

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

**ANTON VERINOV**

Wix Engineering



**frontend** →



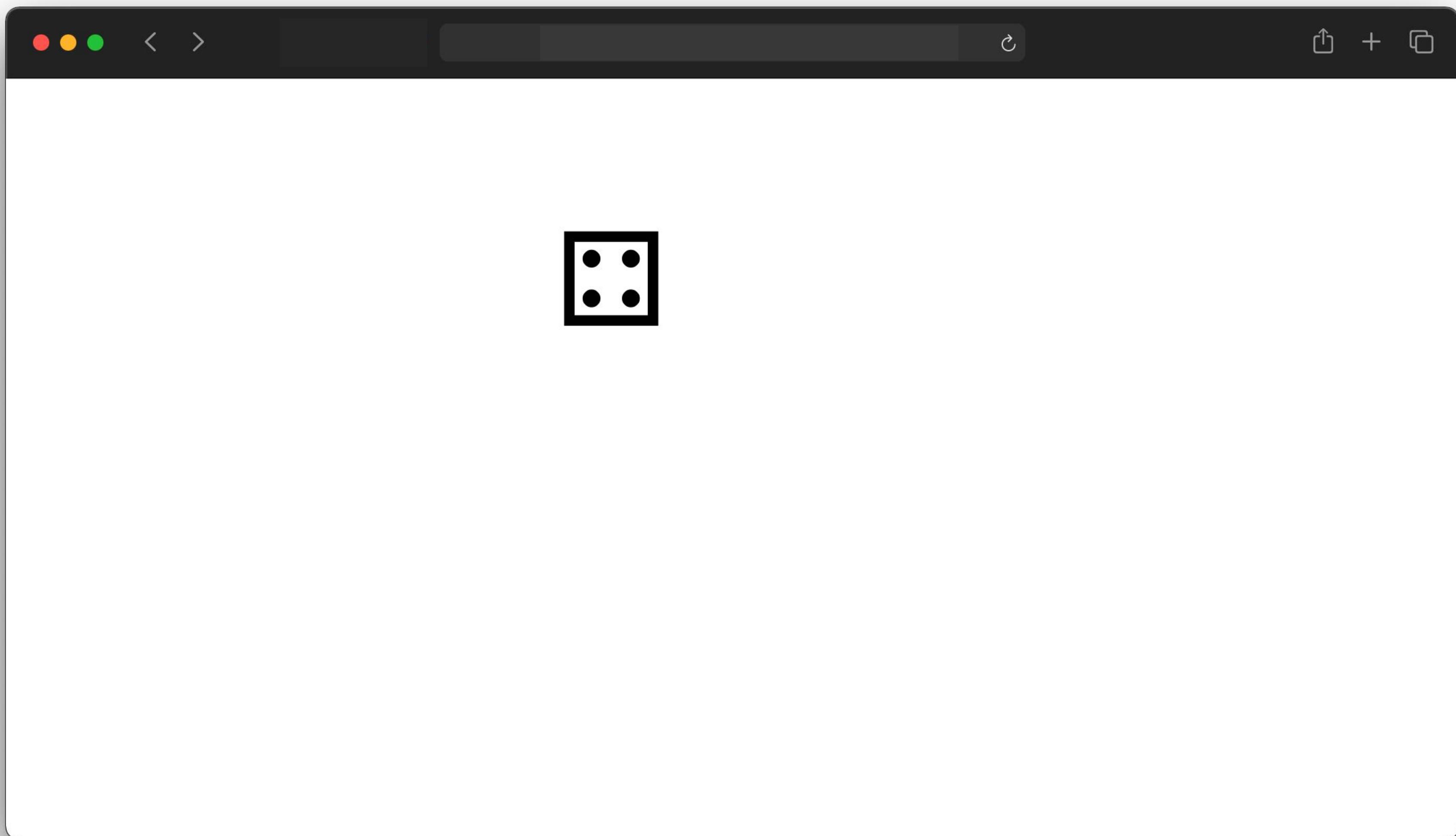
← **@zemlanin**

1

9

9

5



A screenshot of a dark-themed code editor window titled "noscript.html — mo-dice". The code editor displays the following content:

```
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 |
```

The editor's status bar at the bottom shows "Line 8, Column 1" on the left, and "main Spaces: 2 HTML" on the right.



This plug-in is not supported.



# COMPUTERWORLD

FROM IDG



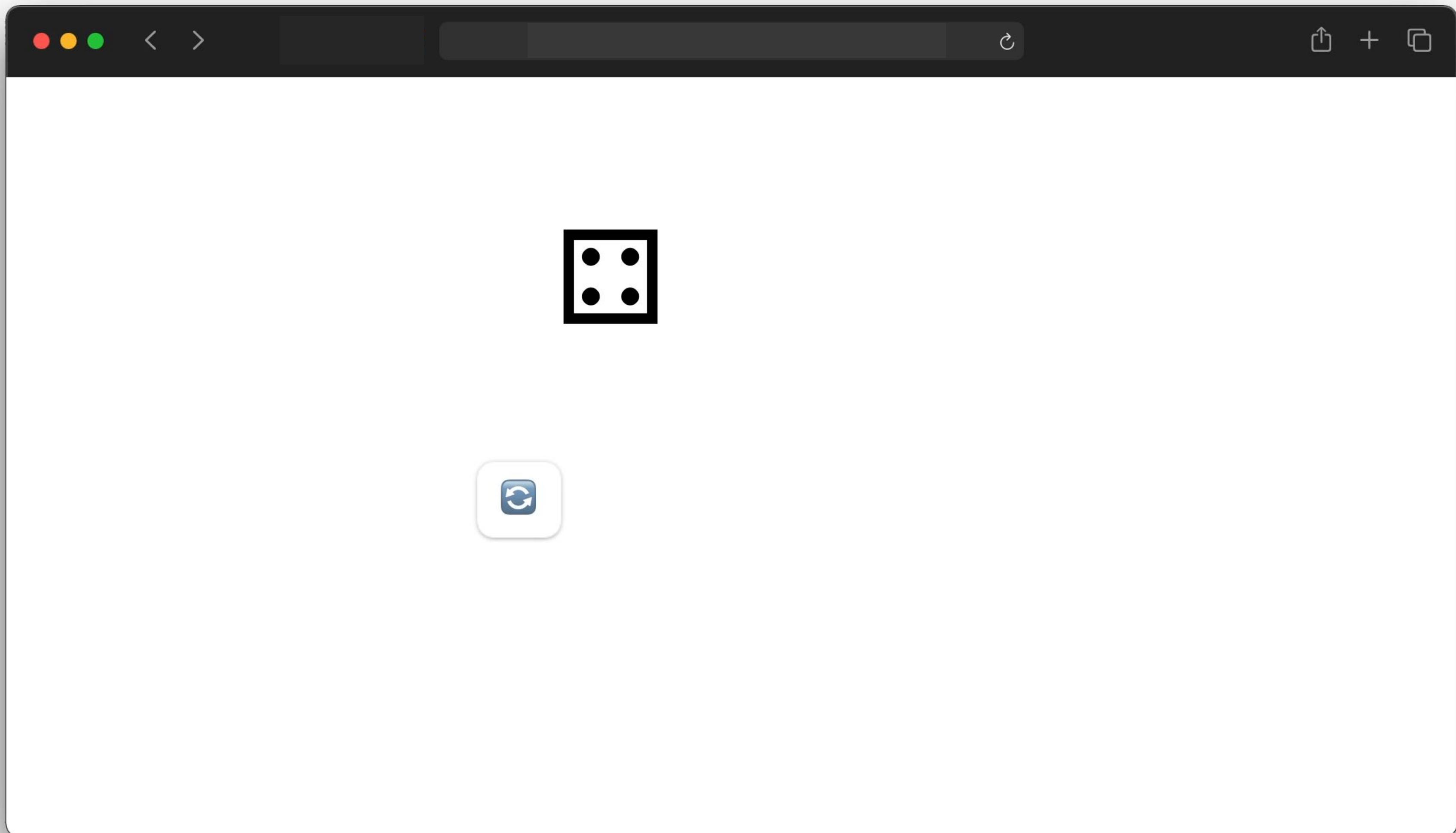
The impetus was the belief on the part of at least Marc Andreessen and myself, along with Bill Joy of Sun, that HTML needed a "scripting language", a programming language that was easy to use by amateurs and novices, where the code could be written directly in source form as part of the Web page markup. 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
1 const R = (n) => parseInt(Math.random() * n);
2
3 class Core {
4     static symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
5
6     constructor(options) {
7         options = options || {};
8
9         this.symbols = options.symbols || Core.symbols;
10        this.lastRoll =
11            typeof options.lastRoll === "number"
12            ? // 0 < options.lastRoll < this.symbols.length
13                Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
14            : R(this.symbols.length);
15        this.history = options.history || [];
16    }
17
18    roll() {
19        this.lastRoll = R(this.symbols.length);
20        this.history = [this.lastRoll, ...this.history].slice(0, 10);
21    }
22
23    pretty() {
24        return {
25            lastRoll: this.symbols[this.lastRoll],

```

Line 36, Column 1      main      Spaces: 2      JavaScript

core.js — mo-dice

