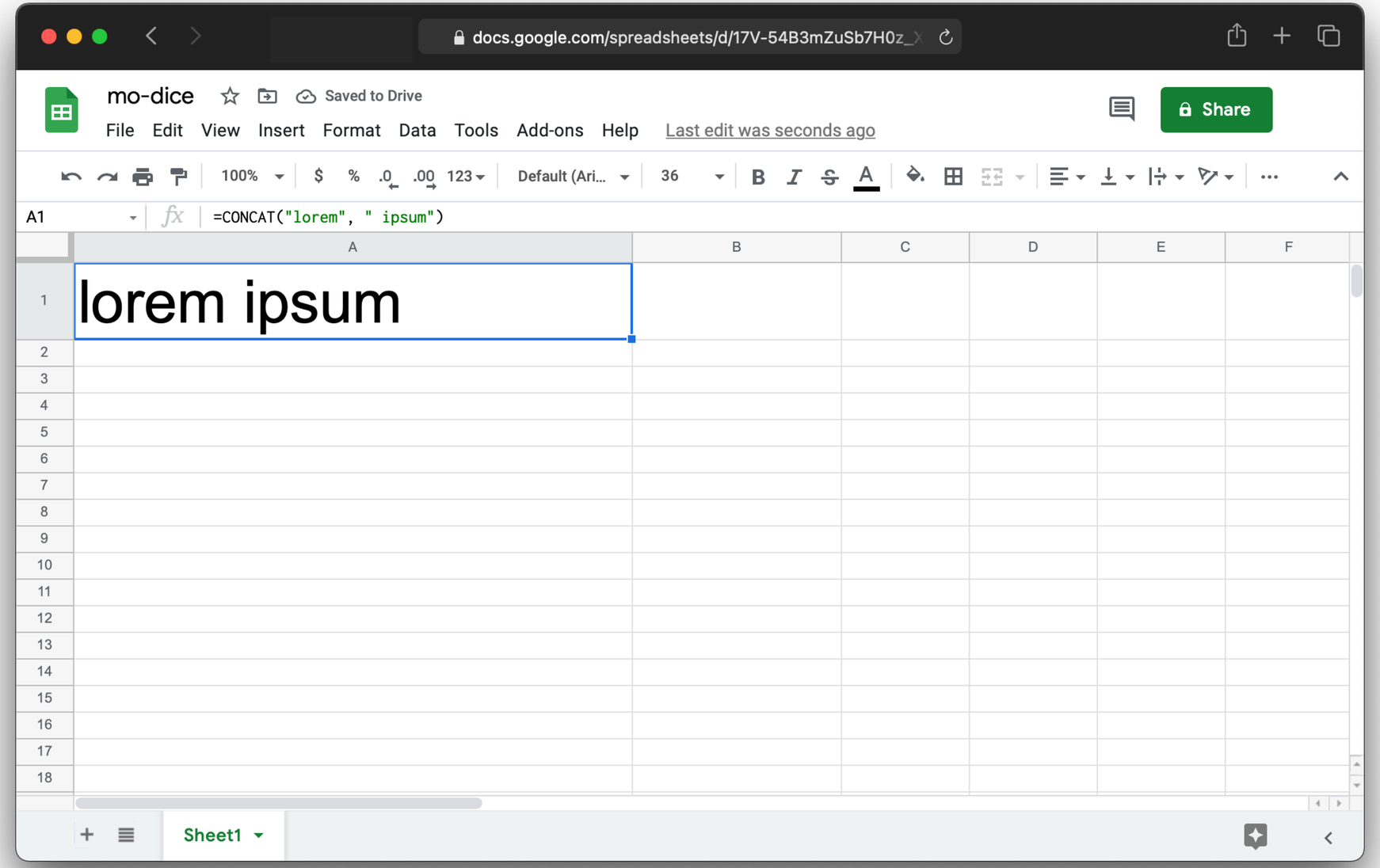
IMPORTXML

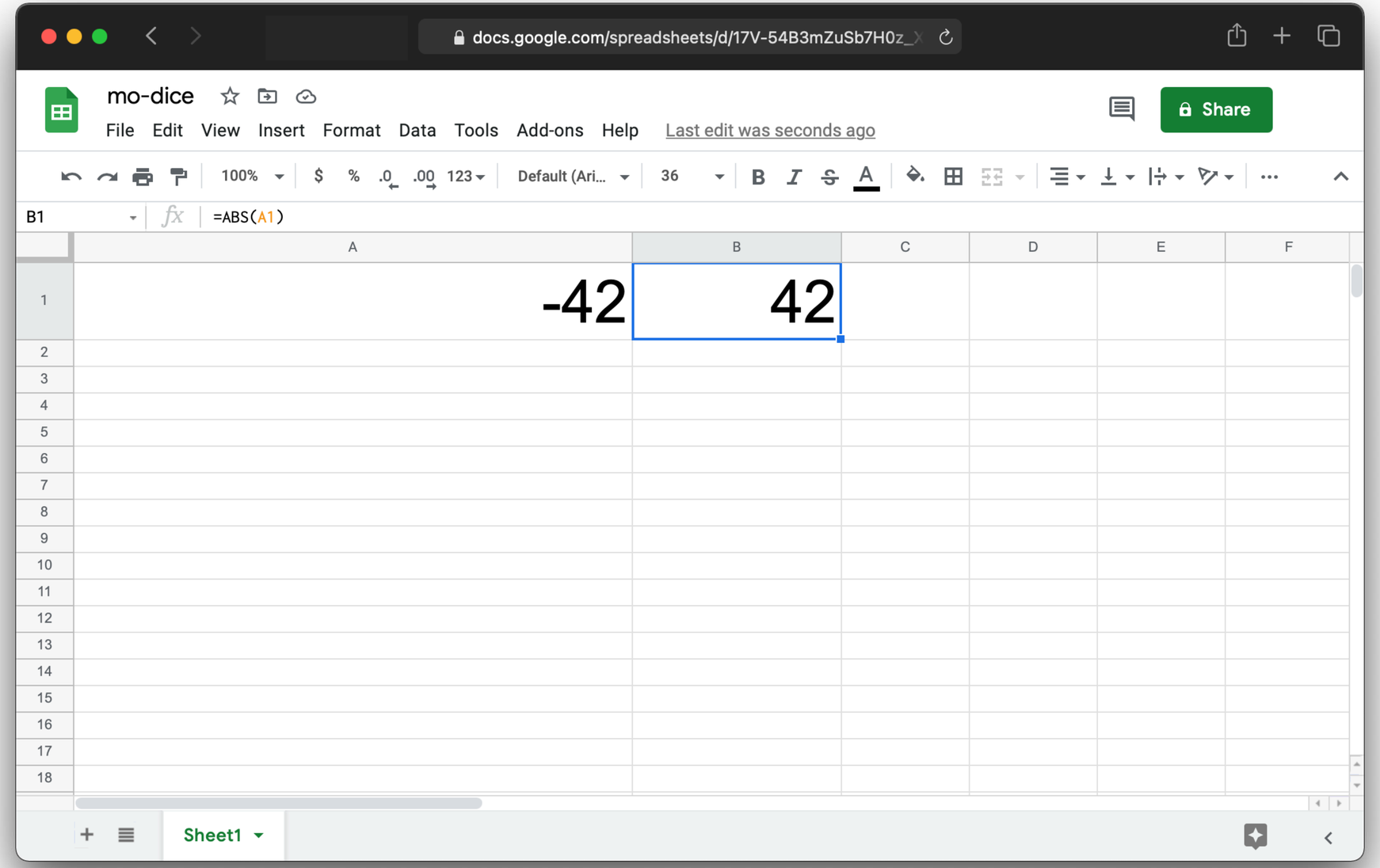
ISURL

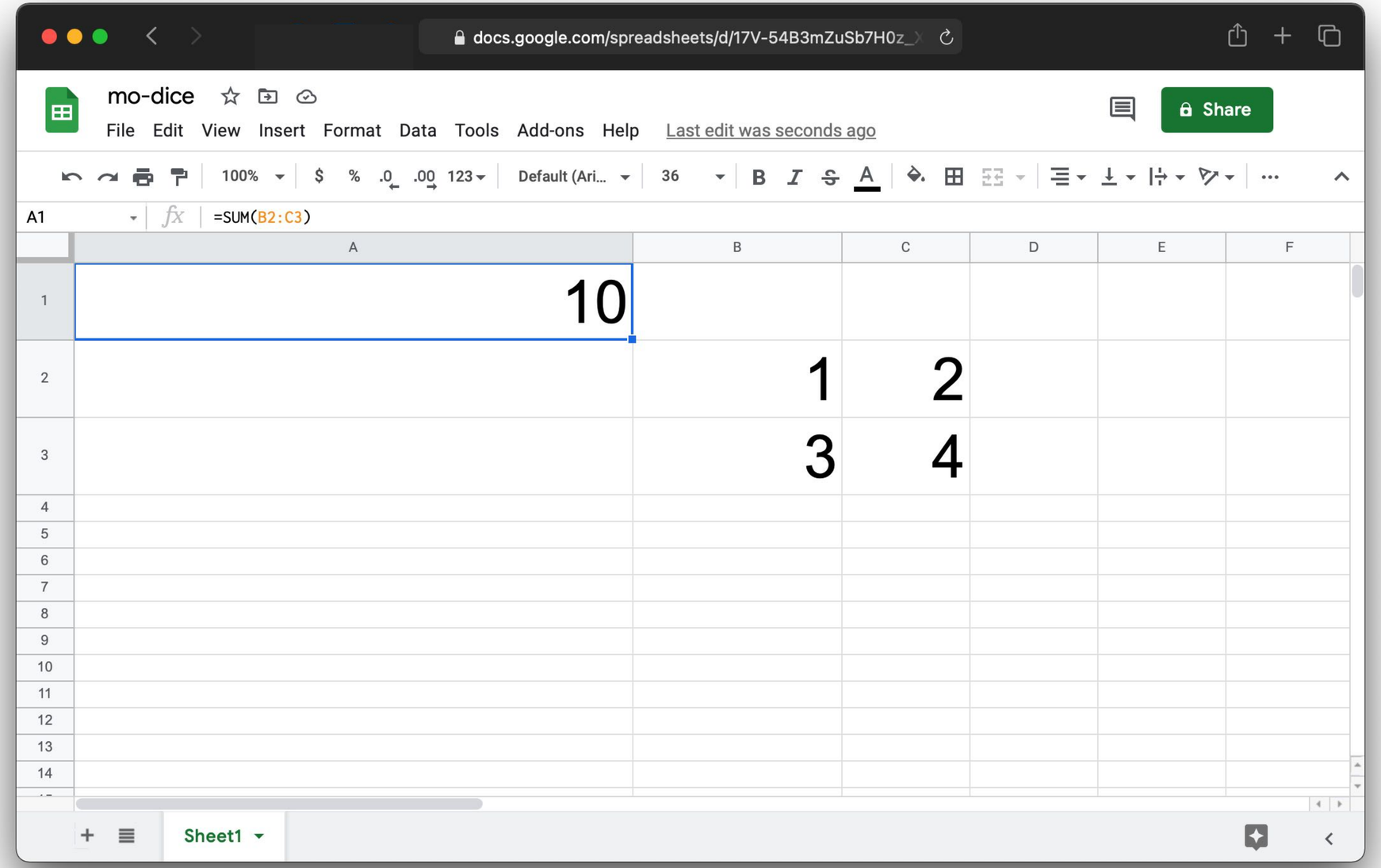
Visit the Learning Center

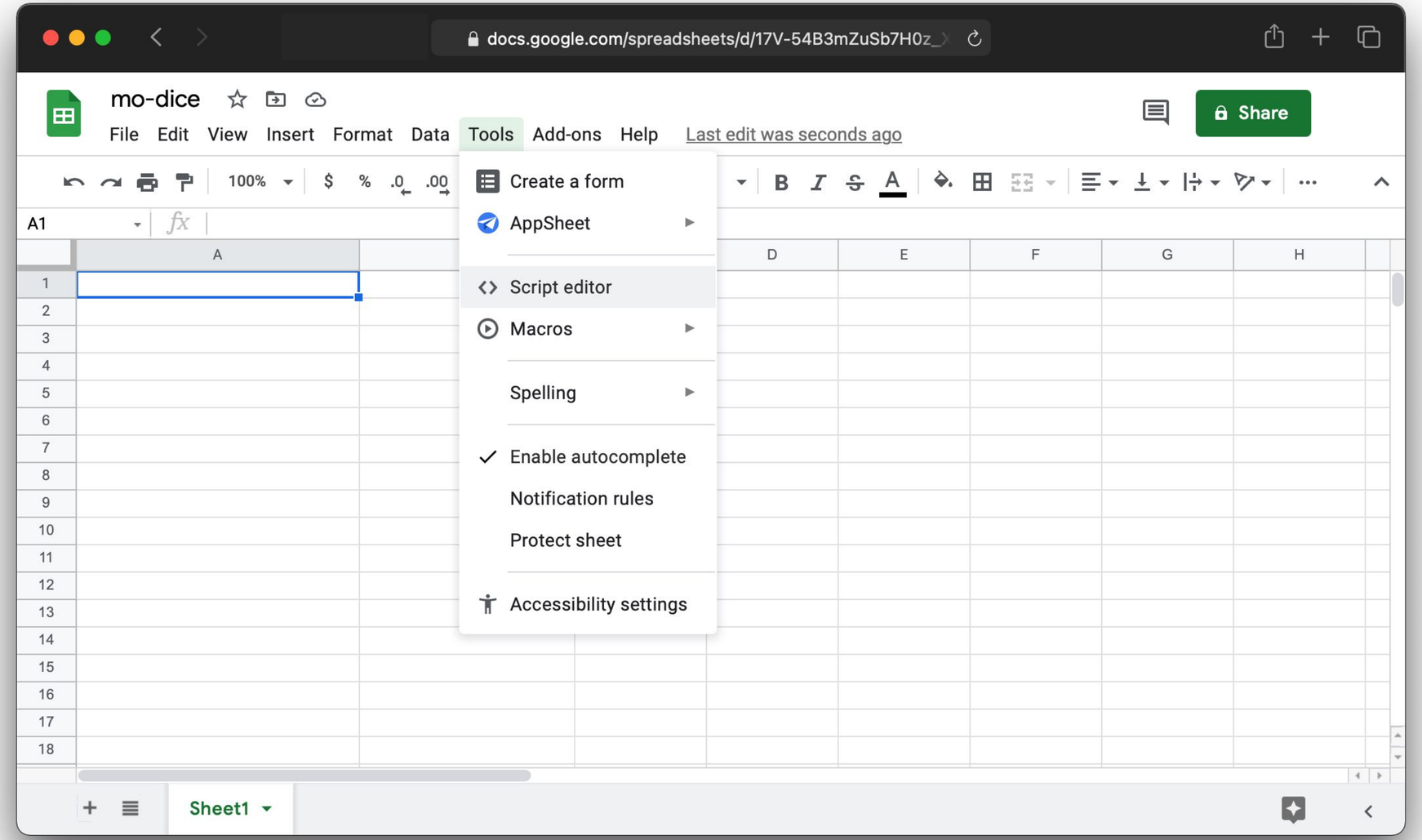
Using Google products, like Google Docs, at work or school? Try powerful tips, tutorials, and templates. Learn to work on

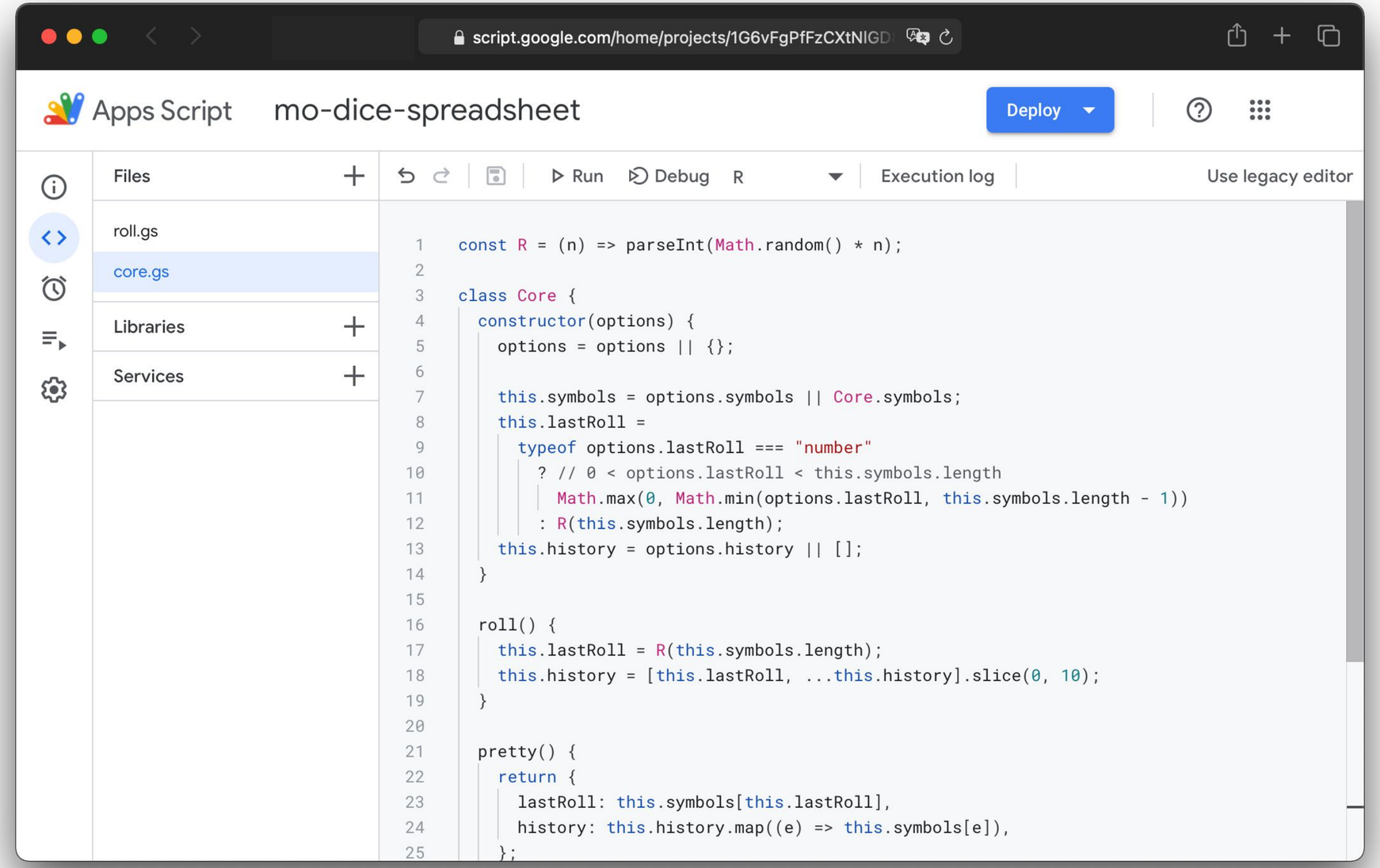

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
A1
=NOW()
A B C D E F
1 6/3/2021 19:21:40
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Sheet1

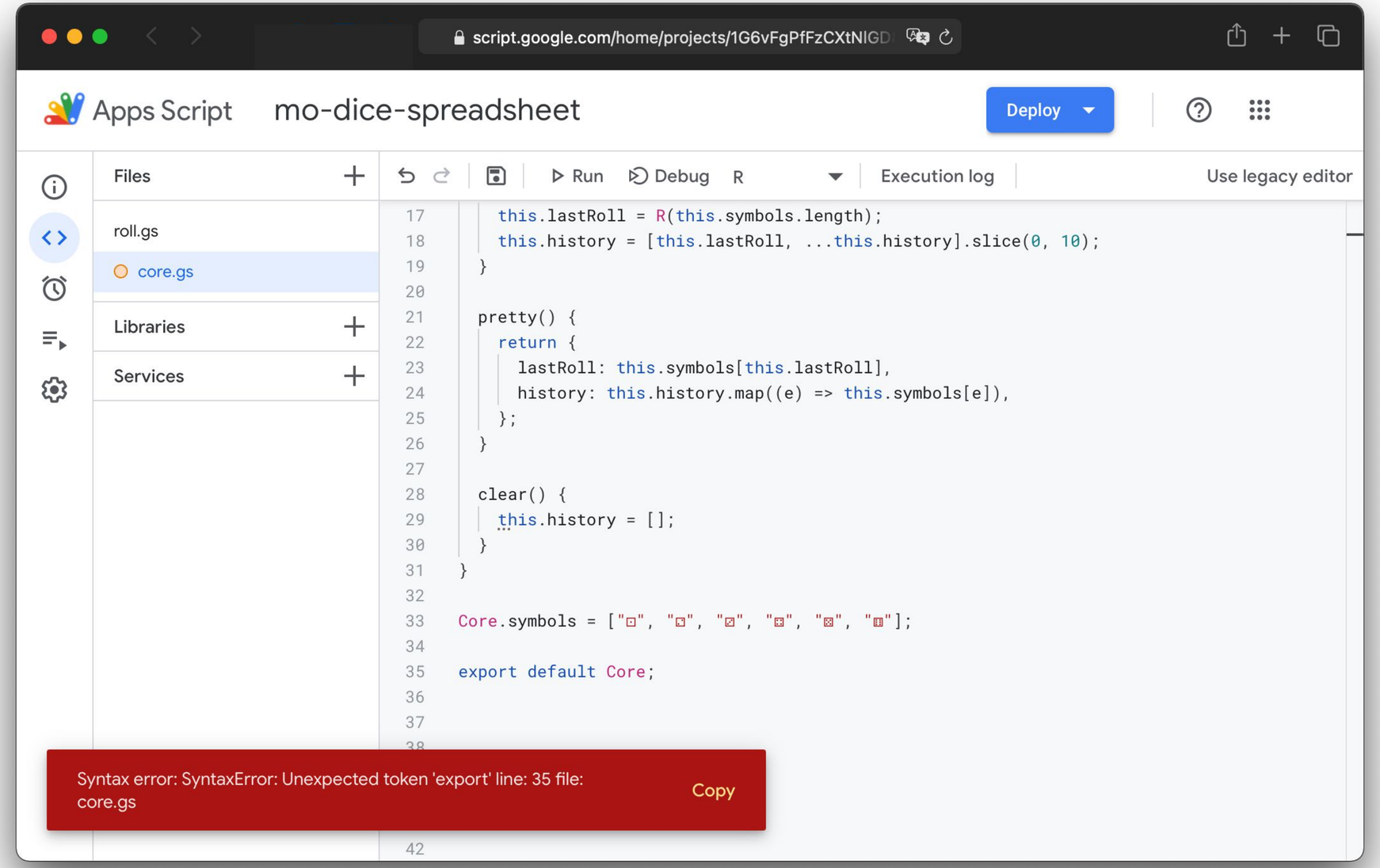

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
Saved to Drive
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
A1
=CONCAT("lorem", " ipsum")
A B C D E F
1 lorem ipsum
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Sheet1

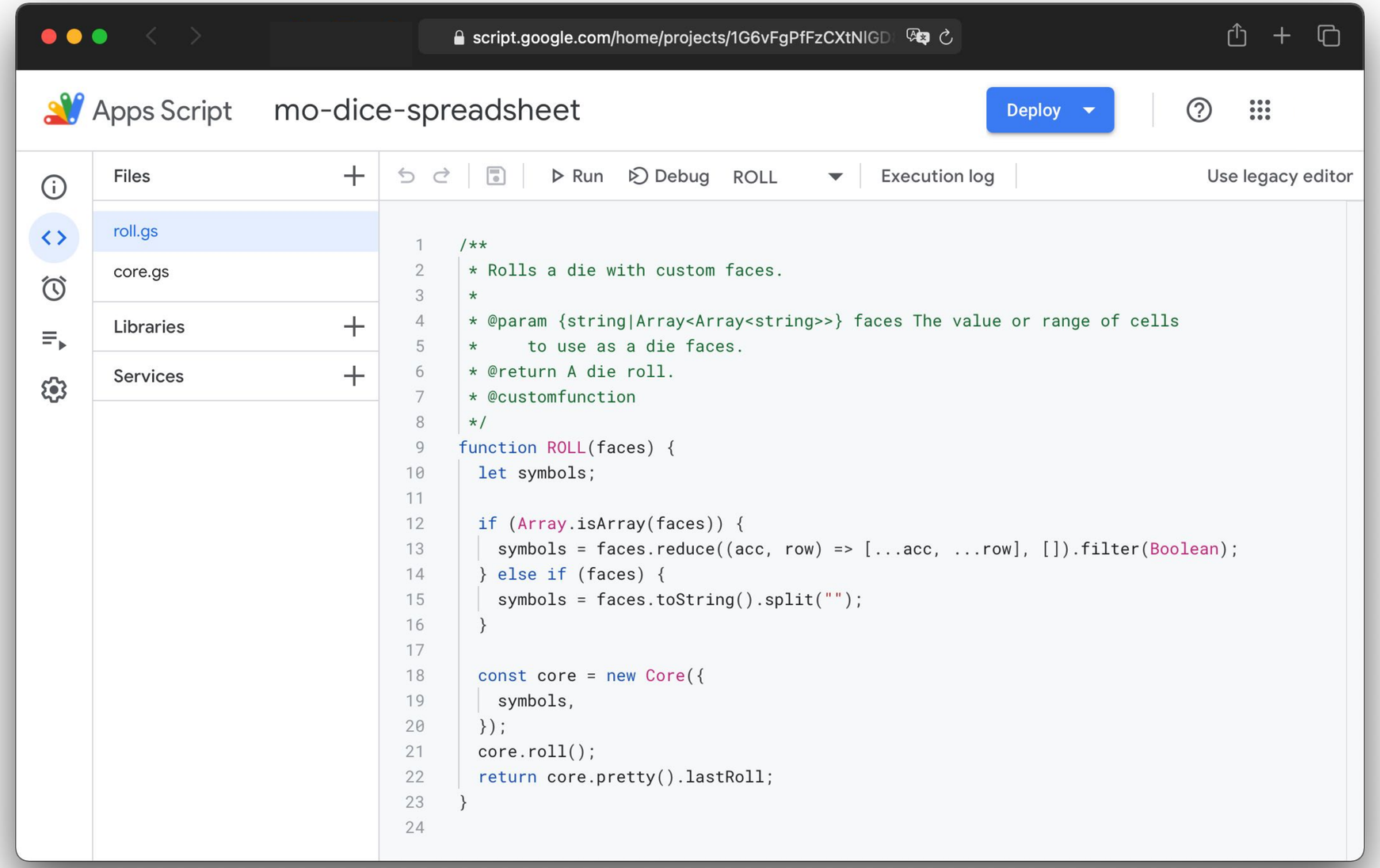

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
B1
=ABS(A1)
A B C D E F
1 -42 42
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Sheet1

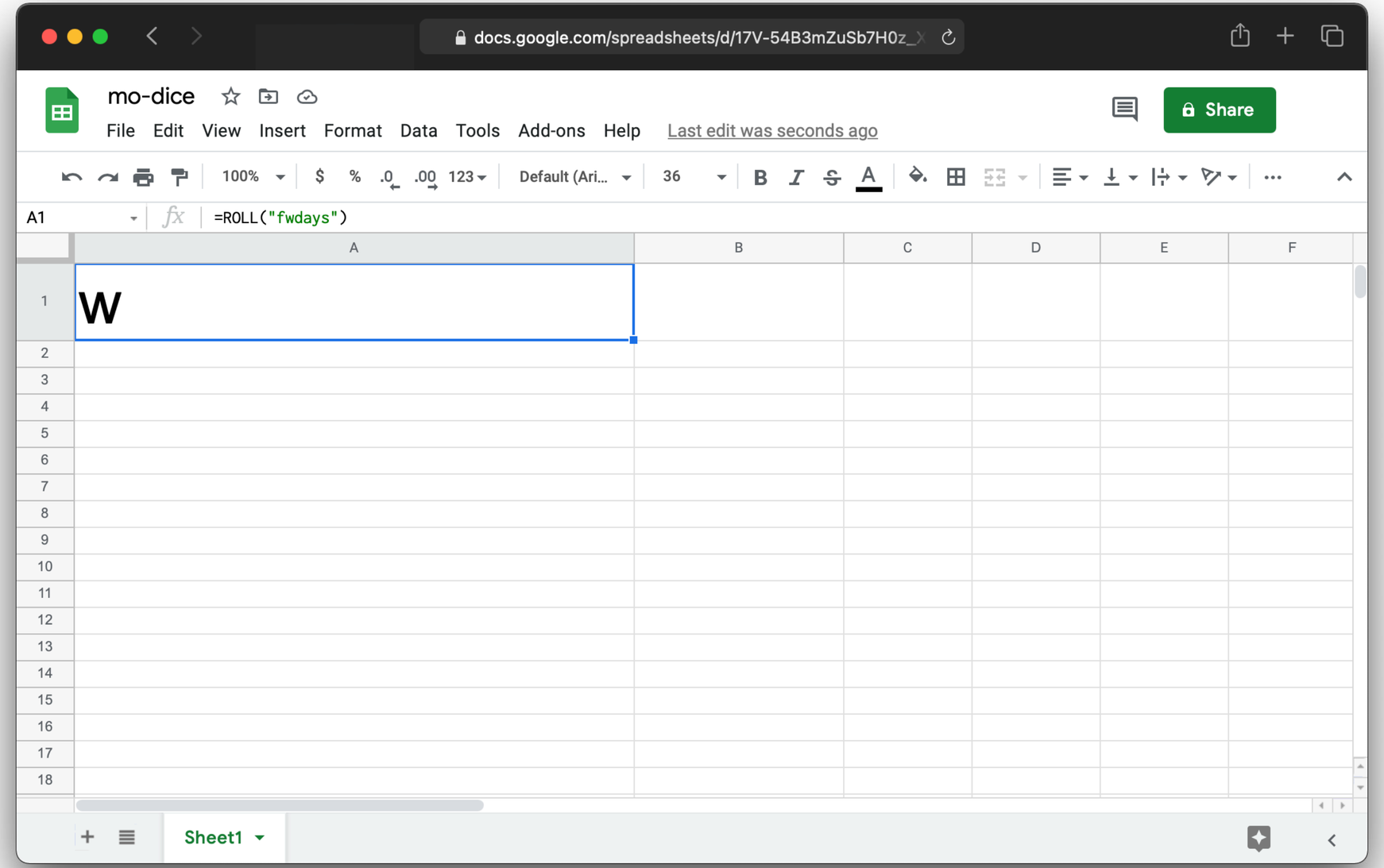

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
A1
=SUM(B2:C3)
A B C D E F
1 10
2 1 2
3 3 4
4 5 6 7 8 9 10 11 12 13 14
Sheet1

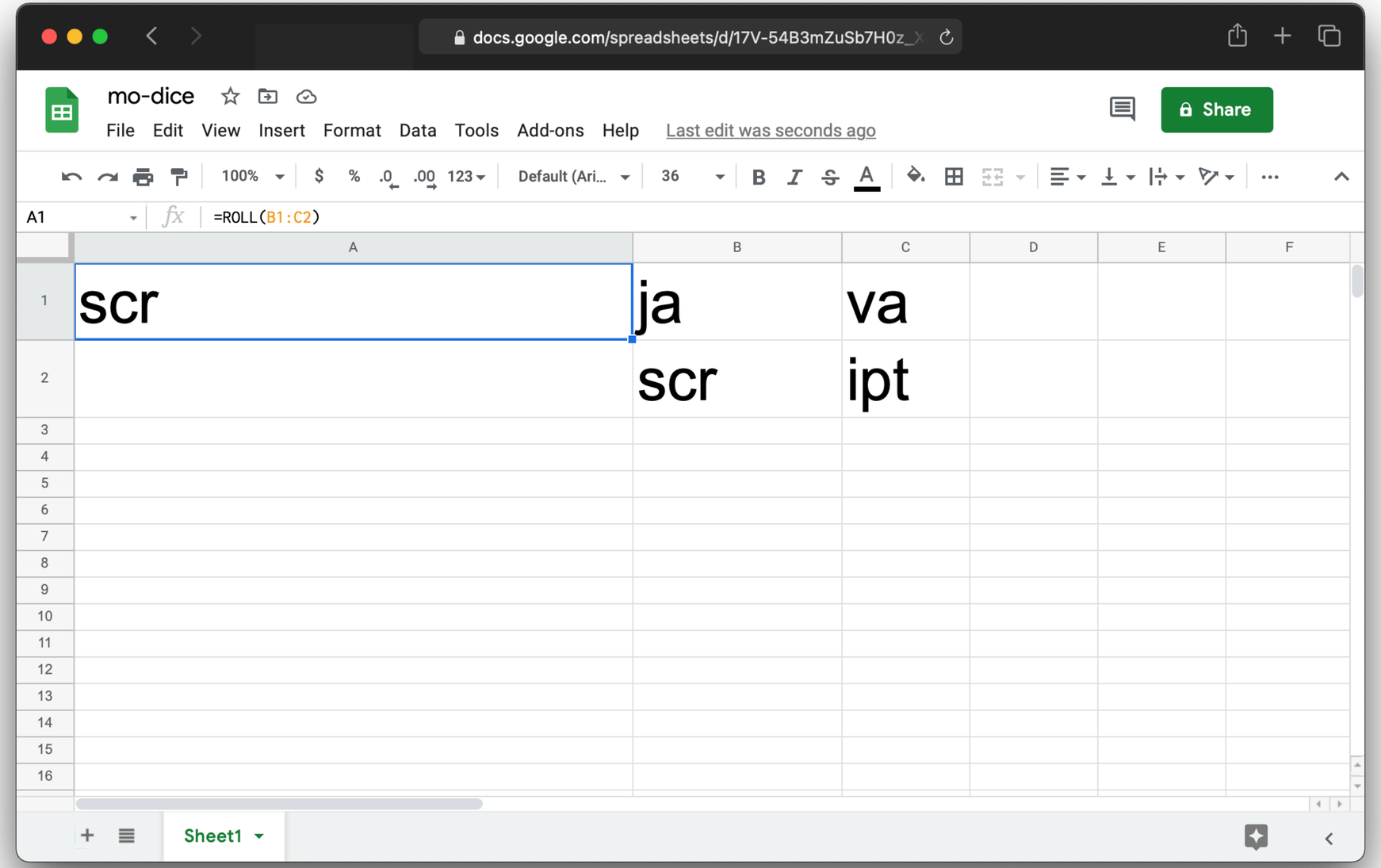

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100%
A1
Create a form
AppSheet
Script editor
Macros
Spelling
Enable autocomplete
Notification rules
Protect sheet
Accessibility settings
Sheet1

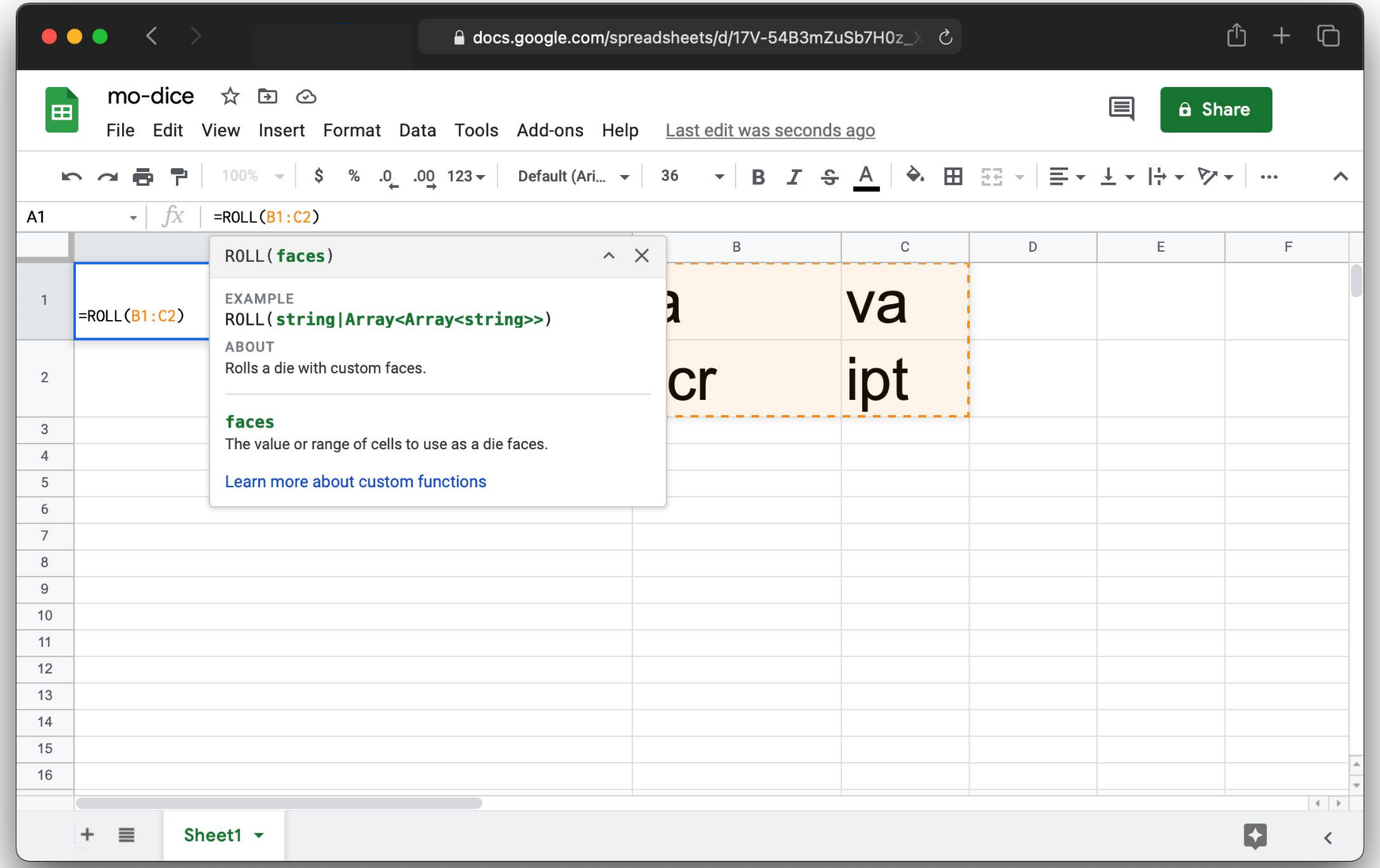

script.google.com/home/projects/1G6vFgPfFzCXtNlGD
Apps Script
mo-dice-spreadsheet
Deploy
Files
roll.gs
core.gs
Libraries
Services
Run
Debug
R
Execution log
Use legacy editor
const R = (n) => parseInt(Math.random() * n);
class Core {
constructor(options) {
options = options || {};
this.symbols = options.symbols || Core.symbols;
this.lastRoll =
typeof options.lastRoll === "number"
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
history: this.history.map((e) => this.symbols[e]),
};


script.google.com/home/projects/1G6vFgPfFzCXtNIGD
Apps Script
mo-dice-spreadsheet
Deploy
Files
roll.gs
core.gs
Libraries
Services
Run
Debug
R
Execution log
Use legacy editor
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
history: this.history.map((e) => this.symbols[e]),
};
}
clear() {
this.history = [];
}
}
Core.symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
export default Core;
Syntax error: SyntaxError: Unexpected token 'export' line: 35 file: core.gs
Copy