```
12     ? // 0 < options.lastRoll < this.symbols.length
13         Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
14     : R(this.symbols.length);
15     this.history = options.history || [];
16 }
17
18 roll() {
19     this.lastRoll = R(this.symbols.length);
20     this.history = [this.lastRoll, ...this.history].slice(0, 10);
21 }
22
23 pretty() {
24     return {
25         lastRoll: this.symbols[this.lastRoll],
26         history: this.history.map((e) => this.symbols[e]),
27     };
28 }
29
30 clear() {
31     this.history = [];
32 }
33 }
34
35 export default Core;
36
```

Line 1, Column 1

main Spaces: 2 JavaScript

A screenshot of a code editor window titled "index.html — mo-dice". The code is a combination of CSS and JavaScript. The CSS part includes styling for the body, a large font-size for the last roll, and a log of rolls. The JavaScript part imports a Core module and defines a renderRoll function that updates the last roll's position using CSS transforms.

```
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"></ul>
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

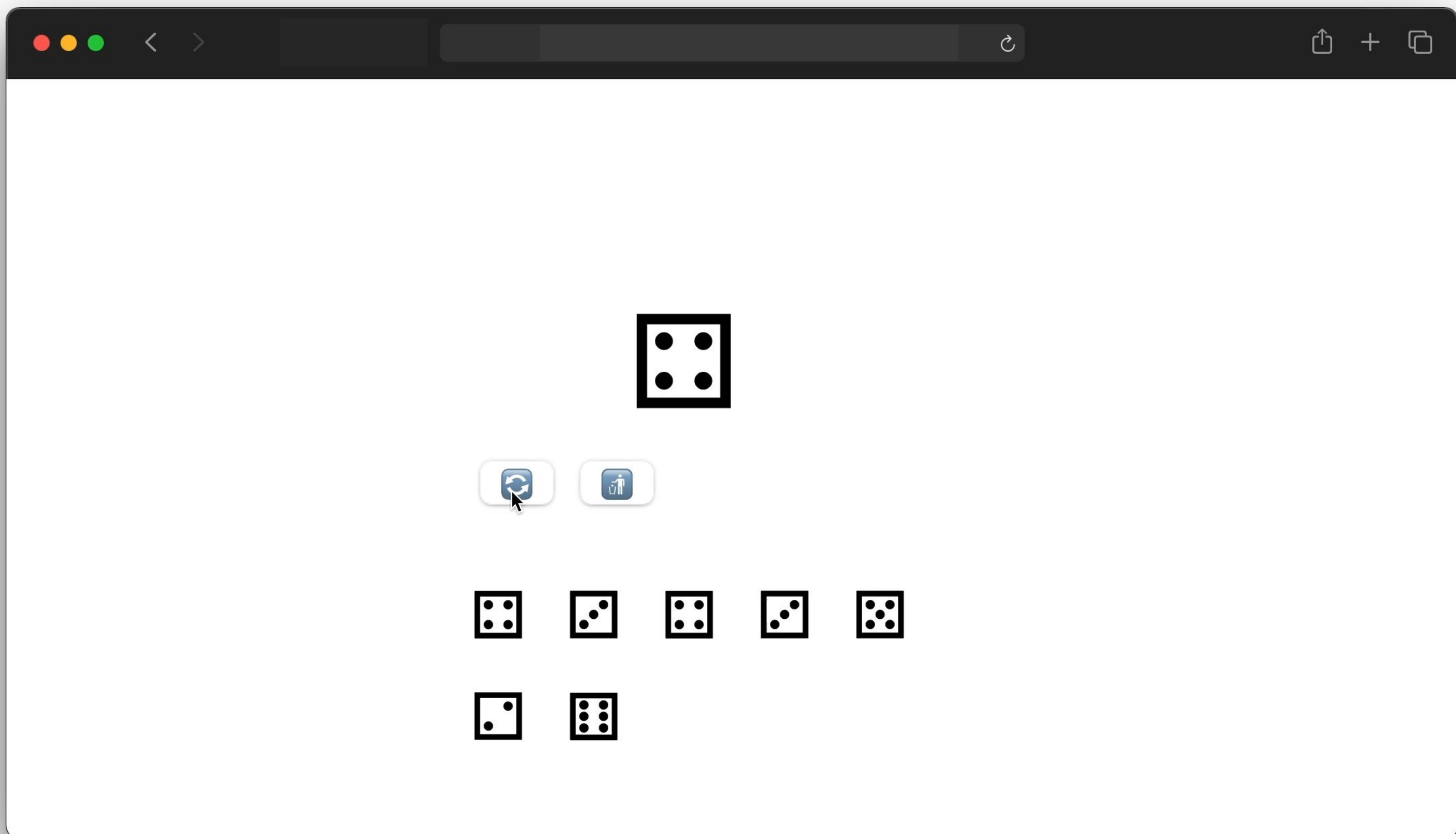
main Spaces: 2 HTML

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-standards

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.

A screenshot of a macOS terminal window displaying a GitHub gist of C++ code. The window has a dark theme with red, yellow, and green window controls. The title bar shows the URL [github.com/WebKit/WebKit/blob/36582be546b17e05d6](https://github.com/WebKit/WebKit/blob/36582be546b17e05d6). The main area contains the following code:

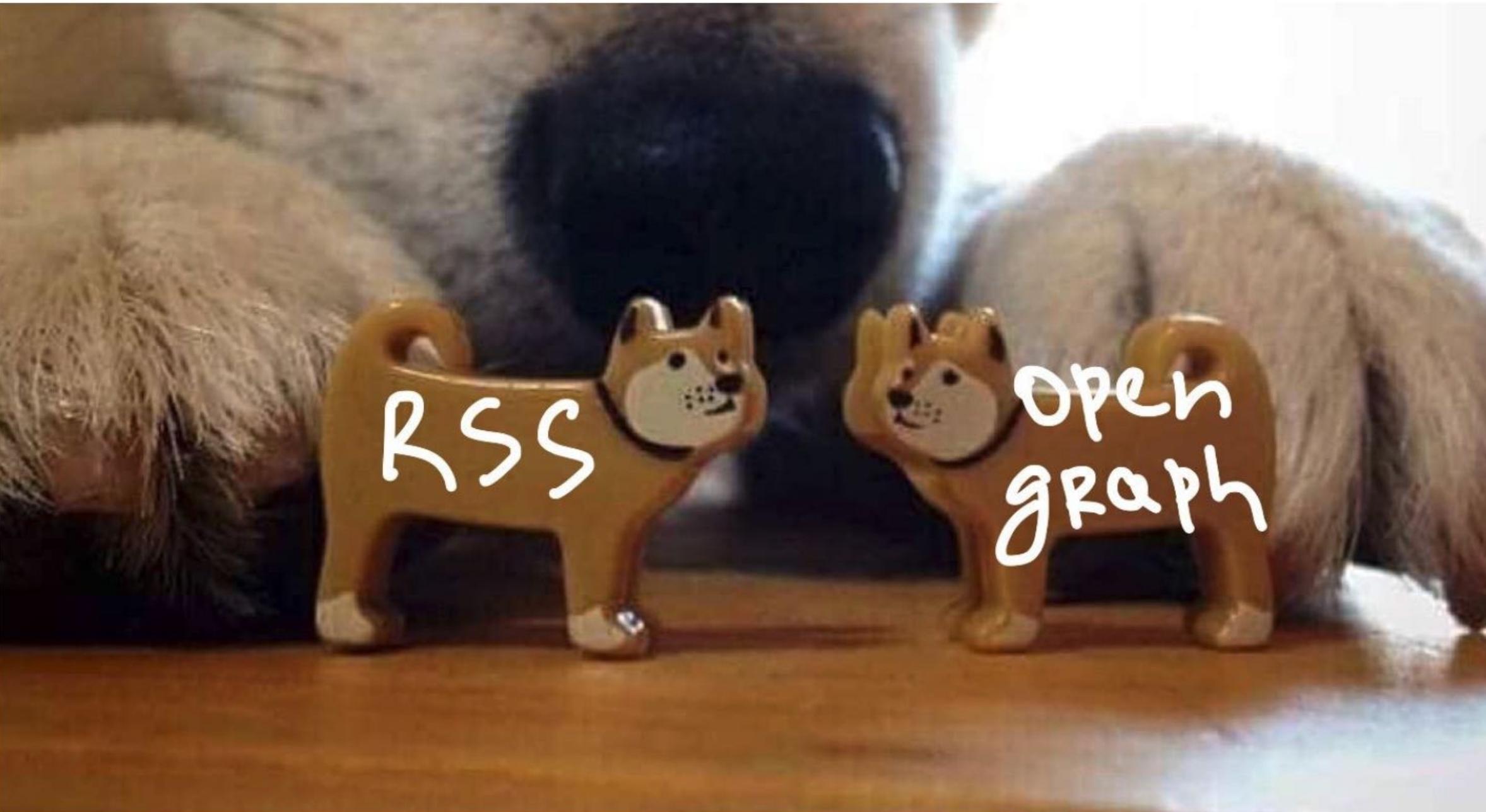
```
377     if (isAmazon())
378         return ShouldDispatchSimulatedMouseEvents::Yes;
379     if (isGoogleMaps())
380         return ShouldDispatchSimulatedMouseEvents::Yes;
381
382     auto& url = m_document->topDocument().url();
383     auto host = url.host().convertToASCIILowercase();
384
385     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 <sup>ka</sup>

zemlan.in <sup>ka</sup>

Склепал <sup>а</sup> линклог



...из скуки и открытых стандартов

816 слов →

2015  
2011-04-19

The screenshot shows a web browser window displaying the official nginx documentation page at [nginx.org/en/](https://nginx.org/en/). The page features a dark header bar with the URL and a top navigation bar with links to "NGINX Service Mesh 1.0" and "Learn more.". Below this is a large green "nginx" logo. The main content area contains a sidebar with links to various nginx features and a central column with a brief history and success stories. A vertical sidebar on the right lists additional links related to the nginx project.

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

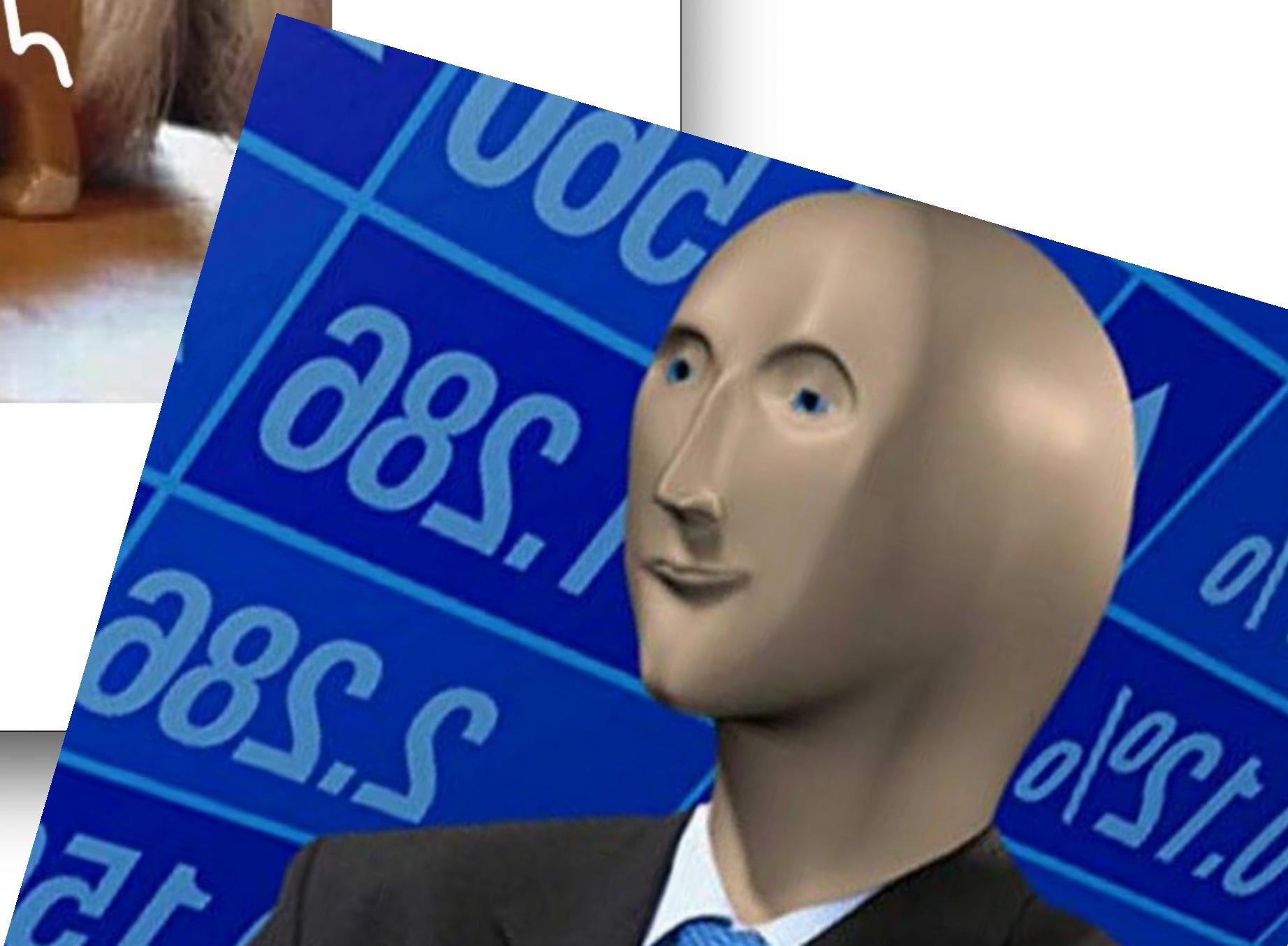
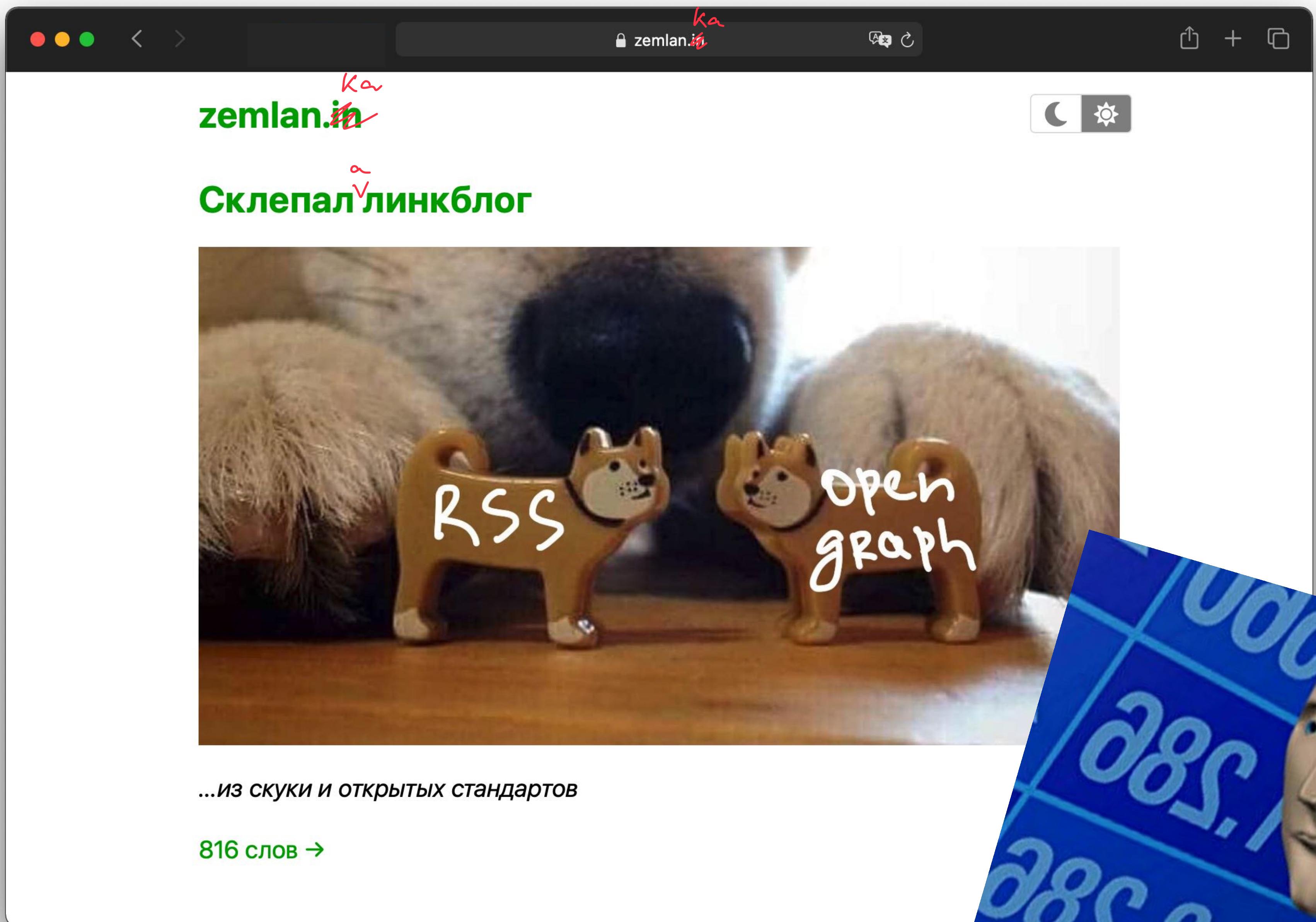
- Serving static and [index](#) files, [autoindexing](#); [open file descriptor cache](#);
- [Accelerated reverse proxying with caching](#); [load balancing and fault tolerance](#);
- Accelerated support with caching of [FastCGI](#), [uwsgi](#), [SCGI](#), and [memcached](#) servers; [load balancing 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;

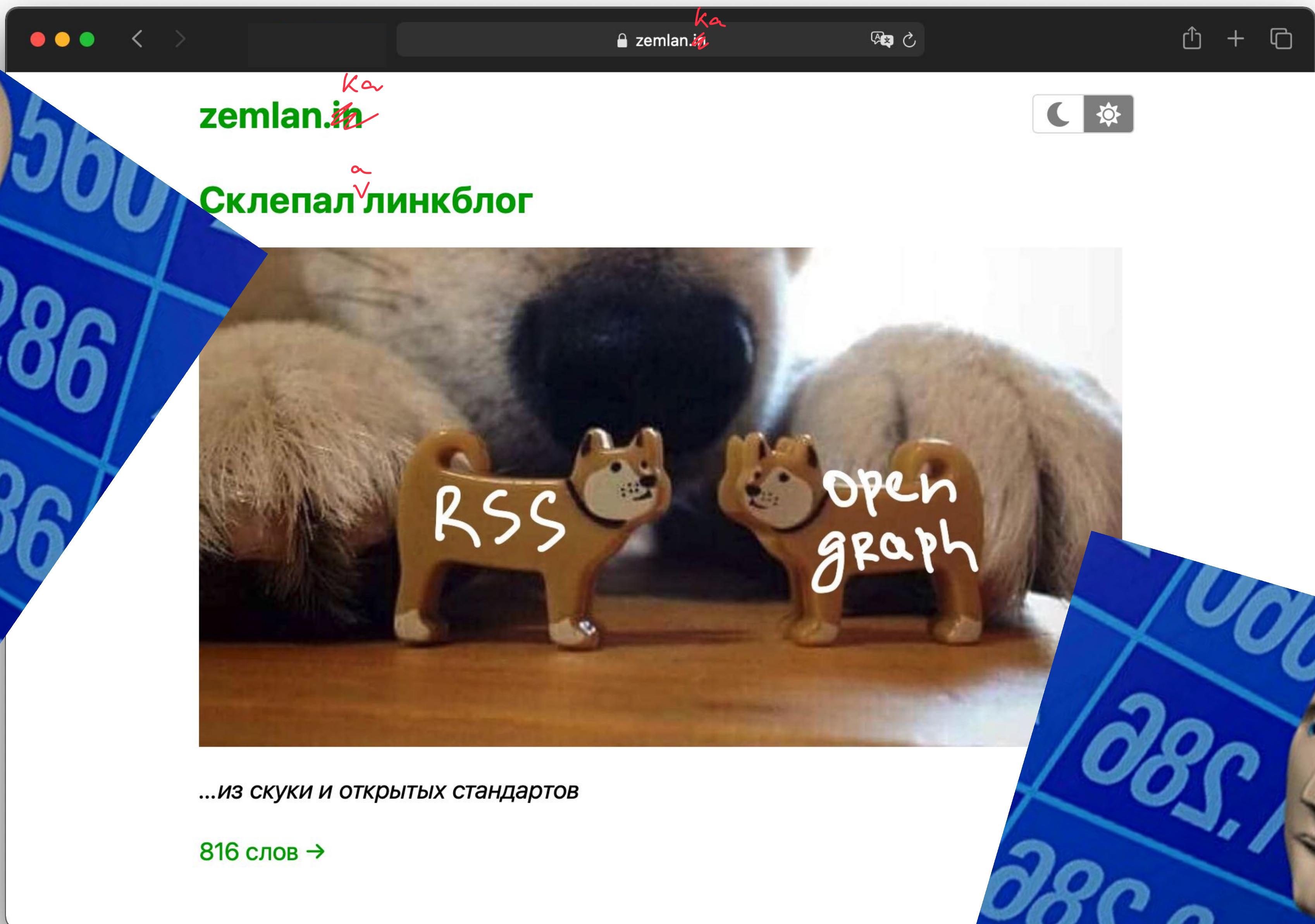
english  
русский

[news](#)  
[about](#)  
[download](#)  
[security](#)  
[documentation](#)  
[faq](#)  
[books](#)  
[support](#)