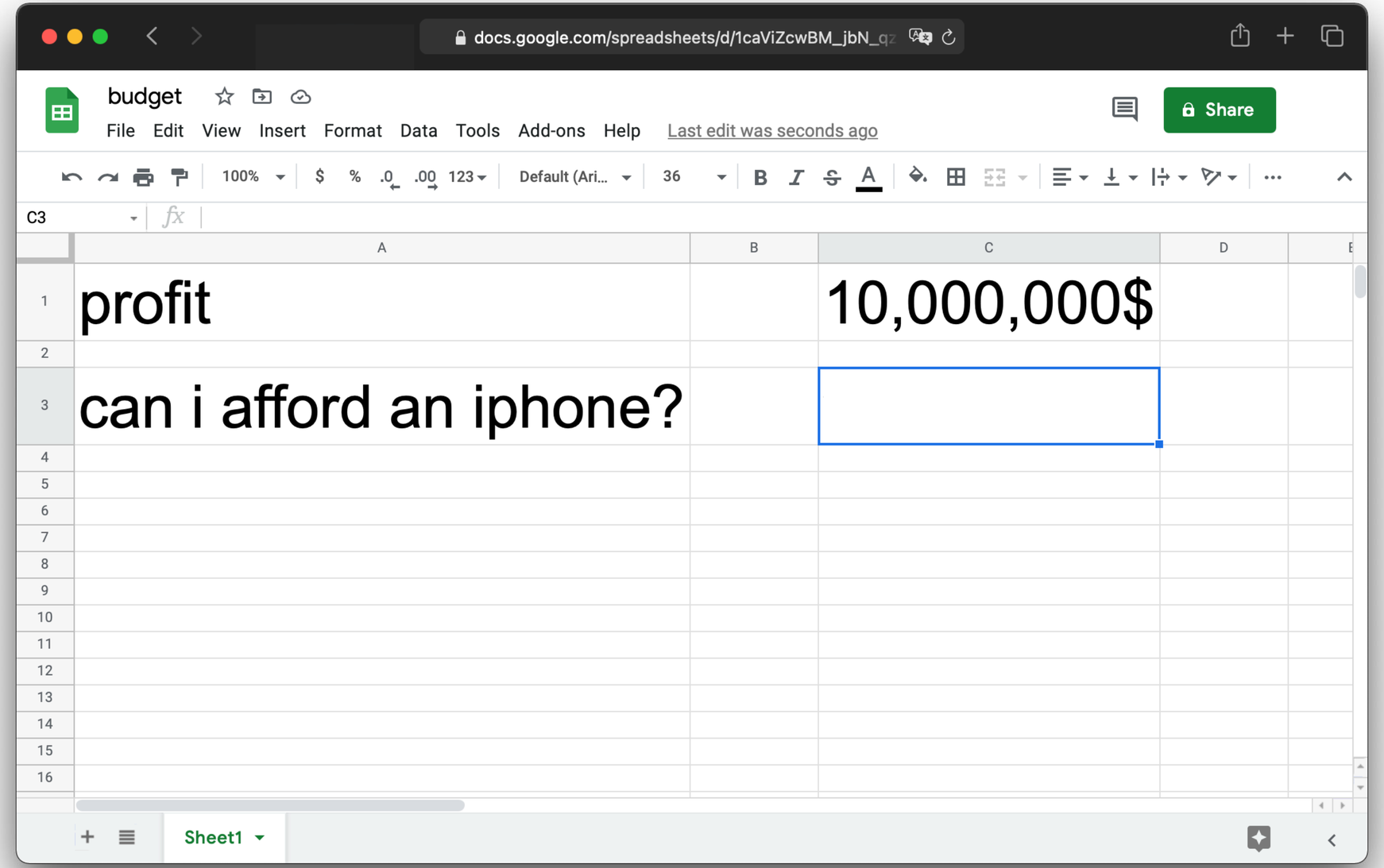

script.google.com/home/projects/1G6vFgPfFzCXtNlGD
Apps Script
mo-dice-spreadsheet
Deploy
Files
roll.gs
core.gs
Libraries
Services
Run
Debug
ROLL
Execution log
Use legacy editor
/**
* Rolls a die with custom faces.
*
* @param {string|Array<Array<string>>} faces The value or range of cells
*   to use as a die faces.
* @return A die roll.
* @customfunction
*/
function ROLL(faces) {
let symbols;
if (Array.isArray(faces)) {
symbols = faces.reduce((acc, row) => [...acc, ...row], []).filter(Boolean);
} else if (faces) {
symbols = faces.toString().split("");
}
const core = new Core({
symbols,
});
core.roll();
return core.pretty().lastRoll;
}

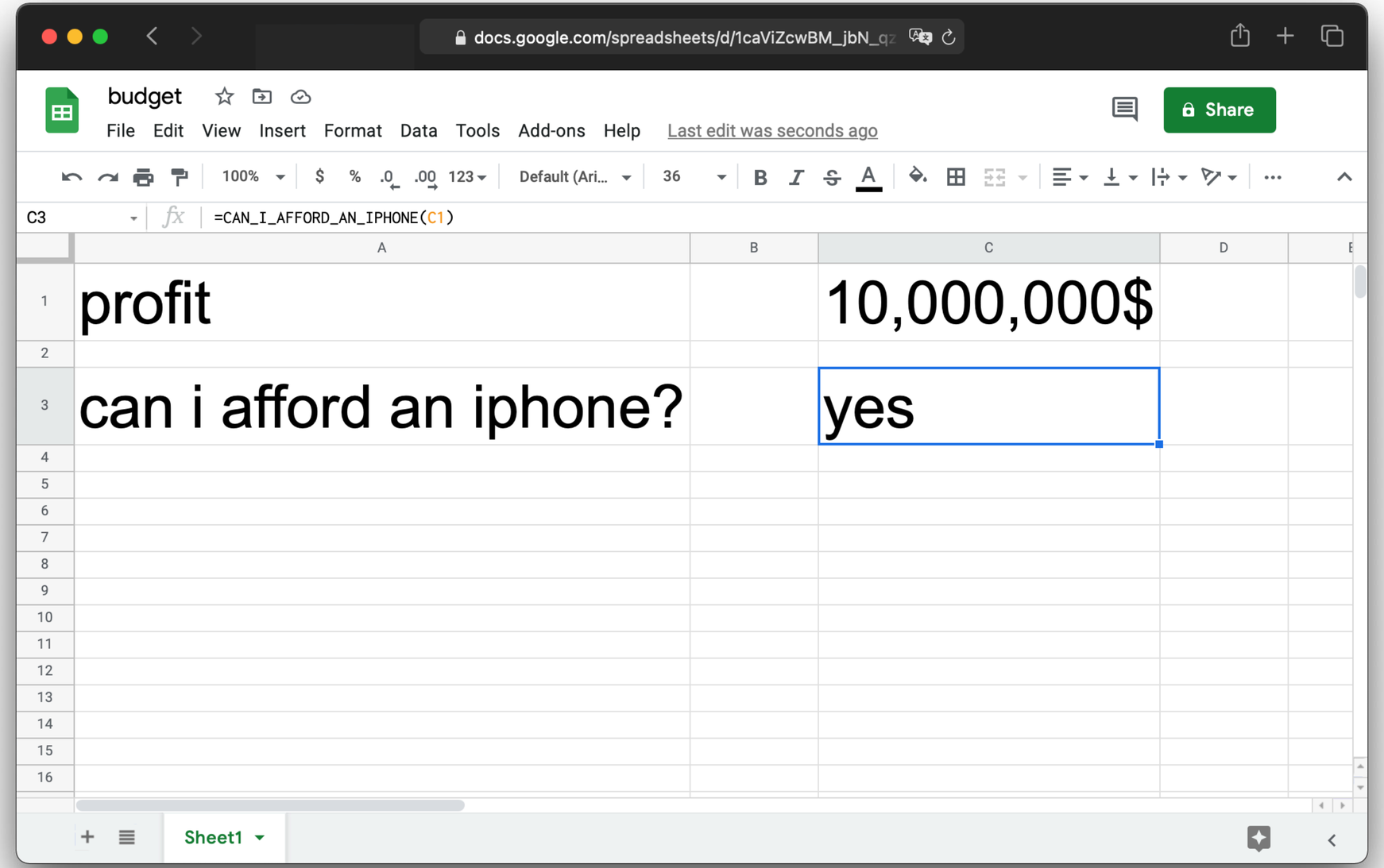

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
A1
=ROLL("fwdays")
A B C D E F
1 W
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Sheet1

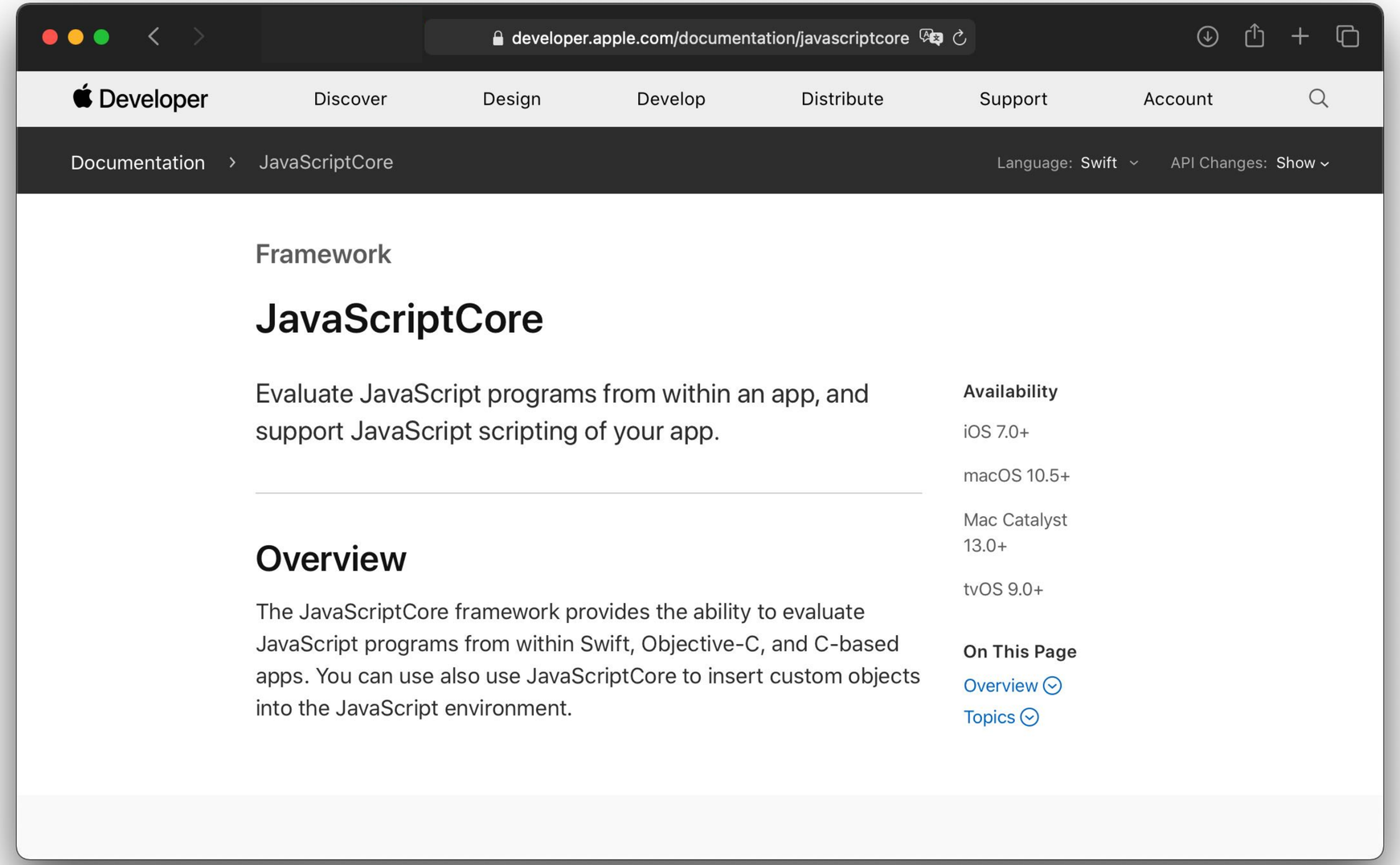

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
A1
=ROLL(B1:C2)
A B C D E F
1 scr ja va
2 scr ipt
3 4 5 6 7 8 9 10 11 12 13 14 15 16
Sheet1

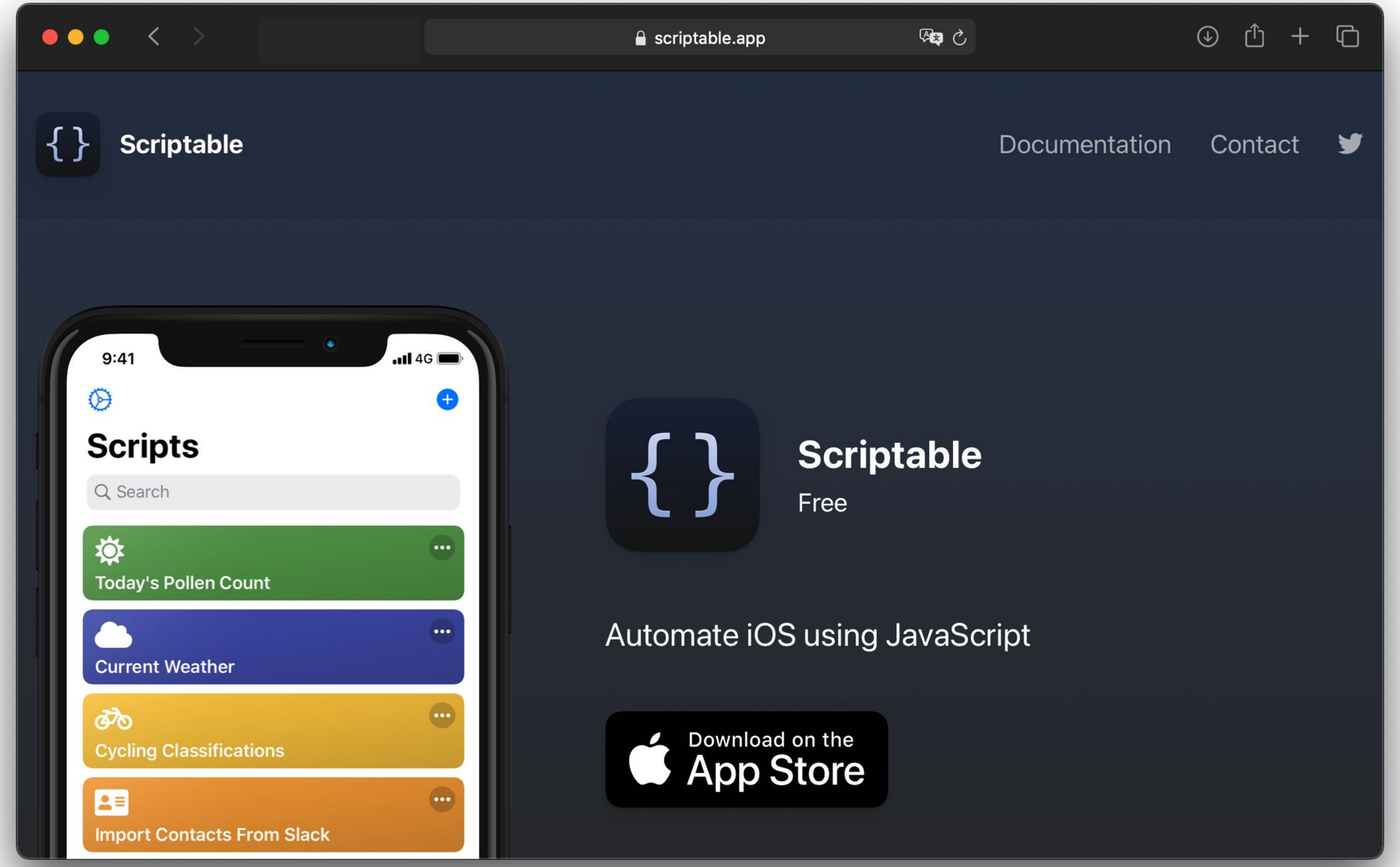

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z_
mo-dice
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
A1
=ROLL(B1:C2)
ROLL(faces)
EXAMPLE
ROLL(string|Array<Array<string>>)
ABOUT
Rolls a die with custom faces.
faces
The value or range of cells to use as a die faces.
Learn more about custom functions
=ROLL(B1:C2)
va
ipt
Sheet1

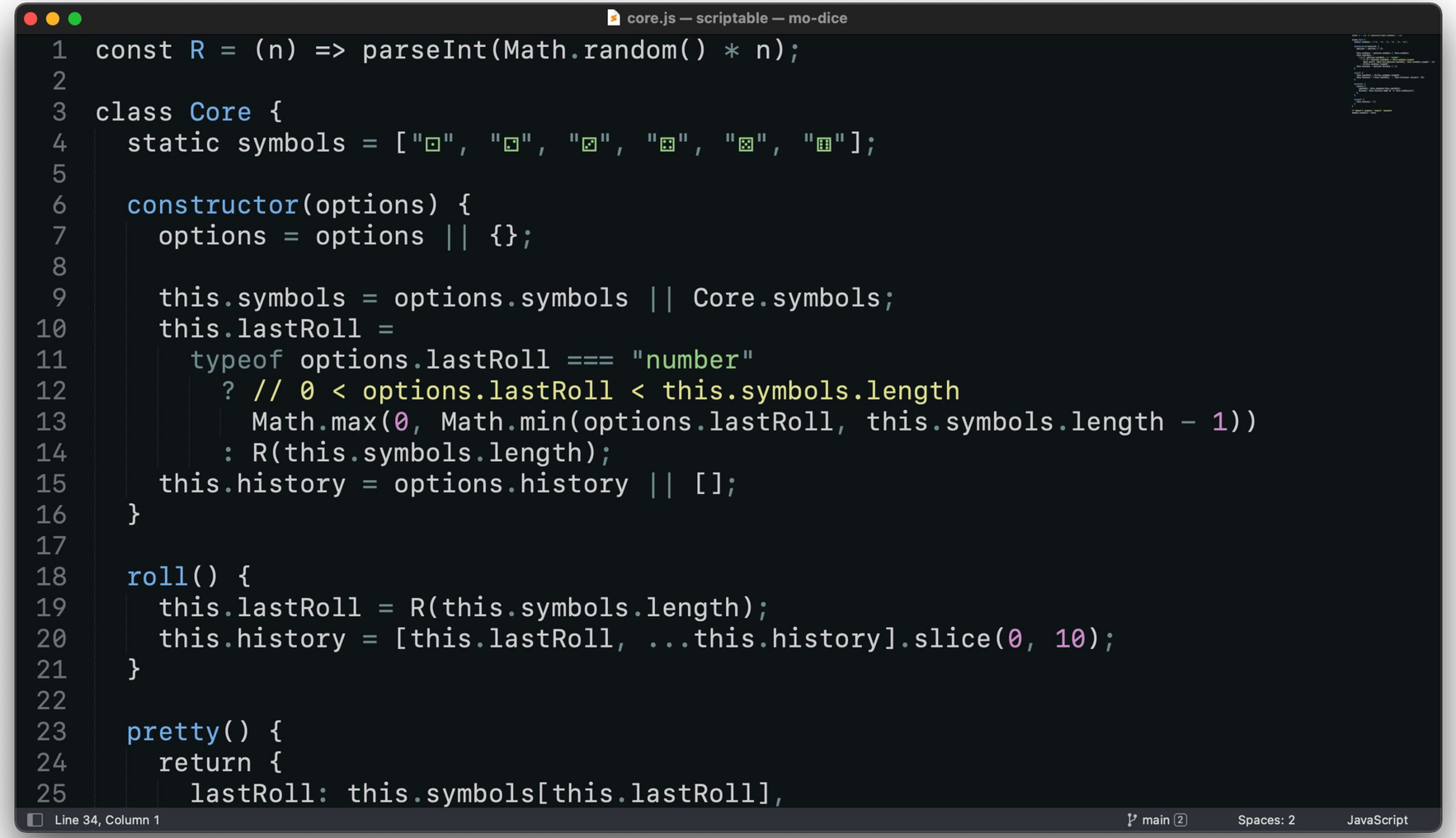
2018

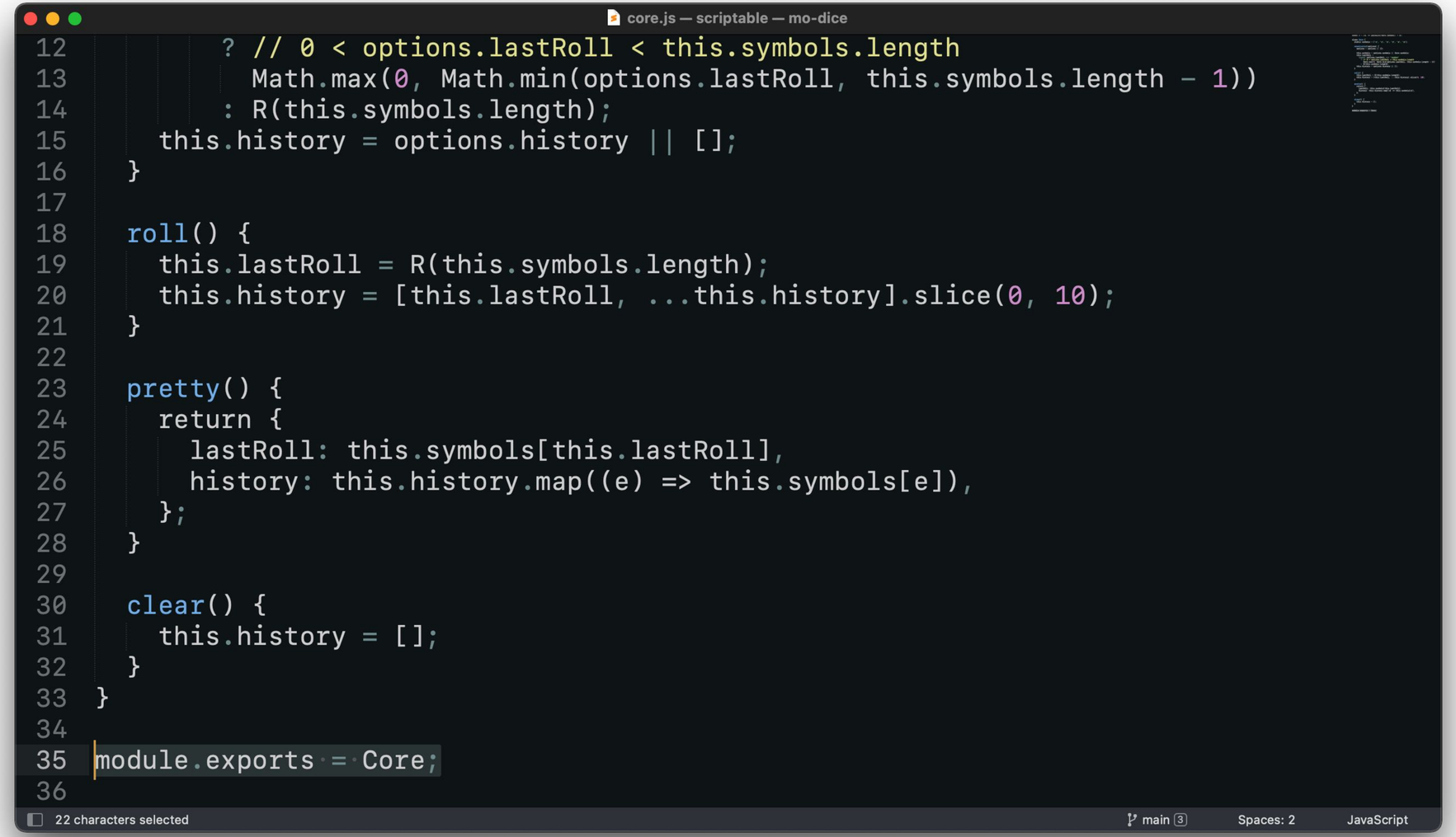

docs.google.com/spreadsheets/d/1caViZcwBM_jbN_qz
budget
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100%
Default (Ari...
36
C3
A
B
C
D
profit
10,000,000$
can i afford an iphone?
Sheet1

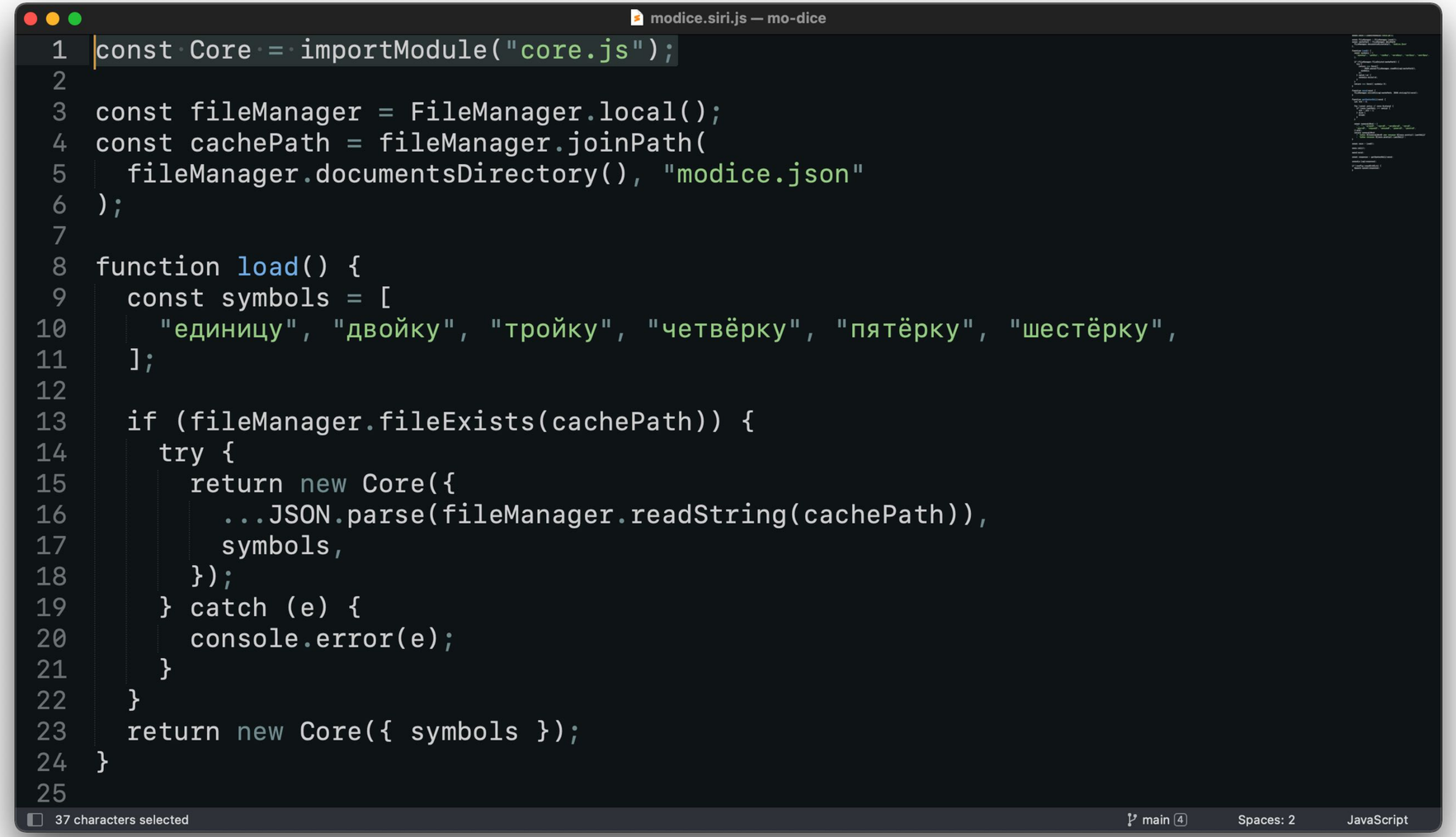

docs.google.com/spreadsheets/d/1caViZcwBM_jbN_qz
budget
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share
100% $ % .0 .00 123 Default (Ari... 36
C3
=CAN_I_AFFORD_AN_IPHONE(C1)
A B C D
1 profit 10,000,000$
2
3 can i afford an iphone? yes
4
5
6
7
8
9
10
11
12
13
14
15
16
Sheet1

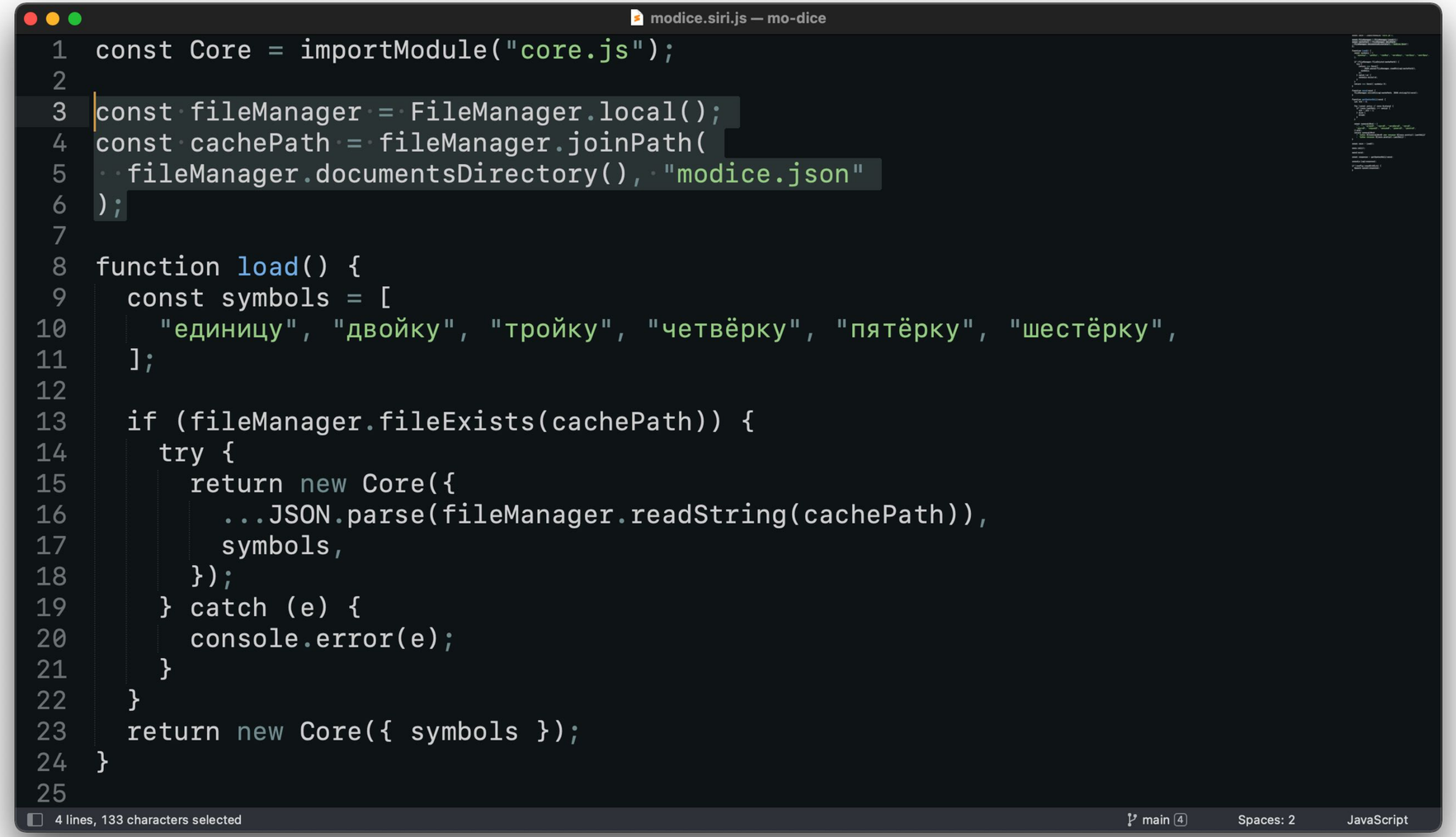

developer.apple.com/documentation/javascriptcore
Developer
Discover
Design
Develop
Distribute
Support
Account
Documentation
JavaScriptCore
Language: Swift
API Changes: Show
Framework
JavaScriptCore
Evaluate JavaScript programs from within an app, and support JavaScript scripting of your app.
Availability
iOS 7.0+
macOS 10.5+
Mac Catalyst 13.0+
tvOS 9.0+
On This Page
Overview
Topics
Overview
The JavaScriptCore framework provides the ability to evaluate JavaScript programs from within Swift, Objective-C, and C-based apps. You can use also use JavaScriptCore to insert custom objects into the JavaScript environment.

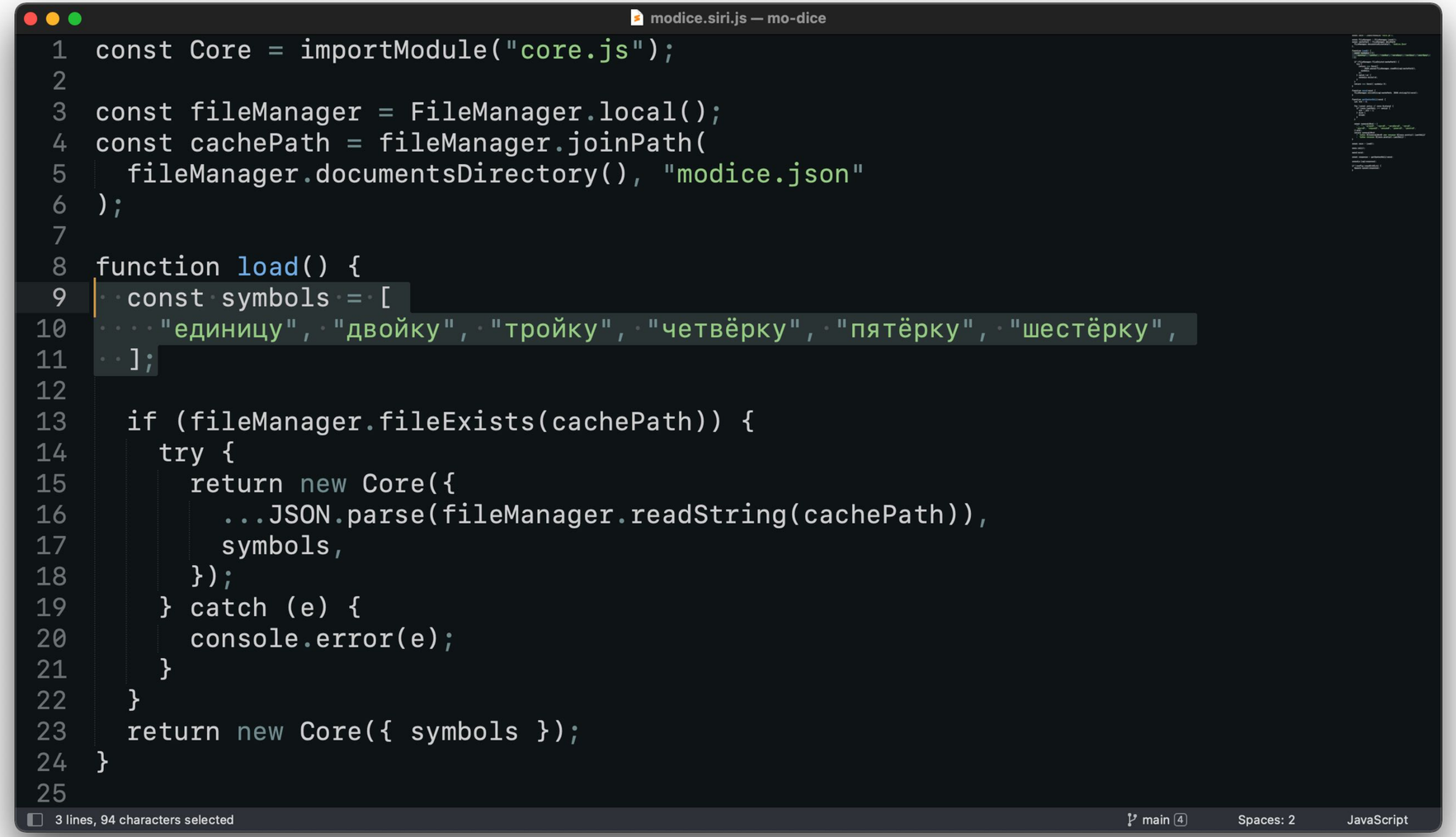

scriptable.app
Scriptable
Documentation
Contact
9:41
4G
Scripts
Search
Today's Pollen Count
Current Weather
Cycling Classifications
Import Contacts From Slack
Scriptable
Free
Automate iOS using JavaScript
Download on the
App Store


core.js — scriptable — mo-dice
const R = (n) => parseInt(Math.random() * n);
class Core {
static symbols = ["⚀", "⚁", "⚂", "⚃", "⚄", "⚅"];
constructor(options) {
options = options || {};
this.symbols = options.symbols || Core.symbols;
this.lastRoll =
typeof options.lastRoll === "number"
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
Line 34, Column 1
main 2
Spaces: 2
JavaScript


core.js — scriptable — mo-dice
? // 0 < options.lastRoll < this.symbols.length
Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
: R(this.symbols.length);
this.history = options.history || [];
}
roll() {
this.lastRoll = R(this.symbols.length);
this.history = [this.lastRoll, ...this.history].slice(0, 10);
}
pretty() {
return {
lastRoll: this.symbols[this.lastRoll],
history: this.history.map((e) => this.symbols[e]),
};
}
clear() {
this.history = [];
}
}
module.exports = Core;
22 characters selected
main 3
Spaces: 2
JavaScript


modice.siri.js — mo-dice
const Core = importModule("core.js");
const fileManager = FileManager.local();
const cachePath = fileManager.joinPath(
fileManager.documentsDirectory(), "modice.json"
);
function load() {
const symbols = [
"единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
];
if (fileManager.fileExists(cachePath)) {
try {
return new Core({
...JSON.parse(fileManager.readString(cachePath)),
symbols,
});
} catch (e) {
console.error(e);
}
}
return new Core({ symbols });
}
37 characters selected
main 4
Spaces: 2
JavaScript