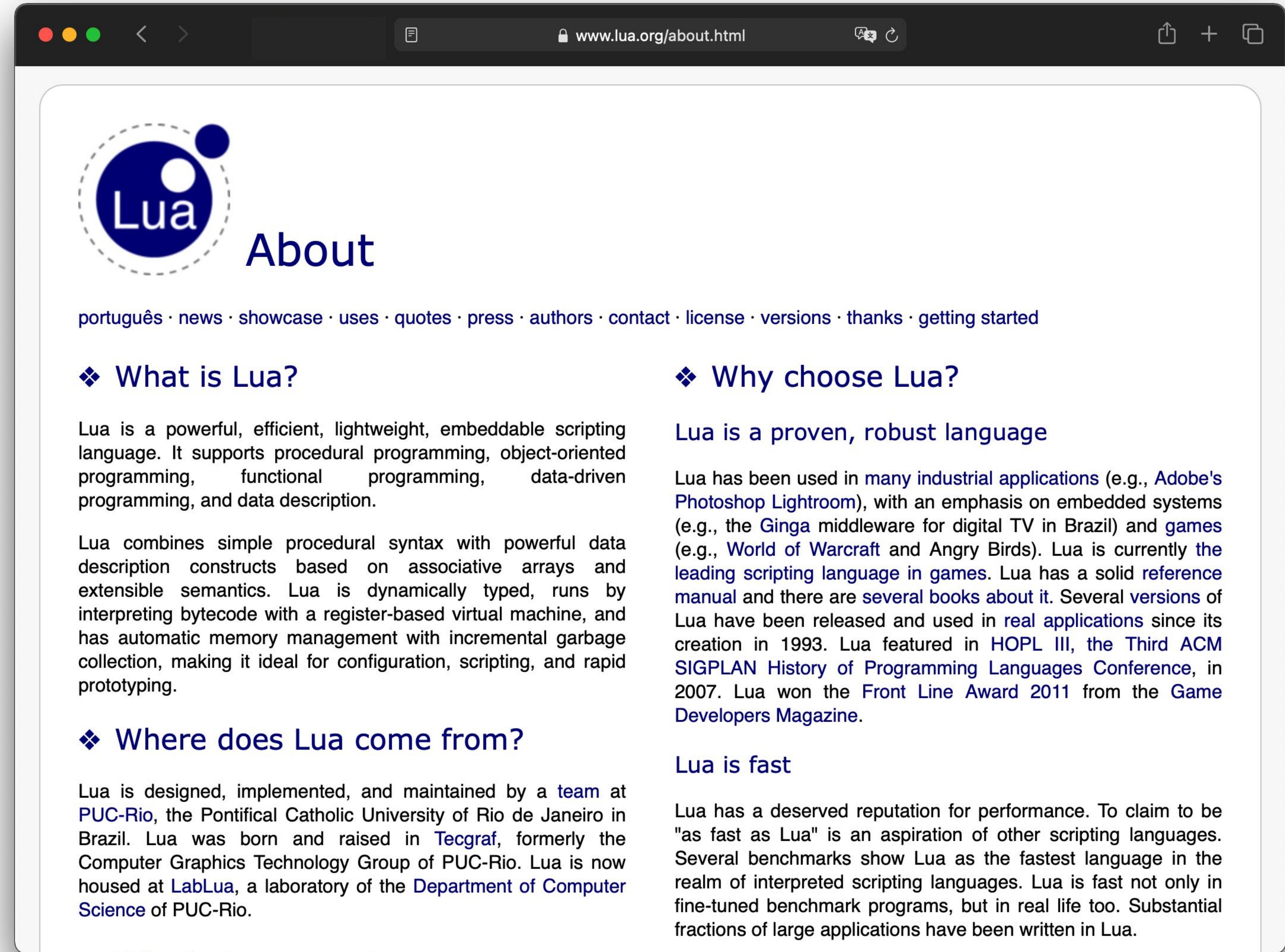
[trac](#)  
[twitter](#)  
[blog](#)

[unit](#)  
[njs](#)









The screenshot shows a web browser window displaying the "About" page of the Lua website at [www.lua.org/about.html](http://www.lua.org/about.html). The page features the Lua logo (a blue circle with white text) and navigation links for various sections like português, news, showcase, uses, quotes, press, authors, contact, license, versions, thanks, and getting started. The main content is divided into two columns: "What is Lua?" and "Why choose Lua?", each with its own sub-sections and descriptions.

## 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.

The screenshot shows a web browser window with the URL [nginx.org/en/docs/njs/](https://nginx.org/en/docs/njs/) in the address bar. The page content is as follows:

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](#)

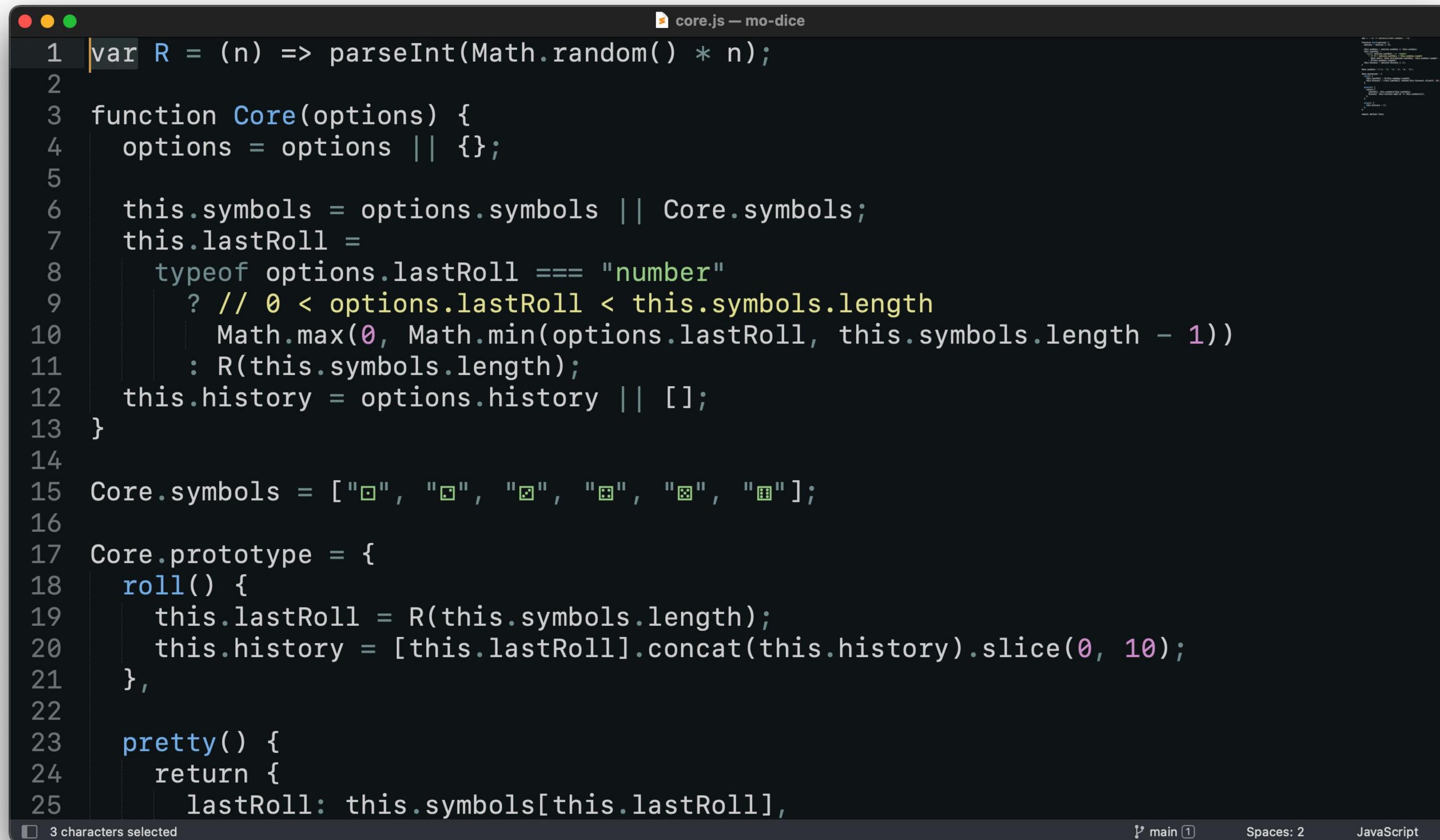
**Use cases**

- 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.

On the right side of the page, there is a vertical sidebar with the NGINX logo at the top. Below the logo, there are links in two columns:

- english
- русский
- news
- about
- download
- security
- documentation
- faq
- books
- support
- trac
- twitter
- blog
- unit
- njs



A screenshot of a code editor window titled "core.js — mo-dice". The code is written in JavaScript and defines a class named "Core". The editor has a dark theme with syntax highlighting. The code includes a static array "Core.symbols" containing six dice faces, and methods "roll()" and "pretty()". The "roll()" method generates a random index for "symbols" and updates "history". The "pretty()" method returns an object with "lastRoll" set to the current value of "lastRoll".