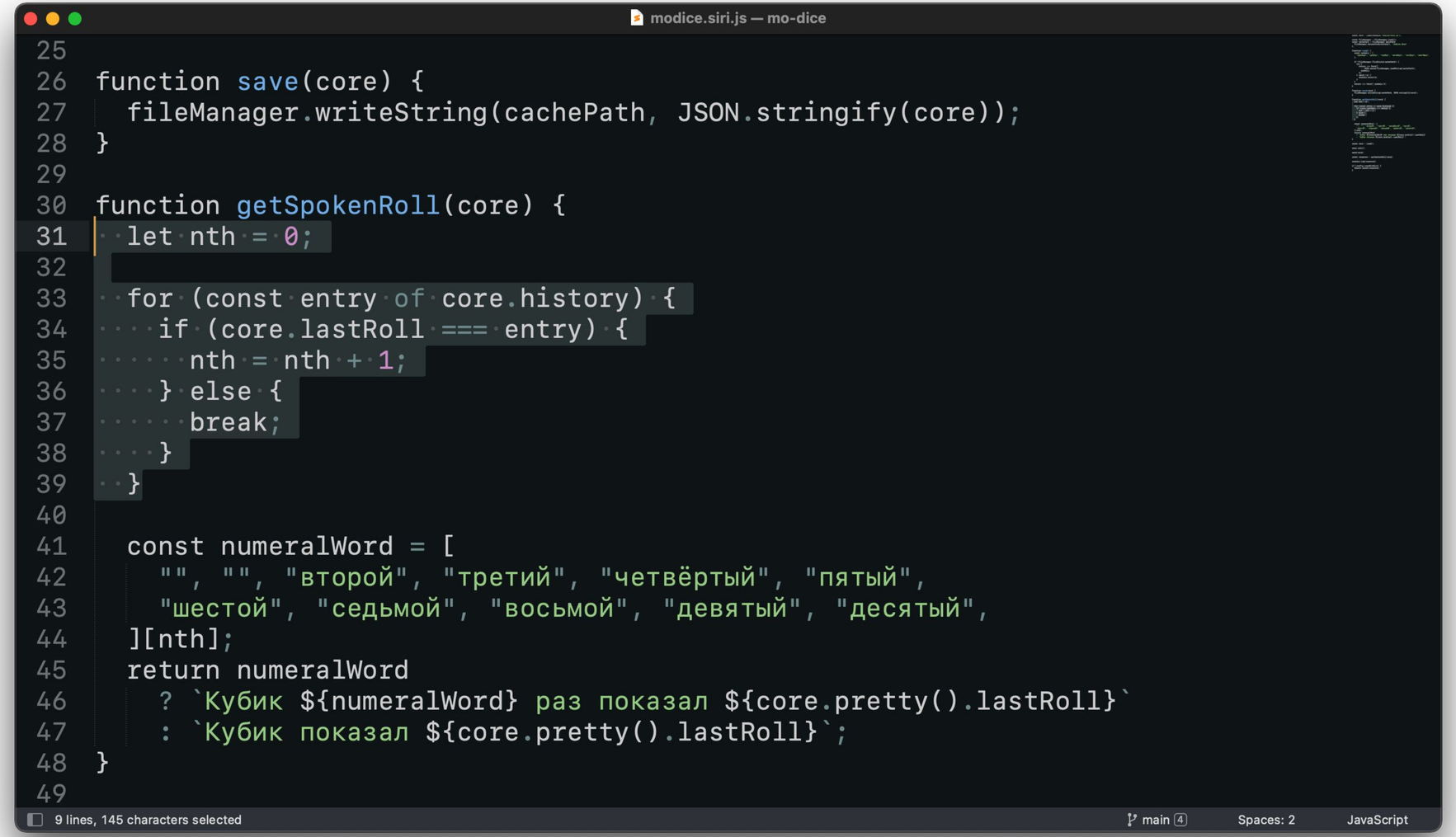

modice.siri.js — mo-dice
const Core = importModule("core.js");
const fileManager = FileManager.local();
const cachePath = fileManager.joinPath(
fileManager.documentsDirectory(), "modice.json"
);
function load() {
const symbols = [
"единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
];
if (fileManager.fileExists(cachePath)) {
try {
return new Core({
...JSON.parse(fileManager.readString(cachePath)),
symbols,
});
} catch (e) {
console.error(e);
}
}
return new Core({ symbols });
}
4 lines, 133 characters selected
main 4
Spaces: 2
JavaScript

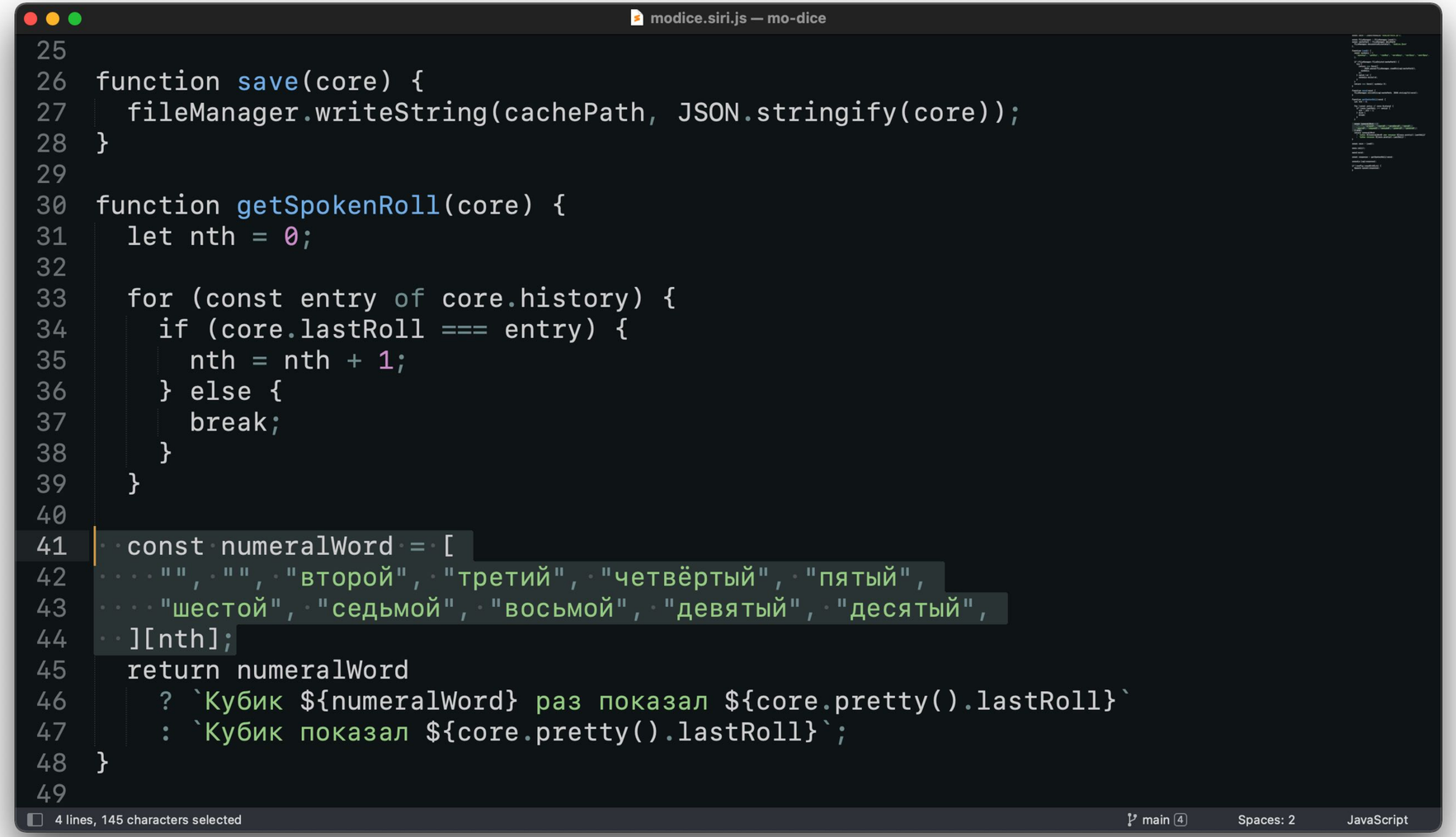

modice.siri.js — mo-dice
const Core = importModule("core.js");
const fileManager = FileManager.local();
const cachePath = fileManager.joinPath(
fileManager.documentsDirectory(), "modice.json"
);
function load() {
const symbols = [
"единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
];
if (fileManager.fileExists(cachePath)) {
try {
return new Core({
...JSON.parse(fileManager.readString(cachePath)),
symbols,
});
} catch (e) {
console.error(e);
}
}
return new Core({ symbols });
}
3 lines, 94 characters selected
main 4
Spaces: 2
JavaScript

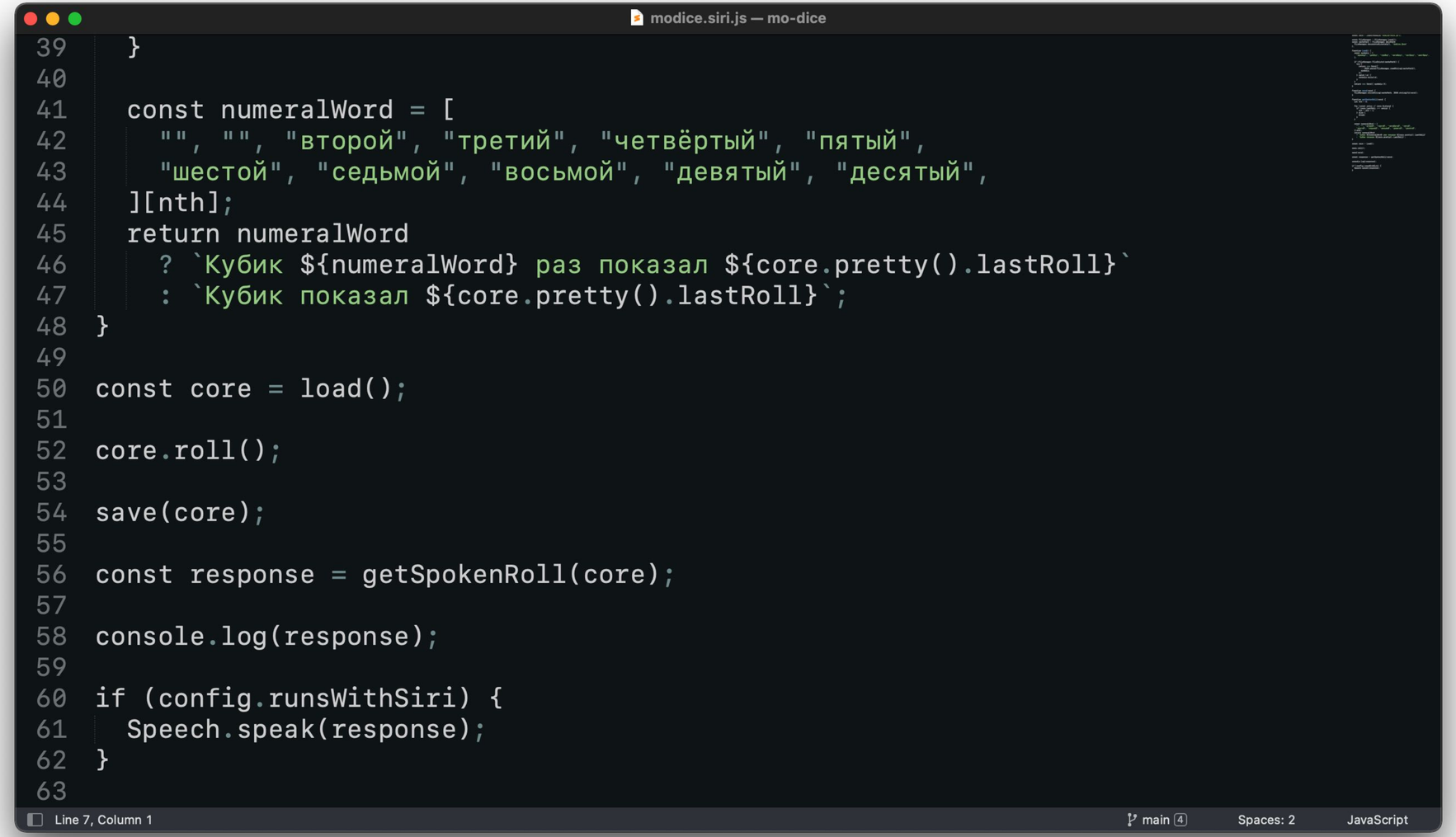
modice.siri.js — mo-dice

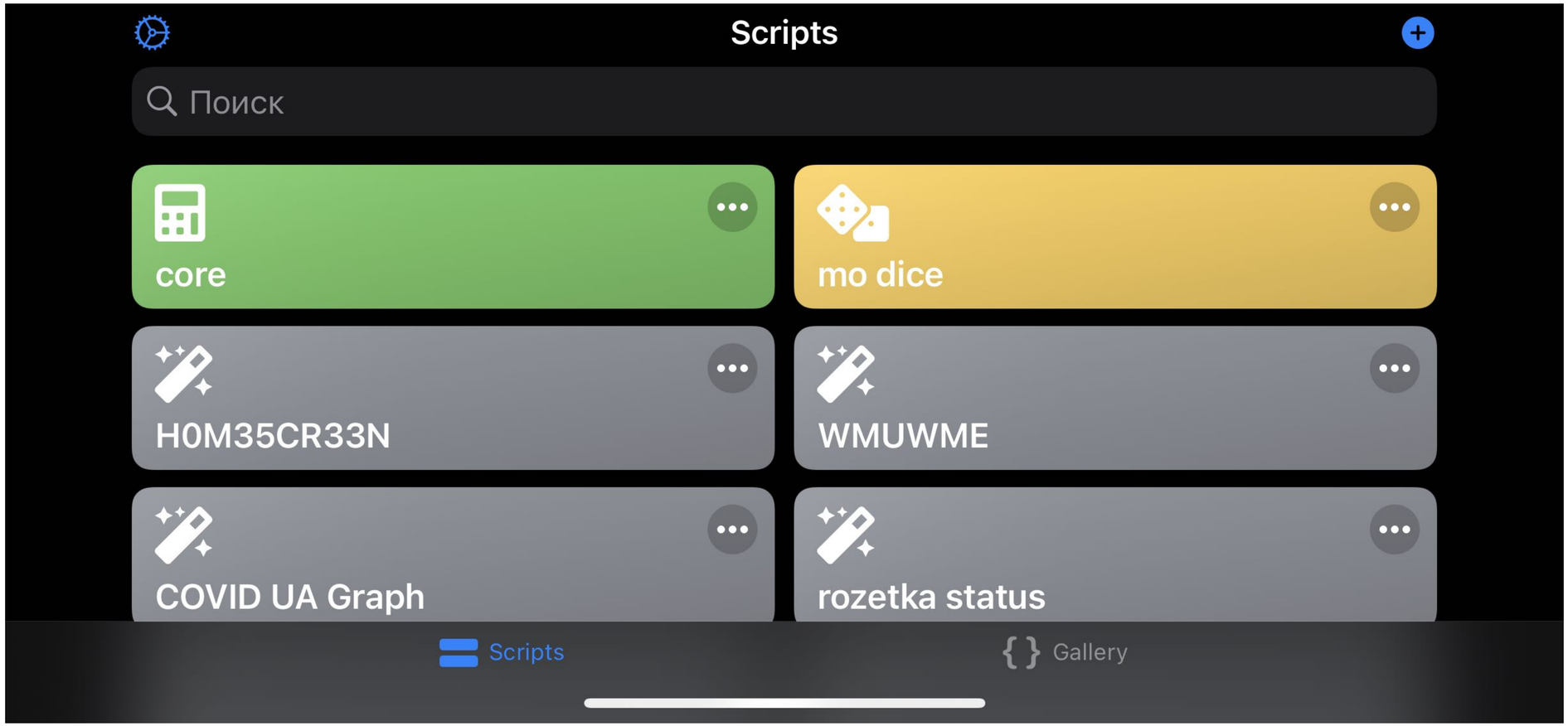
```
25
26  function save(core) {
27    fileManager.writeString(cachePath, JSON.stringify(core));
28  }
29
30  function getSpokenRoll(core) {
31    let nth = 0;
32
33    for (const entry of core.history) {
34      if (core.lastRoll === entry) {
35        nth = nth + 1;
36      } else {
37        break;
38      }
39    }
40
41    const numeralWord = [
42      "", "", "второй", "третий", "четвёртый", "пятый",
43      "шестой", "седьмой", "восьмой", "девятый", "десятый",
44    ][nth];
45    return numeralWord
46      ? `Кубик ${numeralWord} раз показал ${core.pretty().lastRoll}`
47      : `Кубик показал ${core.pretty().lastRoll}`;
48  }
49
```

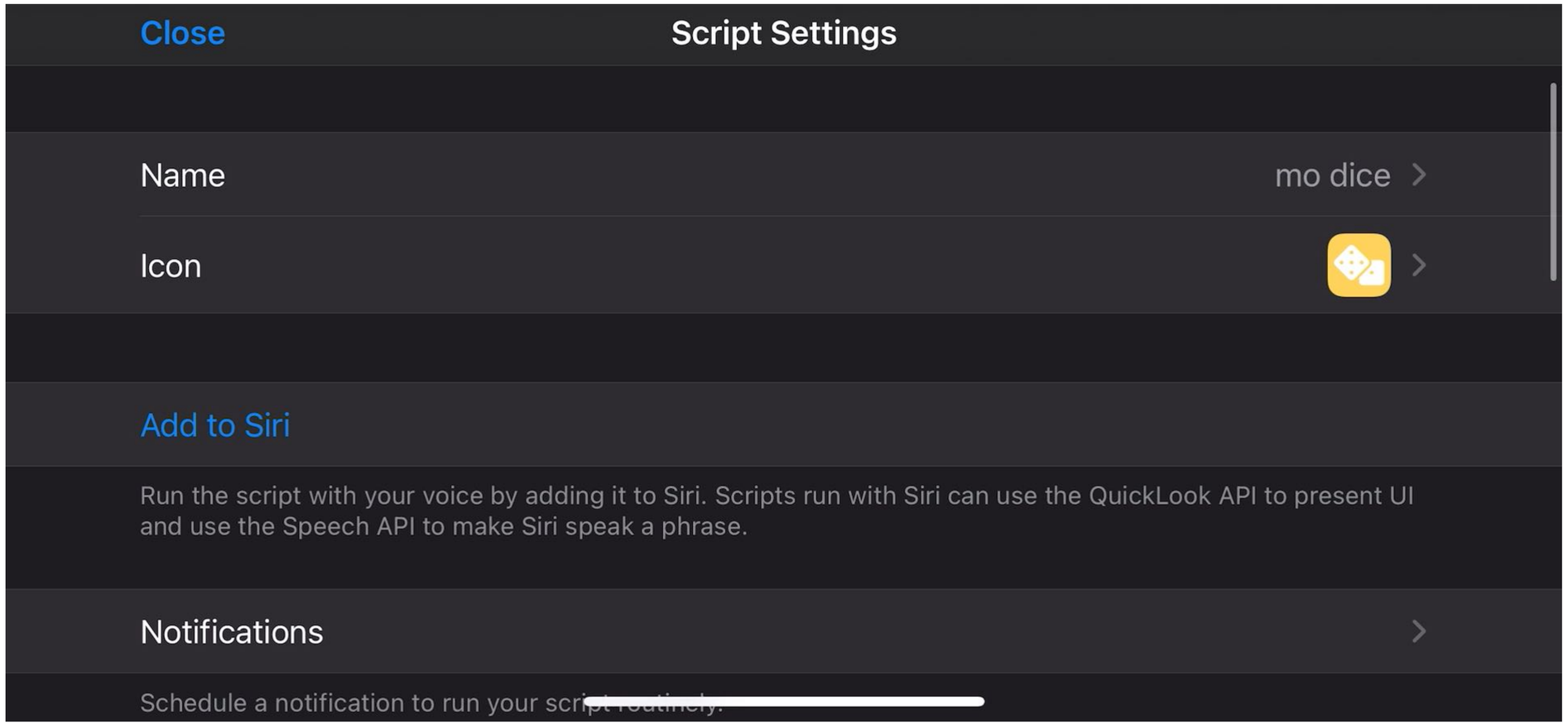
Line 1, Column 1 main 4 Spaces: 2 JavaScript

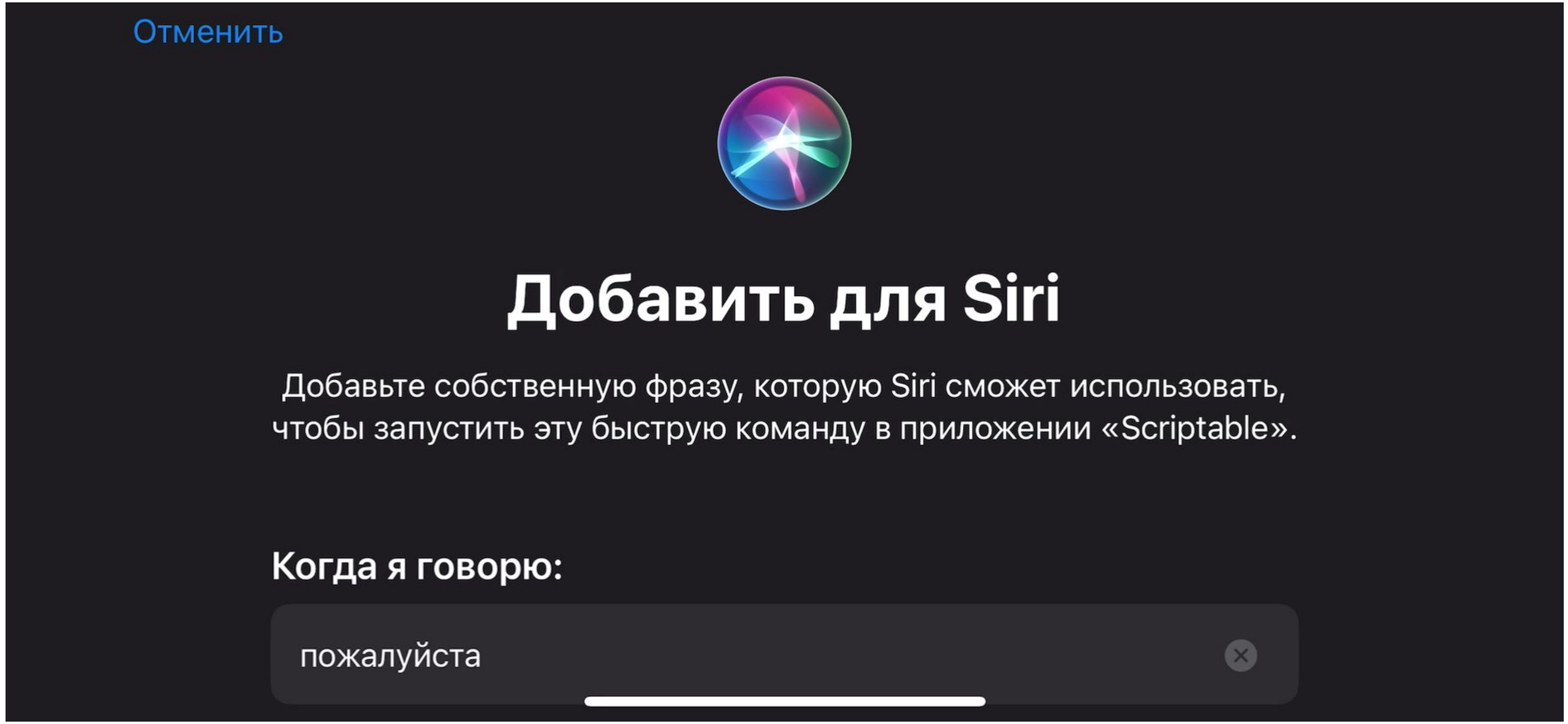

modice.siri.js — mo-dice
function save(core) {
fileManager.writeString(cachePath, JSON.stringify(core));
}
function getSpokenRoll(core) {
let nth = 0;
for (const entry of core.history) {
if (core.lastRoll === entry) {
nth = nth + 1;
} else {
break;
}
}
const numeralWord = [
"", "", "второй", "третий", "четвёртый", "пятый",
"шестой", "седьмой", "восьмой", "девятый", "десятый",
][nth];
return numeralWord
? `Кубик ${numeralWord} раз показал ${core.pretty().lastRoll}`
: `Кубик показал ${core.pretty().lastRoll}`;
}
9 lines, 145 characters selected
main 4
Spaces: 2
JavaScript

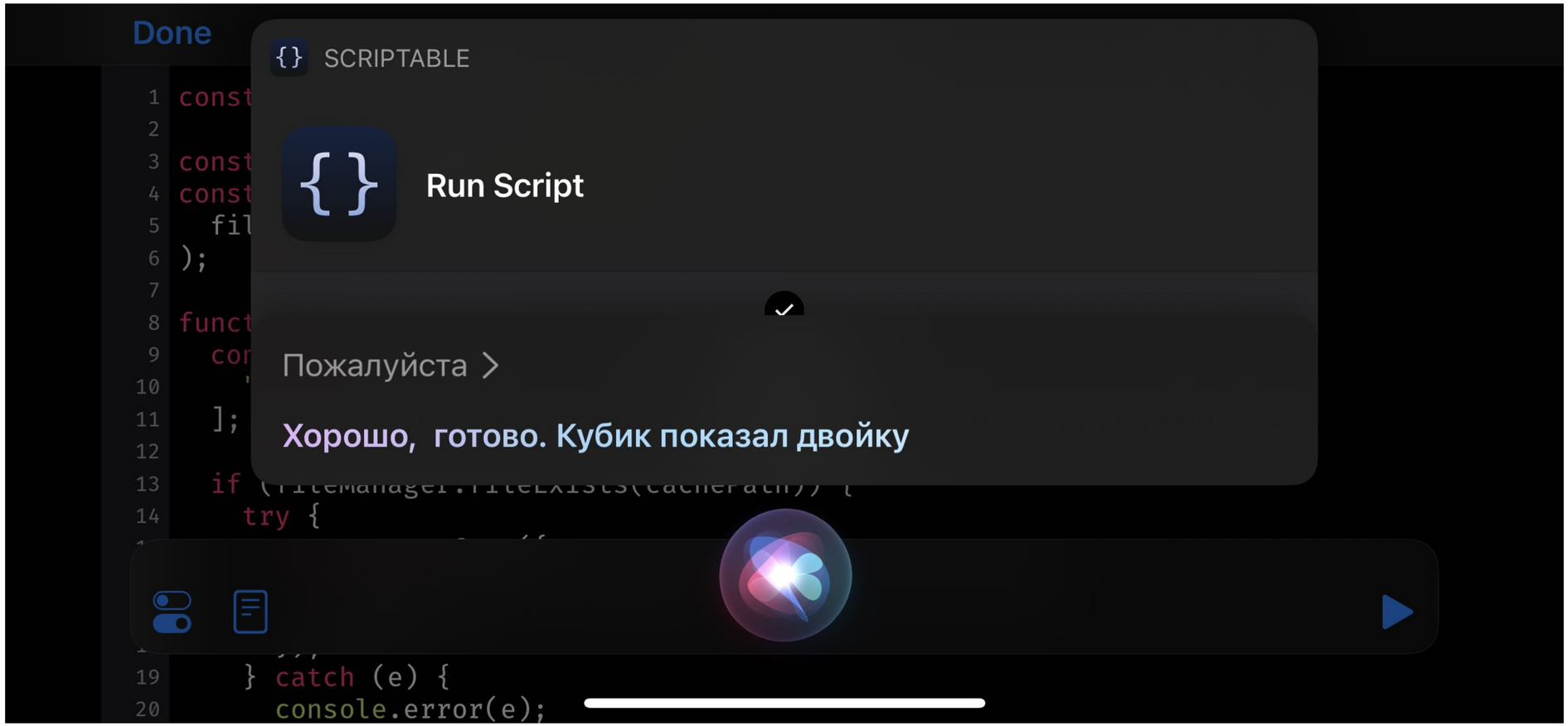

modice.siri.js — mo-dice
function save(core) {
fileManager.writeString(cachePath, JSON.stringify(core));
}
function getSpokenRoll(core) {
let nth = 0;
for (const entry of core.history) {
if (core.lastRoll === entry) {
nth = nth + 1;
} else {
break;
}
}
const numeralWord = [
"", "", "второй", "третий", "четвёртый", "пятый",
"шестой", "седьмой", "восьмой", "девятый", "десятый",
][nth];
return numeralWord
? `Кубик ${numeralWord} раз показал ${core.pretty().lastRoll}`
: `Кубик показал ${core.pretty().lastRoll}`;
}
4 lines, 145 characters selected
main 4
Spaces: 2
JavaScript

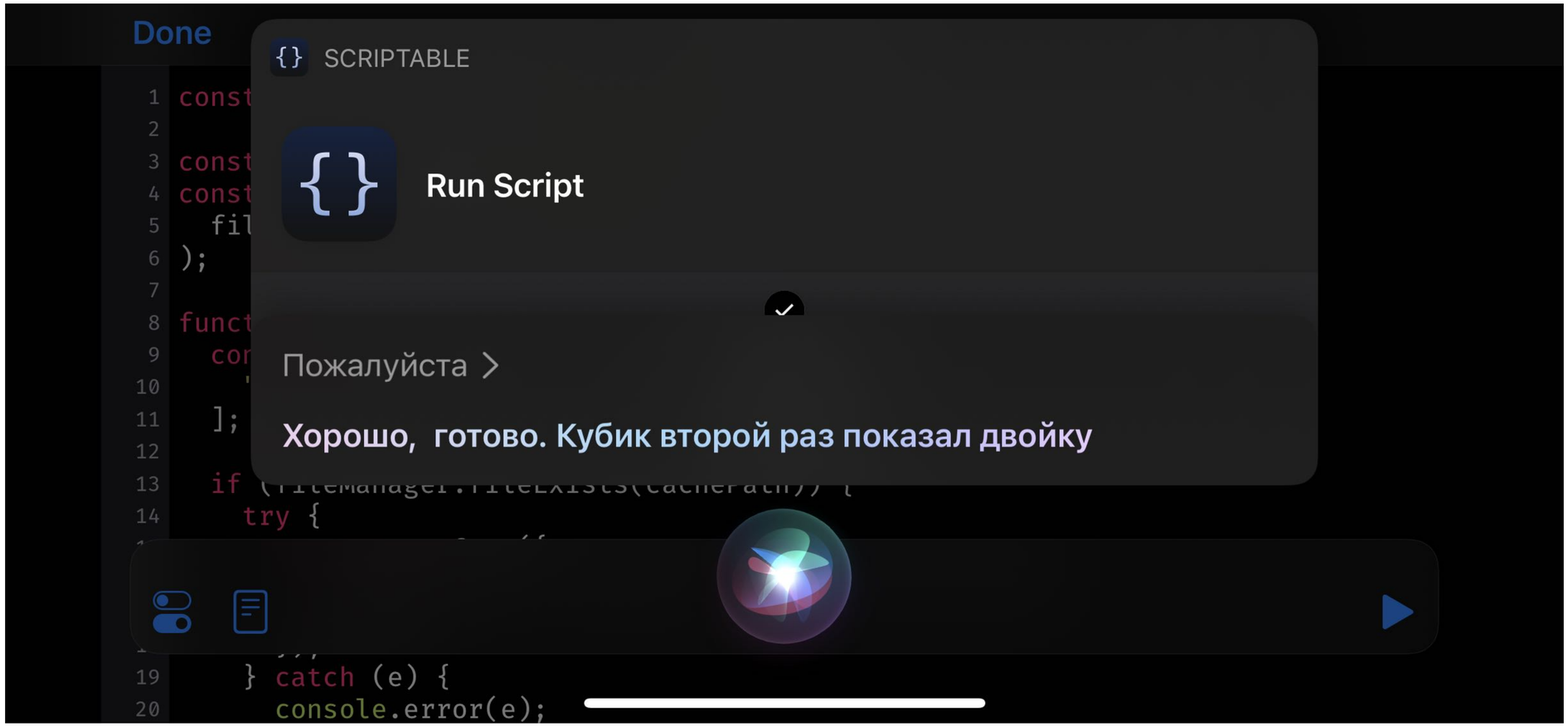

modice.siri.js — mo-dice
39 }
40
41 const numeralWord = [
42 "", "", "второй", "третий", "четвёртый", "пятый",
43 "шестой", "седьмой", "восьмой", "девятый", "десятый",
44][nth];
45 return numeralWord
46 ? `Кубик ${numeralWord} раз показал ${core.pretty().lastRoll}`
47 : `Кубик показал ${core.pretty().lastRoll}`;
48 }
49
50 const core = load();
51
52 core.roll();
53
54 save(core);
55
56 const response = getSpokenRoll(core);
57
58 console.log(response);
59
60 if (config.runsWithSiri) {
61 Speech.speak(response);
62 }
63
Line 7, Column 1
main 4
Spaces: 2
JavaScript


Scripts
Поиск
core
mo dice
H0M35CR33N
WMUWME
COVID UA Graph
rozetka status
Scripts
Gallery

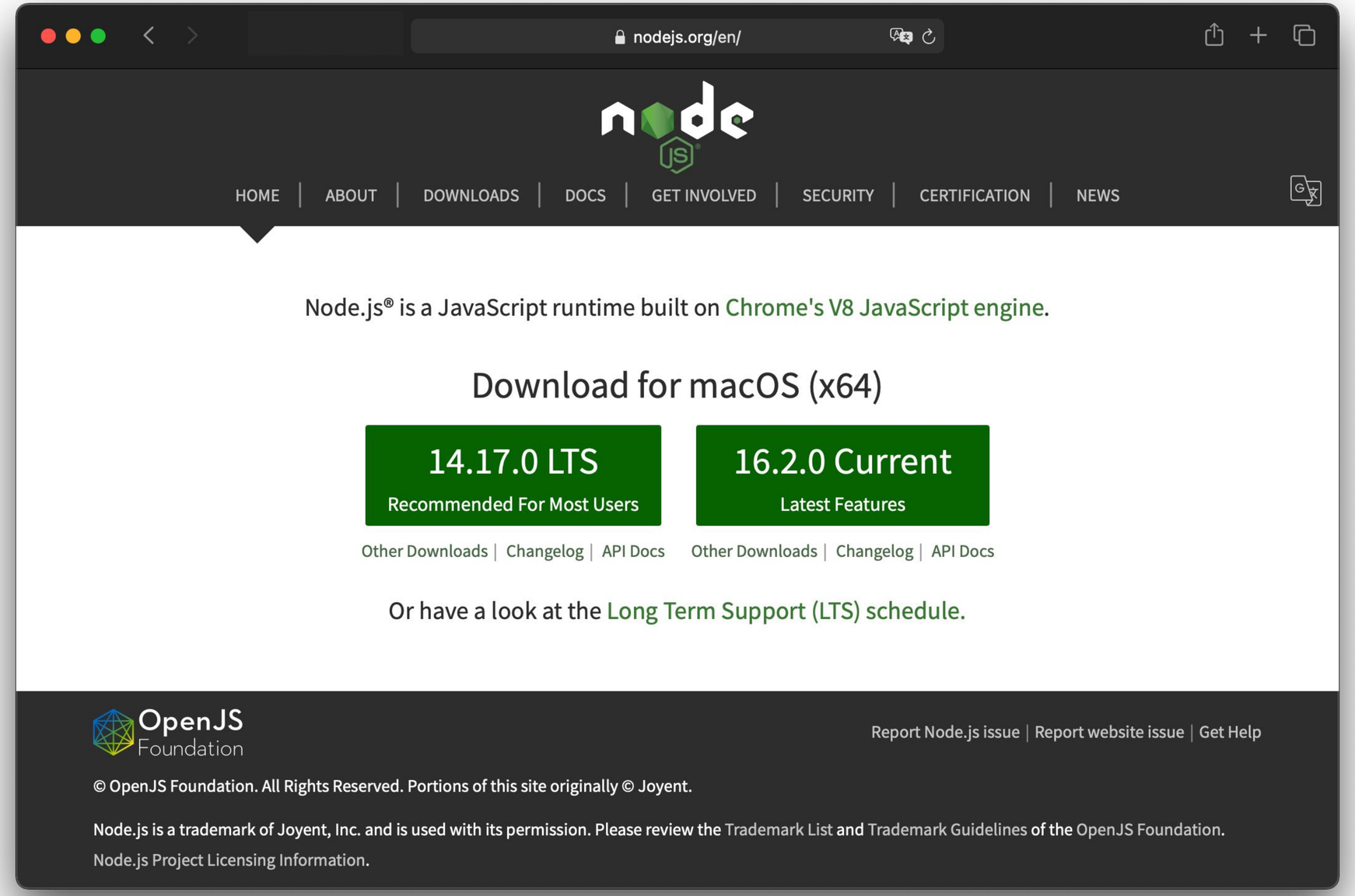

Close
Script Settings
Name
mo dice
Icon
Add to Siri
Run the script with your voice by adding it to Siri. Scripts run with Siri can use the QuickLook API to present UI and use the Speech API to make Siri speak a phrase.
Notifications
Schedule a notification to run your script routinely.

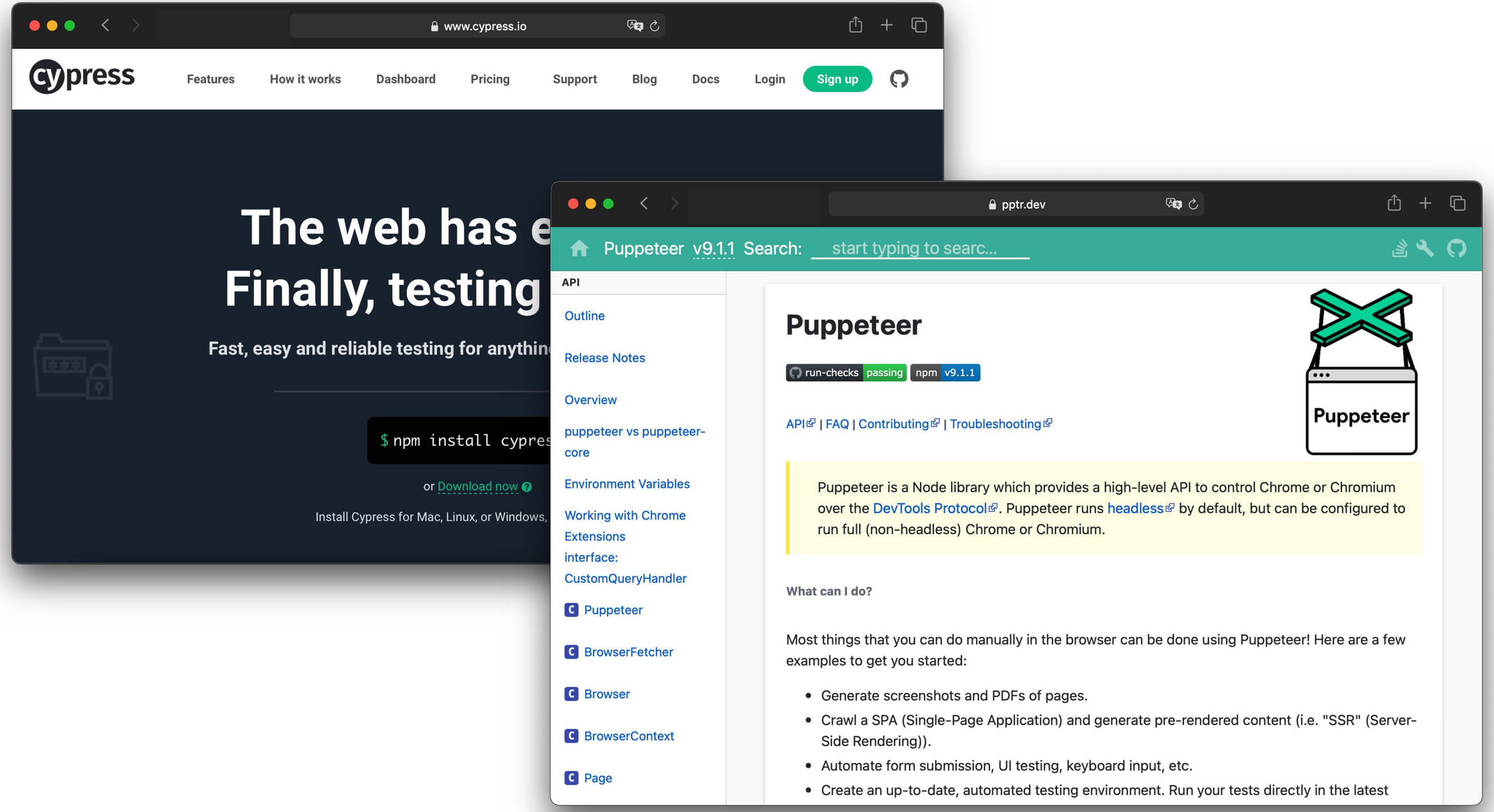

Отменить
Добавить для Siri
Добавьте собственную фразу, которую Siri сможет использовать, чтобы запустить эту быструю команду в приложении «Scriptable».
Когда я говорю:
пожалуйста


Done
SCRIPTABLE
Run Script
Пожалуйста
Хорошо, готово. Кубик показал двойку


Done
SCRIPTABLE
Run Script
Пожалуйста
Хорошо, готово. Кубик второй раз показал двойку

everywhere


nodejs.org/en/
node
HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS
Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.
Download for macOS (x64)
14.17.0 LTS
Recommended For Most Users
16.2.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs
Or have a look at the Long Term Support (LTS) schedule.
OpenJS Foundation
Report Node.js issue | Report website issue | Get Help
© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.
Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark List and Trademark Guidelines of the OpenJS Foundation.
Node.js Project Licensing Information.


www.cypress.io
cypress
Features
How it works
Dashboard
Pricing
Support
Blog
Docs
Login
Sign up
The web has e
Finally, testing
Fast, easy and reliable testing for anythin
$ npm install cypres
or Download now
Install Cypress for Mac, Linux, or Windows,
pptr.dev
Puppeteer v9.1.1 Search:
start typing to searc...
API
Outline
Release Notes
Overview
puppeteer vs puppeteer-core
Environment Variables
Working with Chrome Extensions
interface: CustomQueryHandler
Puppeteer
BrowserFetcher
Browser
BrowserContext
Page
Puppeteer
run-checks passing
npm v9.1.1
API | FAQ | Contributing | Troubleshooting
Puppeteer
Puppeteer is a Node library which provides a high-level API to control Chrome or Chromium over the DevTools Protocol. Puppeteer runs headless by default, but can be configured to run full (non-headless) Chrome or Chromium.
What can I do?
Most things that you can do manually in the browser can be done using Puppeteer! Here are a few examples to get you started:
Generate screenshots and PDFs of pages.
Crawl a SPA (Single-Page Application) and generate pre-rendered content (i.e. "SSR" (Server-Side Rendering)).
Automate form submission, UI testing, keyboard input, etc.
Create an up-to-date, automated testing environment. Run your tests directly in the latest

addons.mozilla.org/en-US/developers/

Add-on Developer Hub

Extension Workshop Documentation Support Blog

Customize Firefox

Whether you're new to extension development, polish
up or porting an existen
custom enterprise solution,
support

Learn how to mak

Ready to submit or manage your extension?