```
1 var R = (n) => parseInt(Math.random() * n);
2
3 function Core(options) {
4     options = options || {};
5
6     this.symbols = options.symbols || Core.symbols;
7     this.lastRoll =
8         typeof options.lastRoll === "number"
9         ? // 0 < options.lastRoll < this.symbols.length
10            Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
11         : R(this.symbols.length);
12     this.history = options.history || [];
13 }
14
15 Core.symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
16
17 Core.prototype = {
18     roll() {
19         this.lastRoll = R(this.symbols.length);
20         this.history = [this.lastRoll].concat(this.history).slice(0, 10);
21     },
22
23     pretty() {
24         return {
25             lastRoll: this.symbols[this.lastRoll],
26         };
27     }
28 }
```

A screenshot of a code editor window titled "core.js — mo-dice". The code is written in JavaScript and defines a class named Core. The editor has a dark theme with syntax highlighting. The code includes a static symbols array and two methods: roll() and pretty(). The roll() method generates a random index and concatenates it with the history. The pretty() method returns an object with lastRoll and its symbol.

```
1 var R = (n) => parseInt(Math.random() * n);
2
3 function Core(options) {
4     options = options || {};
5
6     this.symbols = options.symbols || Core.symbols;
7     this.lastRoll =
8         typeof options.lastRoll === "number"
9         ? // 0 < options.lastRoll < this.symbols.length
10            Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
11         : R(this.symbols.length);
12     this.history = options.history || [];
13 }
14
15 Core.symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
16
17 Core.prototype = {
18     roll() {
19         this.lastRoll = R(this.symbols.length);
20         this.history = [this.lastRoll].concat(this.history).slice(0, 10);
21     },
22
23     pretty() {
24         return {
25             lastRoll: this.symbols[this.lastRoll],
26         };
27     }
28 }
```

```
core.js — mo-dice
12 |   this.history = options.history || [];
13 |
14 |
15 Core.symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
16
17 Core.prototype = {
18   roll() {
19     this.lastRoll = R(this.symbols.length);
20     this.history = [this.lastRoll].concat(this.history).slice(0, 10);
21   },
22
23   pretty() {
24     return {
25       lastRoll: this.symbols[this.lastRoll],
26       history: this.history.map((e) => this.symbols[e]),
27     };
28   },
29
30   clear() {
31     this.history = [];
32   },
33 };
34
35 export default Core;
36
```

20 characters selected    main ①    Spaces: 2    JavaScript

A screenshot of a terminal window on a Mac OS X desktop. The window title is "modice.js — mo-dice". The terminal displays a Node.js script with line numbers and syntax highlighting. The script performs several tasks:

- Imports "Core" from "./core.js".
- Defines a function "load(r)" which attempts to parse a cookie from the URL component "r.variables.cookie\_modice" into a "Core" object using JSON.parse(). If this fails, it logs the error to "ngx.ERR".
- If "load(r)" fails, it returns a new "Core" object.
- Defines a function "save(r, core)" which sets a "Set-Cookie" header in the response object "r.headersOut" with the value "modice=\${encodeURIComponent(JSON.stringify(core))}".
- Defines a function "luckyResponse(r)" which uses the "ngx" module to fetch the URL "http://zemlan.in/" and then replies with the fetched content.

The terminal status bar at the bottom shows "Line 68, Column 1" and tabs for "main [2]", "Spaces: 2", and "JavaScript".

```
1 import Core from "./core.js";
2
3 function load(r) {
4     try {
5         return new Core(
6             JSON.parse(decodeURIComponent(
7                 r.variables.cookie_modice
8             )));
9     } catch (e) {
10        ngx.log(ngx.ERR, e);
11    }
12    return new Core();
13 }
14
15
16 function save(r, core) {
17     r.headersOut["Set-Cookie"] = `modice=${encodeURIComponent(
18         JSON.stringify(core)
19     )}`;
20 }
21
22 function luckyResponse(r) {
23     ngx
24         .fetch("http://zemlan.in/")
25         .then((reply) => {
```

```
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.referrer);
65 }
66
67 export default { index, clear };
68
```

Line 2, Column 1      main [2]      Spaces: 2      JavaScript

modice.js — mo-dice