Sign in to the Developer Hub to submit or manage extensions and themes.

Register for a developer account or log in to the Developer Hub

developer.apple.com/safari/extensions/

Safari

Overview Tools Features Extensions Preview Resources What's New Download

Safari Extensions

Enhance and customize the web browsing experience on Mac,

developer.chrome.com/docs/extensions/

Chrome Developers

Search docs, blogs and more

Home

Docs

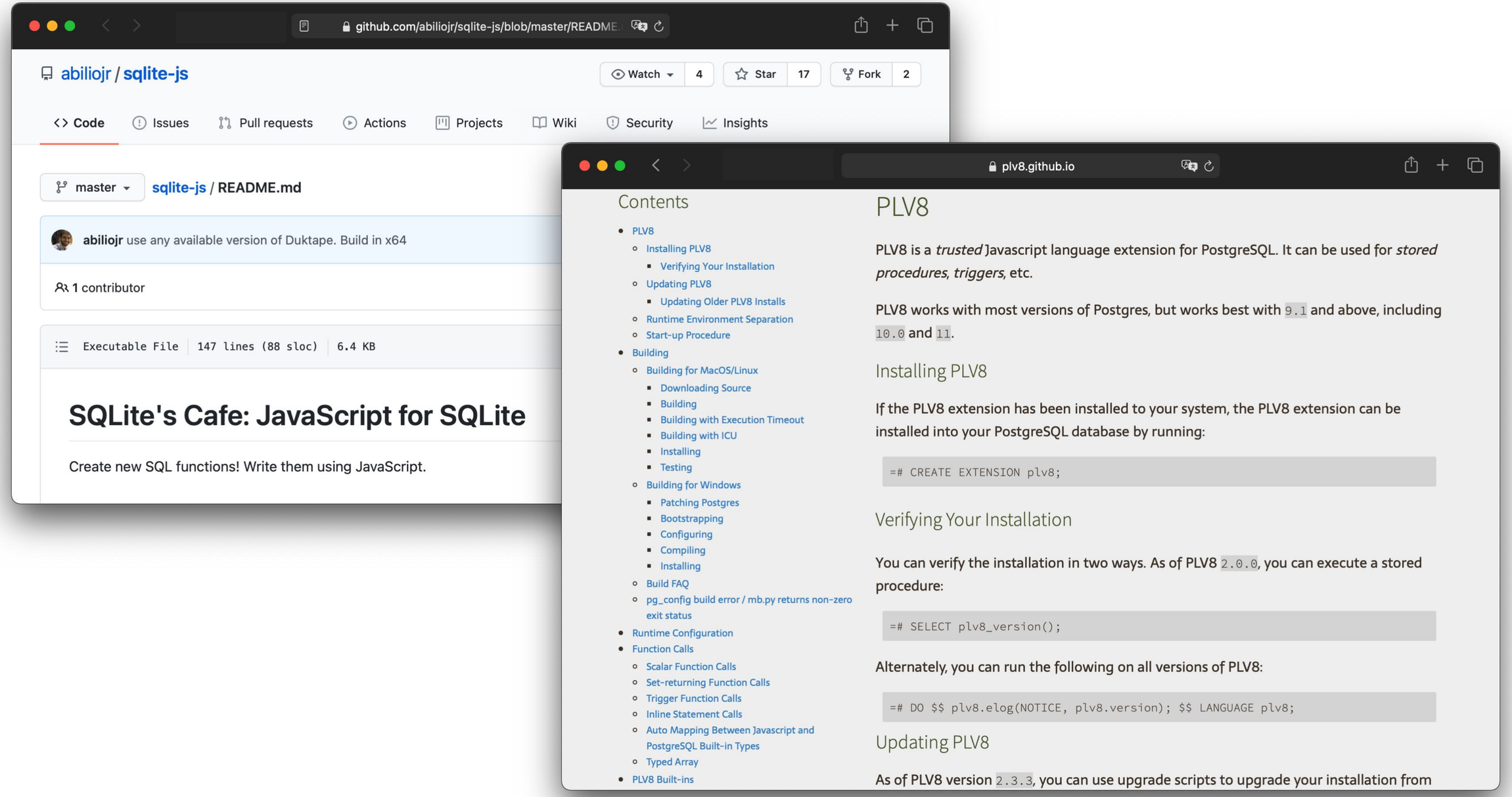
Blog

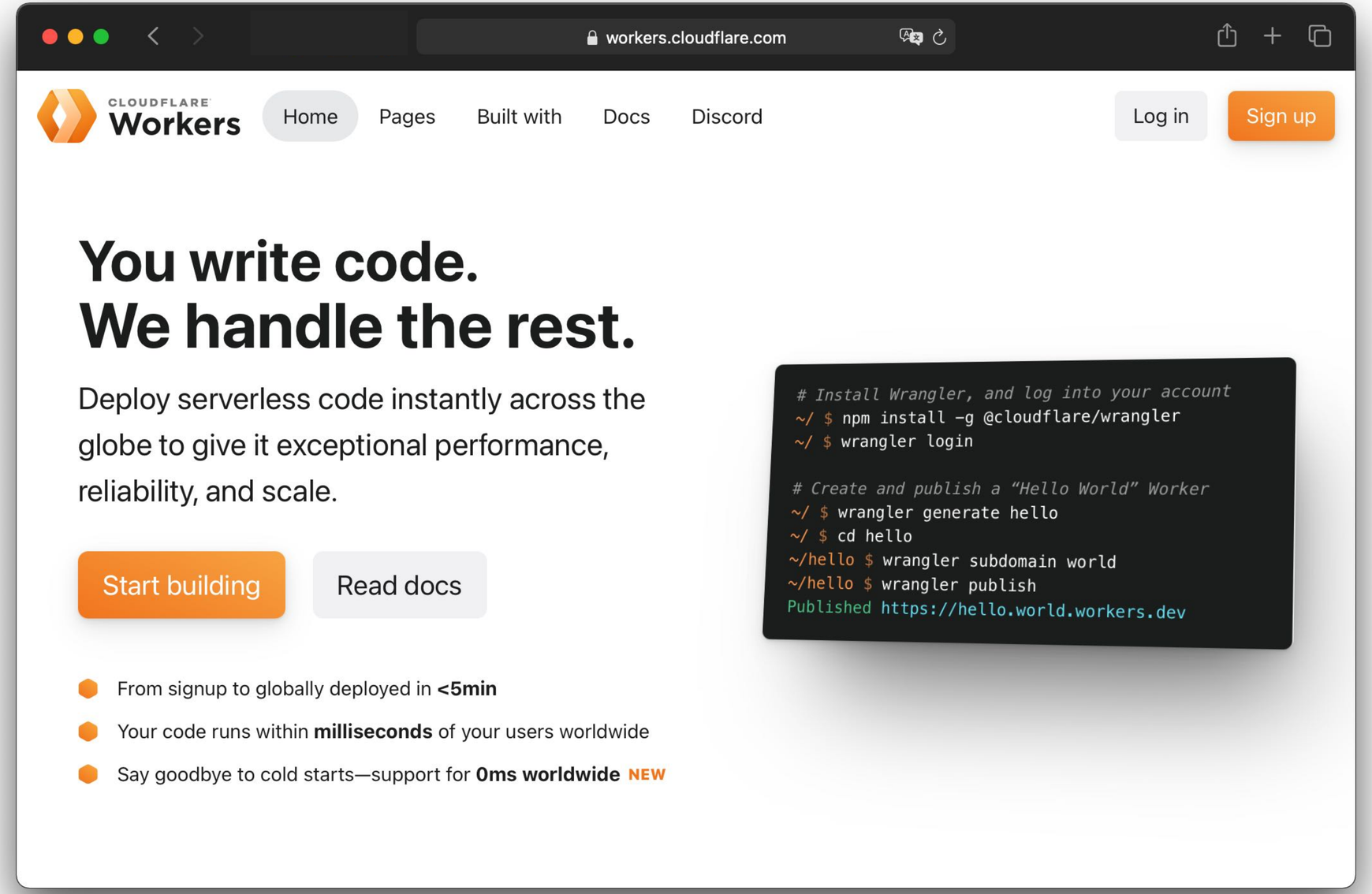
Documentation

Extensions

Extensions are software programs, built on web technologies (such as HTML, CSS, and JavaScript) that enable users to customize the Chrome browsing experience.

Welcome	Documentation for Chrome extensions developers.
What's new in Chrome extensions	Recent changes to the Chrome extensions platform, documentation, and policy

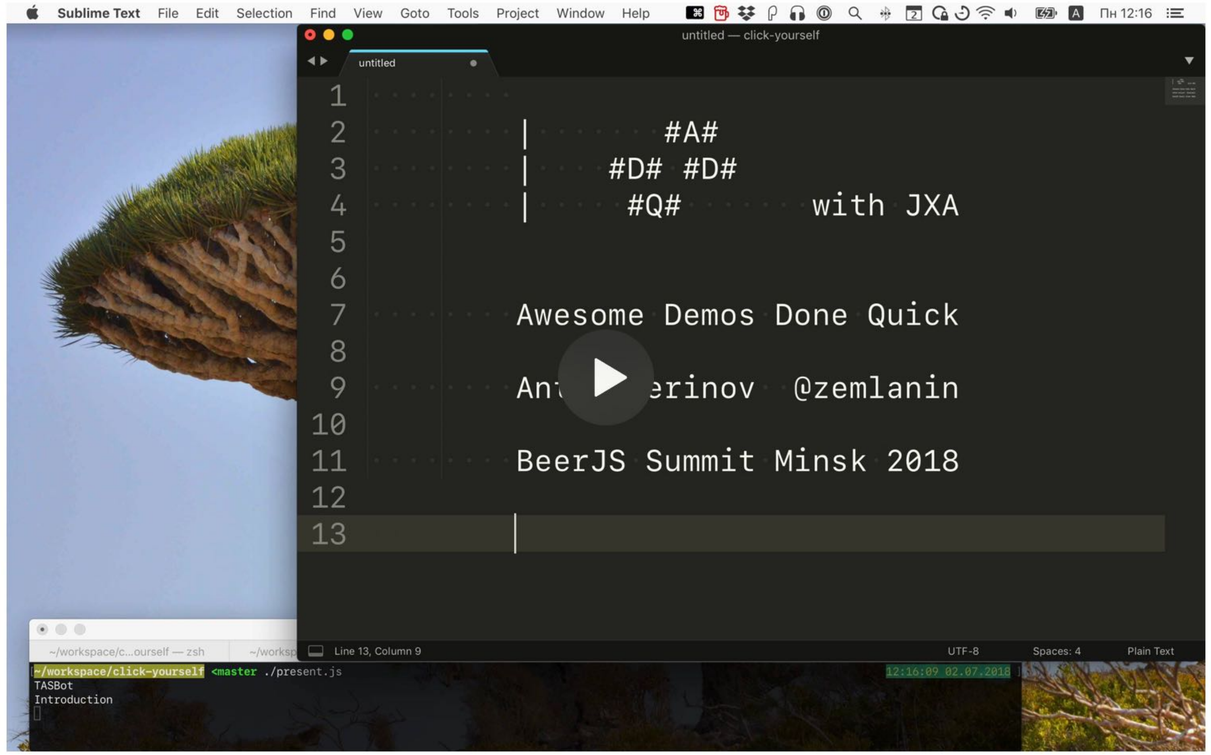

github.com/abiliojr/sqlite-js/blob/master/README
abiliojr / sqlite-js
Watch 4
Star 17
Fork 2
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
master
sqlite-js / README.md
abiliojr use any available version of Duktape. Build in x64
1 contributor
Executable File | 147 lines (88 sloc) | 6.4 KB
SQLite's Cafe: JavaScript for SQLite
Create new SQL functions! Write them using JavaScript.
plv8.github.io
Contents
PLV8
Installing PLV8
Verifying Your Installation
Updating PLV8
Updating Older PLV8 Installs
Runtime Environment Separation
Start-up Procedure
Building
Building for MacOS/Linux
Downloading Source
Building
Building with Execution Timeout
Building with ICU
Installing
Testing
Building for Windows
Patching Postgres
Bootstrapping
Configuring
Compiling
Installing
Build FAQ
pg_config build error / mb.py returns non-zero exit status
Runtime Configuration
Function Calls
Scalar Function Calls
Set-returning Function Calls
Trigger Function Calls
Inline Statement Calls
Auto Mapping Between Javascript and PostgreSQL Built-in Types
Typed Array
PLV8 Built-ins
PLV8
PLV8 is a trusted Javascript language extension for PostgreSQL. It can be used for stored procedures, triggers, etc.
PLV8 works with most versions of Postgres, but works best with 9.1 and above, including 10.0 and 11.
Installing PLV8
If the PLV8 extension has been installed to your system, the PLV8 extension can be installed into your PostgreSQL database by running:
=# CREATE EXTENSION plv8;
Verifying Your Installation
You can verify the installation in two ways. As of PLV8 2.0.0, you can execute a stored procedure:
=# SELECT plv8_version();
Alternately, you can run the following on all versions of PLV8:
=# DO $$ plv8.elog(NOTICE, plv8.version); $$ LANGUAGE plv8;
Updating PLV8
As of PLV8 version 2.3.3, you can use upgrade scripts to upgrade your installation from


workers.cloudflare.com
CLOUDFLARE
Workers
Home
Pages
Built with
Docs
Discord
Log in
Sign up
You write code.
We handle the rest.
Deploy serverless code instantly across the globe to give it exceptional performance, reliability, and scale.
Start building
Read docs
From signup to globally deployed in <5min
Your code runs within milliseconds of your users worldwide
Say goodbye to cold starts—support for 0ms worldwide NEW
Install Wrangler, and log into your account
~/ $ npm install -g @cloudflare/wrangler
~/ $ wrangler login
Create and publish a "Hello World" Worker
~/ $ wrangler generate hello
~/ $ cd hello
~/hello $ wrangler subdomain world
~/hello $ wrangler publish
Published https://hello.world.workers.dev

zemlan.in/awesome-demos-done-quick.html

zemlan.in

Awesome Demos Done Quick



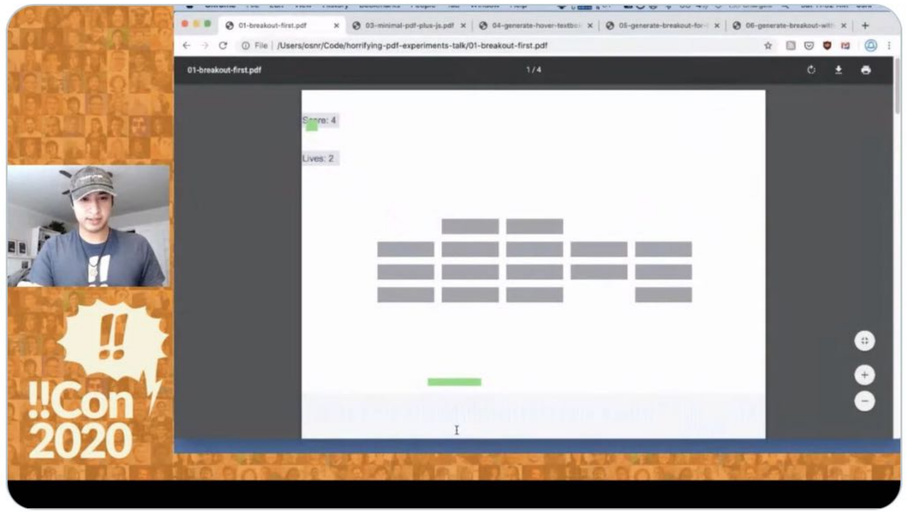
Выступление с BeerJS Summit Minsk 2018, про JXA и на JXA

twitter.com/rsnous/status/1259614184897

Thread

Omar @rsnous · May 11, 2020

it was great to talk about my Breakout-inside-a-PDF project at @bangbangcon yesterday! youtu.be/QEZ0N0rrbL0?t=...



1 37 119

Omar @rsnous · May 11, 2020

full source code (and working PDF!) at github.com/osnr/horrifyin...

horrifying-pdf-experiments

If you're not viewing it right now, try the breakout.pdf file in Chrome.

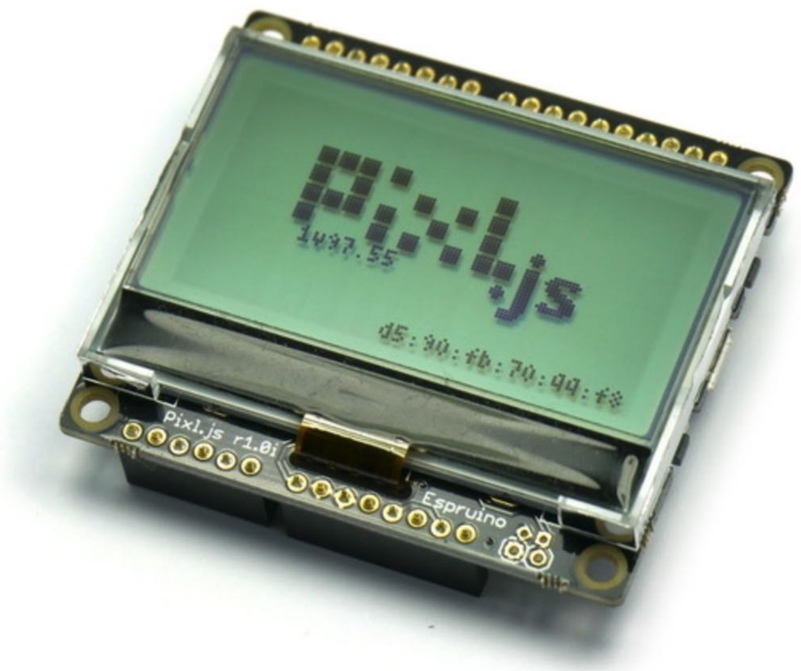
Like many of you, I always thought of PDF as basically a benign format, where the author lays out some text and graphics, and then the PDF sits in front of the reader and doesn't do anything. I heard offhand about vulnerabilities in Adobe Reader years ago, but didn't think too much about why or how they might exist.

www.espruino.com/Pixl.js

Espruino | Get Espruino | Donate | Documentation | Support | Search...

Contents

Pixl.js

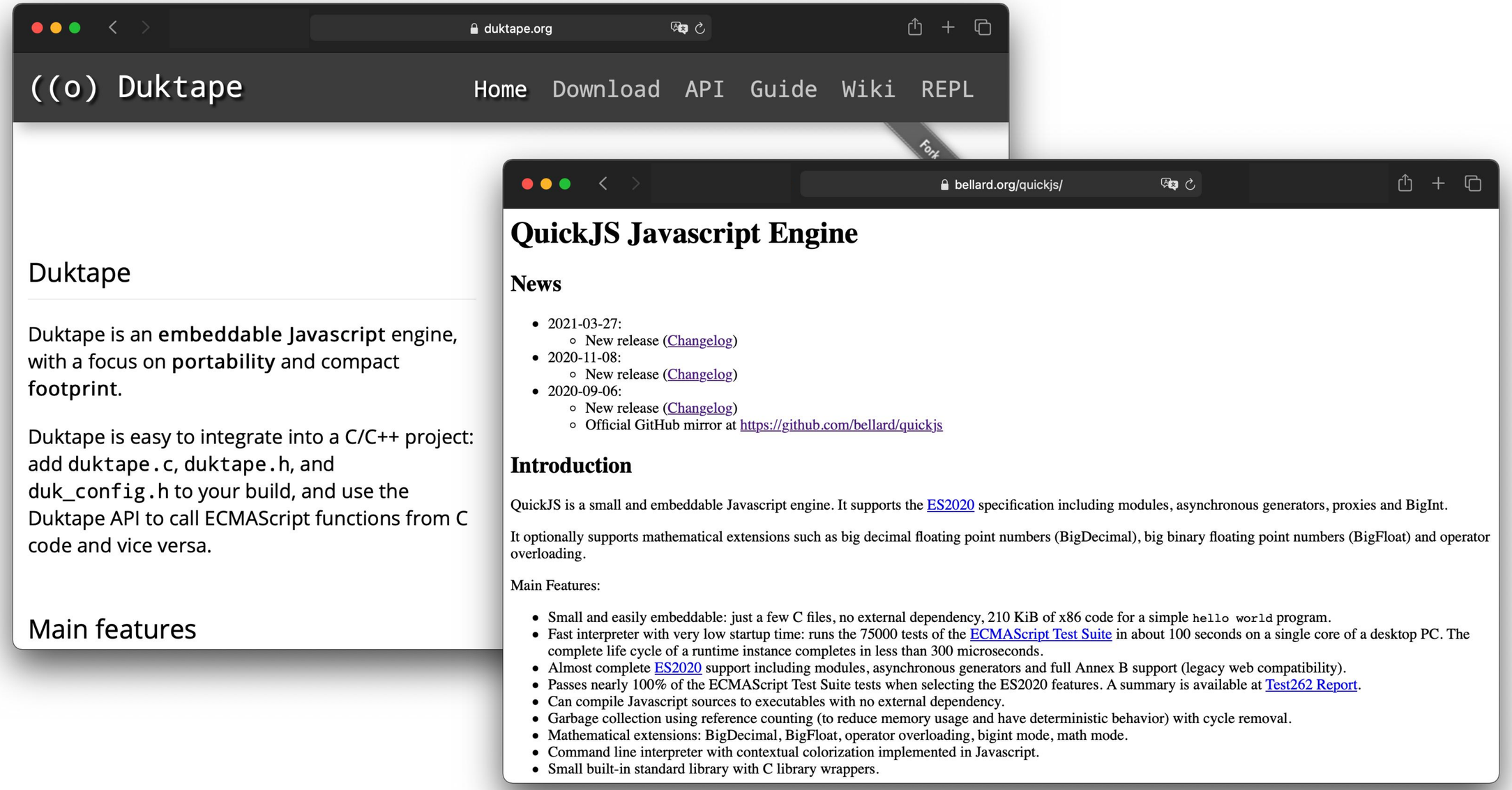


Pixl.js is a smart LCD with Bluetooth LE.

Monitor and control other Bluetooth LE devices, act as a wireless display, create your own smart conference badge, or even just display the status of your code - all while drawing tiny amounts of power.

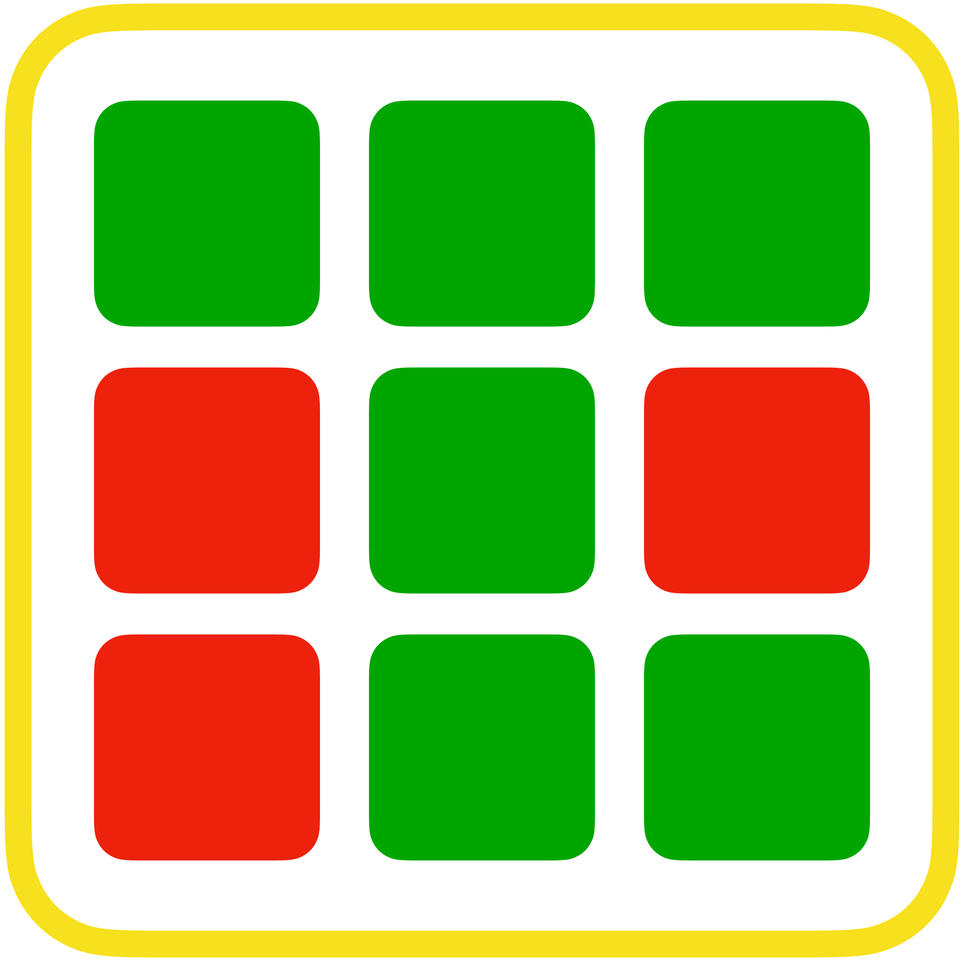
Buy Now

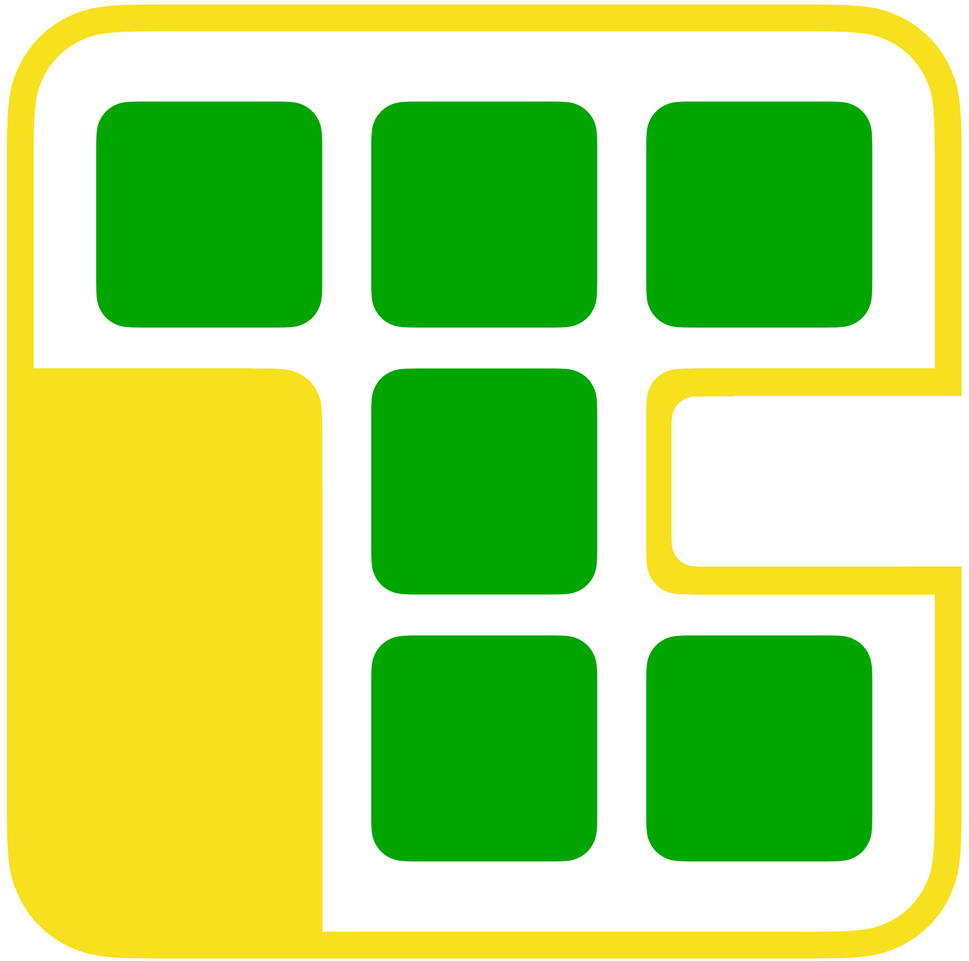
From £36

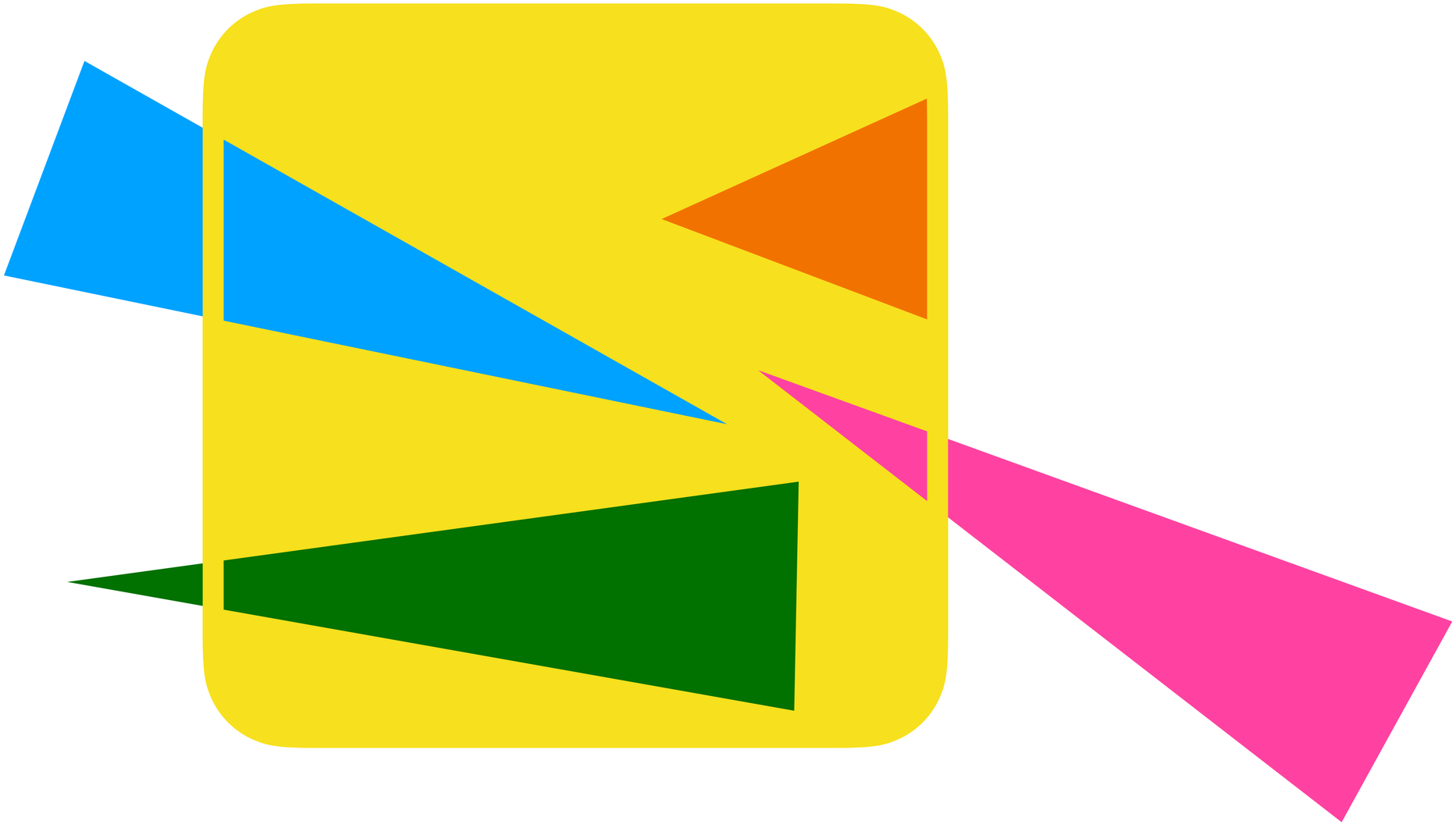

duktape.org
((o) Duktape
Home Download API Guide Wiki REPL
Fork
Duktape
Duktape is an embeddable Javascript engine, with a focus on portability and compact footprint.
Duktape is easy to integrate into a C/C++ project: add duktape.c, duktape.h, and duk_config.h to your build, and use the Duktape API to call ECMAScript functions from C code and vice versa.
Main features
bellard.org/quickjs/
QuickJS Javascript Engine
News
2021-03-27:
New release (Changelog)
2020-11-08:
New release (Changelog)
2020-09-06:
New release (Changelog)
Official GitHub mirror at https://github.com/bellard/quickjs
Introduction
QuickJS is a small and embeddable Javascript engine. It supports the ES2020 specification including modules, asynchronous generators, proxies and BigInt.
It optionally supports mathematical extensions such as big decimal floating point numbers (BigDecimal), big binary floating point numbers (BigFloat) and operator overloading.
Main Features:
Small and easily embeddable: just a few C files, no external dependency, 210 KiB of x86 code for a simple hello world program.
Fast interpreter with very low startup time: runs the 75000 tests of the ECMAScript Test Suite in about 100 seconds on a single core of a desktop PC. The complete life cycle of a runtime instance completes in less than 300 microseconds.
Almost complete ES2020 support including modules, asynchronous generators and full Annex B support (legacy web compatibility).
Passes nearly 100% of the ECMAScript Test Suite tests when selecting the ES2020 features. A summary is available at Test262 Report.
Can compile Javascript sources to executables with no external dependency.
Garbage collection using reference counting (to reduce memory usage and have deterministic behavior) with cycle removal.
Mathematical extensions: BigDecimal, BigFloat, operator overloading, bigint mode, math mode.
Command line interpreter with contextual colorization implemented in Javascript.
Small built-in standard library with C library wrappers.

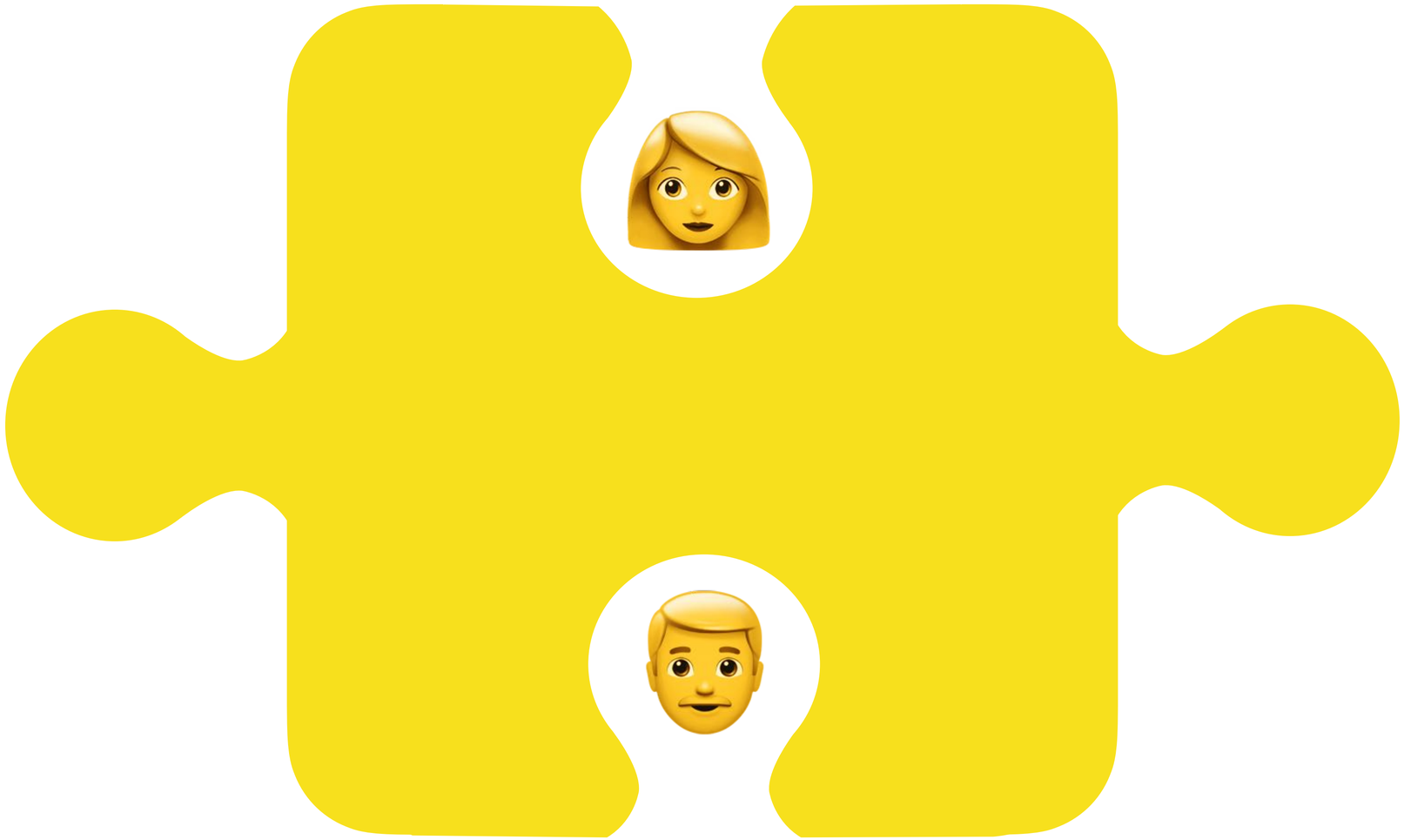

what did we learn Palmer?


JS
JS
JS
JS









anton.click/fwdays

@zemlanin