```
22 function luckyResponse(r) {
23   ngx
24     .fetch("http://zemlan.in/")
25     .then((reply) => {
26       return reply.text().then((body) => {
27         r.headersOut["Content-Type"] = reply.headers.get("Content-Type");
28         r.return(reply.status, body);
29       });
30     })
31     .catch((e) => r.return(500, e.toString()));
32 }
33
34 function rollAgainResponse(r, core) {
35   r.headersOut["Content-Type"] = "text/html; charset=utf-8";
36
37   r.return(
38     403,
39     `<title>Try Again</title>
40     <style>body {text-align: center} code {font-size: 80px} a {display: block}</
41     <hr>
42     <code>${core.pretty().history.join(" ")}</code>
43     <a href="/clear">⬅</a>` 
44   );
45 }
46
```

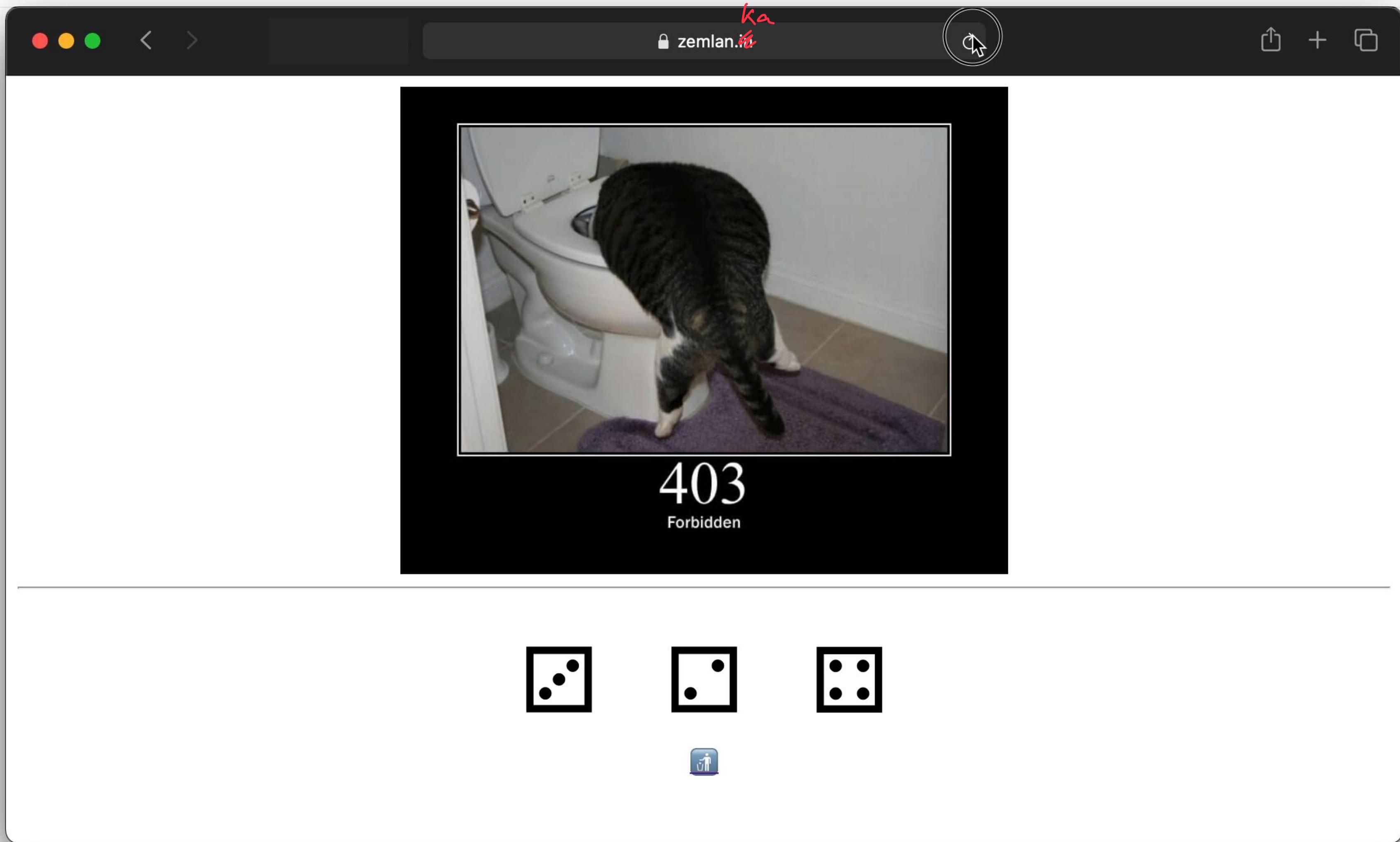
Line 68, Column 1      main [2]      Spaces: 2      JavaScript

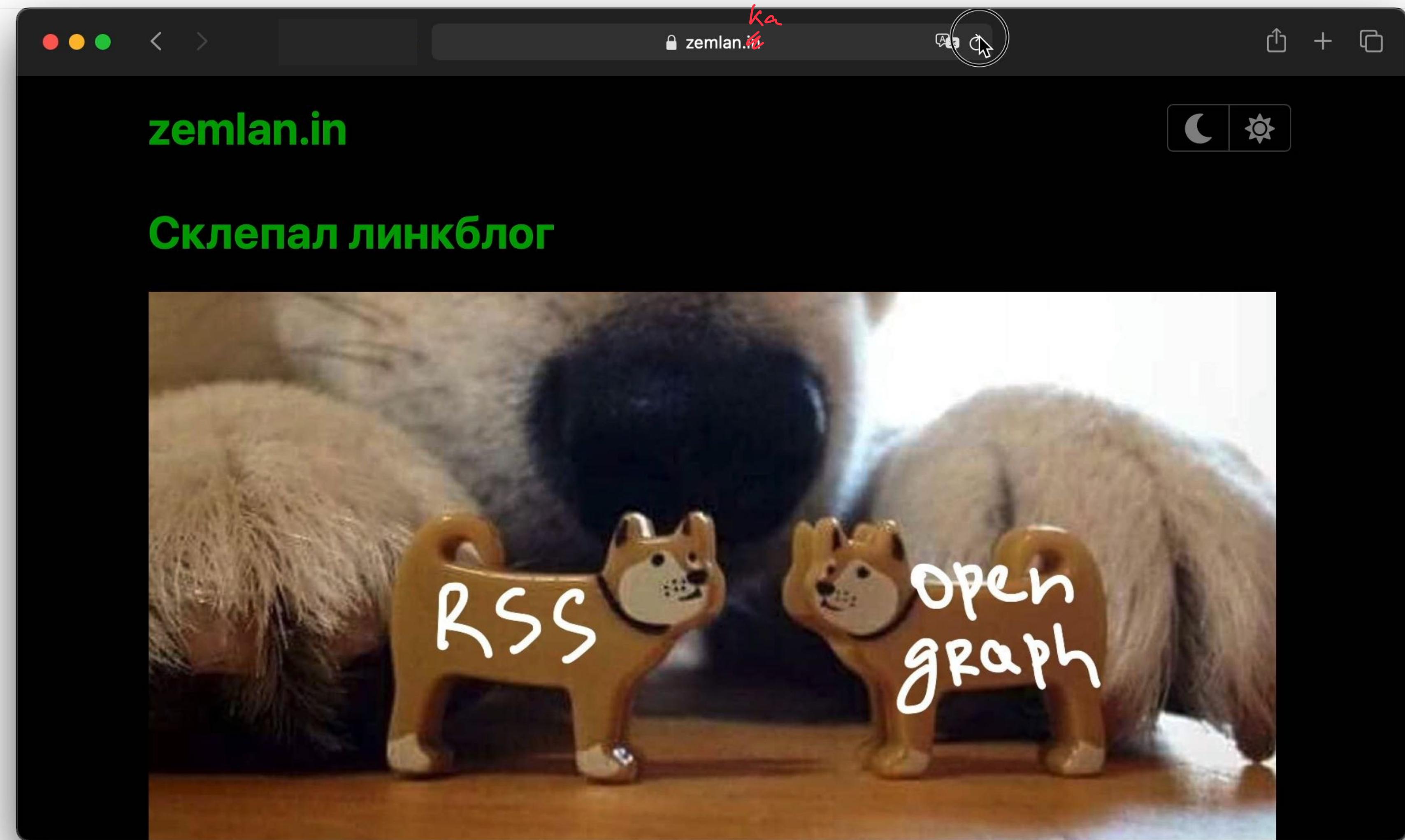
nginx.conf — mo-dice

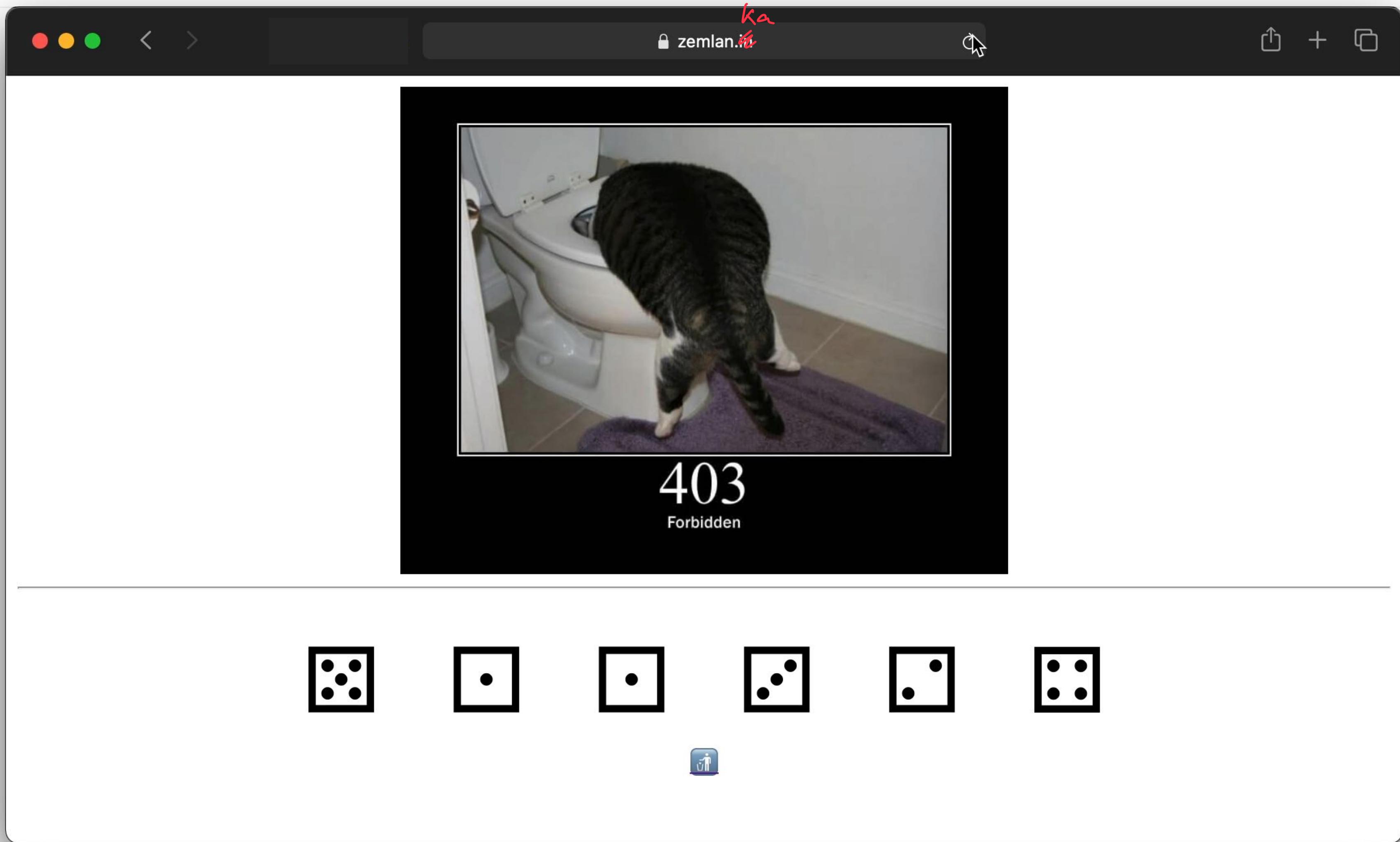
```
1 load_module modules/ngx_http_js_module.so;
2
3 events {}
4
5 http {
6     js_import modice.js;
7
8     server {
9         listen 80;
10        server_name localhost;
11        resolver 1.1.1.1;
12
13        location = / {
14            js_content modice.index;
15        }
16
17        location = /clear {
18            js_content modice.clear;
19        }
20    }
21 }
22 |
```

Line 22, Column 1

main Spaces: 2 Plain Text







2017

A screenshot of a Google Sheets document titled "budget". The document contains one sheet named "Sheet1". In the first row (row 1), cell A1 contains the word "profit" and cell C1 contains the value "10,000,000\$". The cell C1 is currently selected, indicated by a blue border around it. The formula bar at the top shows the value "10000000" next to the cell reference "C1". The spreadsheet interface includes standard toolbar icons for file operations, sharing, and various data and format tools.

	A	B	C	D	E
1	profit		10,000,000\$		
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

# 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	<code>ABS(value)</code>	Returns the absolute value of a number. <a href="#">Learn more</a>
Math	ACOS	<code>ACOS(value)</code>	Returns the inverse cosine of a value, in radians. <a href="#">Learn more</a>
Math	ACOSH	<code>ACOSH(value)</code>	Returns the inverse hyperbolic cosine of a number. <a href="#">Learn more</a>
Math	ACOT	<code>ACOT(value)</code>	Returns the inverse cotangent of a value, in radians. <a href="#">Learn more</a> .
Math	ACOTH	<code>ACOTH(value)</code>	Returns the inverse hyperbolic cotangent of a value, in radians. Must not be between -1 and 1, inclusive. <a href="#">Learn more</a> .
			Returns the inverse sine of

**Functions and formulas**

- [Add formulas & functions](#)
- [See the sum & average](#)
- [Reference data from other sheets](#)
- [Google Sheets function list](#)

# 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	<code>AVEDEV(value1, [value2, ...])</code>	Calculates the average of the magnitudes of deviations of data from a dataset's mean. <a href="#">Learn more</a>
Statistical	AVERAGE	<code>AVERAGE(value1, [value2, ...])</code>	Returns the numerical average value in a dataset, ignoring text. <a href="#">Learn more</a>
Statistical	AVERAGE.WEIGHTED	<code>AVERAGE.WEIGHTED(values, weights, [additional values], [additional weights])</code>	Finds the weighted average of a set of values, given the values and the corresponding weights. <a href="#">Learn more</a> .
Statistical	AVERAGEA	<code>AVERAGEA(value1, [value2, ...])</code>	Returns the numerical average value in a dataset. <a href="#">Learn more</a>
Statistical	AVERAGEIF	<code>AVERAGEIF(criteria_range, criterion, [average_range])</code>	Returns the average of a range depending on

Functions and formulas

- [Add formulas & functions](#)
- [See the sum & average](#)
- [Reference data from other sheets](#)
- [Google Sheets function list](#)

# 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	<code>ARABIC(roman_numeral)</code>	Computes the value of a Roman numeral. <a href="#">Learn more</a>
Text	ASC	<code>ASC(text)</code>	Converts full-width ASCII and katakana characters to their half-width counterparts. All standard-width characters will remain unchanged. <a href="#">Learn more</a>
Text	CHAR	<code>CHAR(table_number)</code>	Convert a number into a character according to the current Unicode table. <a href="#">Learn more</a>
Text	CLEAN	<code>CLEAN(text)</code>	Returns the text with the non-printable ASCII characters removed. <a href="#">Learn more</a>

Functions and formulas

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

support.google.com/docs/answer/3093281

# 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.

**Google**

- [Google Sheets function list](#)
- [ARRAYFORMULA](#)
- [DETECTLANGUAGE](#)
- [GOOGLEFINANCE](#)
- [GOOGLETRANSLATE](#)
- [IMAGE](#)
- [QUERY function](#)
- [SPARKLINE](#)

 Visit the Learning Center

Using Google products, like Google Docs, at work or school? Try powerful tips, tutorials, and templates. Learn to work on Office files without installing Office, create

support.google.com/docs/answer/3093342

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

# 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](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.

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z\_

mo-dice

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

A1 | fx | =NOW()

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

100% | \$ % .0 .00 123 | Default (Ari... | 36 | B I S A | ⚡田 ⚡ | ⚡ | ...

A1 | fx | =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

B1 | fx | =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

A1 | fx | =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

A screenshot of a Google Sheets spreadsheet titled "mo-dice". The spreadsheet has a single sheet named "Sheet1". The "Tools" menu is open, displaying various options: "Create a form", "AppSheet", "Script editor", "Macros", "Spelling", "Enable autocomplete" (which is checked), "Notification rules", "Protect sheet", and "Accessibility settings". The "Tools" tab is highlighted in green. The "Share" button is visible in the top right corner. The spreadsheet area shows rows 1 through 18 and columns A through H. Cell A1 is selected and contains the formula "fx".

mo-dice

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

Create a form

AppSheet

Script editor

Macros

Spelling

✓ Enable autocomplete

Notification rules

Protect sheet

Accessibility settings

A1 fx

A B C D E F G H

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

Sheet1

The screenshot shows the Google Apps Script editor interface. The title bar indicates the project is named "mo-dice-spreadsheet". The left sidebar has icons for Files, Recent, Libraries, and Services, with "core.gs" selected. The main area displays the following code:

```
1 const R = (n) => parseInt(Math.random() * n);
2
3 class Core {
4   constructor(options) {
5     options = options || {};
6
7     this.symbols = options.symbols || Core.symbols;
8     this.lastRoll =
9       typeof options.lastRoll === "number"
10      ? // 0 < options.lastRoll < this.symbols.length
11        Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
12      : R(this.symbols.length);
13     this.history = options.history || [];
14   }
15
16   roll() {
17     this.lastRoll = R(this.symbols.length);
18     this.history = [this.lastRoll, ...this.history].slice(0, 10);
19   }
20
21   pretty() {
22     return {
23       lastRoll: this.symbols[this.lastRoll],
24       history: this.history.map((e) => this.symbols[e]),
25     };
26   }
27 }
```

script.google.com/home/projects/1G6vFgPfFzCXtNIGD

Apps Script mo-dice-spreadsheet Deploy ?

Files + ↺ ↻ ⚡ Run ⚡ Debug R ▾ Execution log Use legacy editor

roll.gs

core.gs

Libraries +

Services +

```
17 |     this.lastRoll = R(this.symbols.length);
18 |     this.history = [this.lastRoll, ...this.history].slice(0, 10);
19 |
20 |
21 |     pretty() {
22 |         return {
23 |             lastRoll: this.symbols[this.lastRoll],
24 |             history: this.history.map((e) => this.symbols[e]),
25 |         };
26 |
27 |
28 |         clear() {
29 |             this.history = [];
30 |         }
31     }
32
33 Core.symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
34
35 export default Core;
```

Syntax error: SyntaxError: Unexpected token 'export' line: 35 file: core.gs

Copy

The screenshot shows the Google Apps Script editor interface. The title bar indicates the project is named "mo-dice-spreadsheet". The left sidebar contains icons for "Files" (selected), "Deploy", "Help", and "Grid". The "Files" section lists "roll.gs" (selected), "core.gs", and "Libraries/Services". The main editor area displays the code for the "ROLL" function:

```
1  /**
2   * Rolls a die with custom faces.
3   *
4   * @param {string|Array<Array<string>>} faces The value or range of cells
5   *        to use as a die faces.
6   * @return A die roll.
7   * @customfunction
8   */
9  function ROLL(faces) {
10    let symbols;
11
12    if (Array.isArray(faces)) {
13      symbols = faces.reduce((acc, row) => [...acc, ...row], []).filter(Boolean);
14    } else if (faces) {
15      symbols = faces.toString().split("");
16    }
17
18    const core = new Core({
19      symbols,
20    });
21    core.roll();
22    return core.pretty().lastRoll;
23  }
24
```

docs.google.com/spreadsheets/d/17V-54B3mZuSb7H0z\_

mo-dice

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

A1 | fx | =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

A1 | fx | =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

A1 | fx | =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

B C D E F

a v  
cr i  
         p t

Sheet1

2018

docs.google.com/spreadsheets/d/1caViZcwBM\_jbN\_qz

**budget**

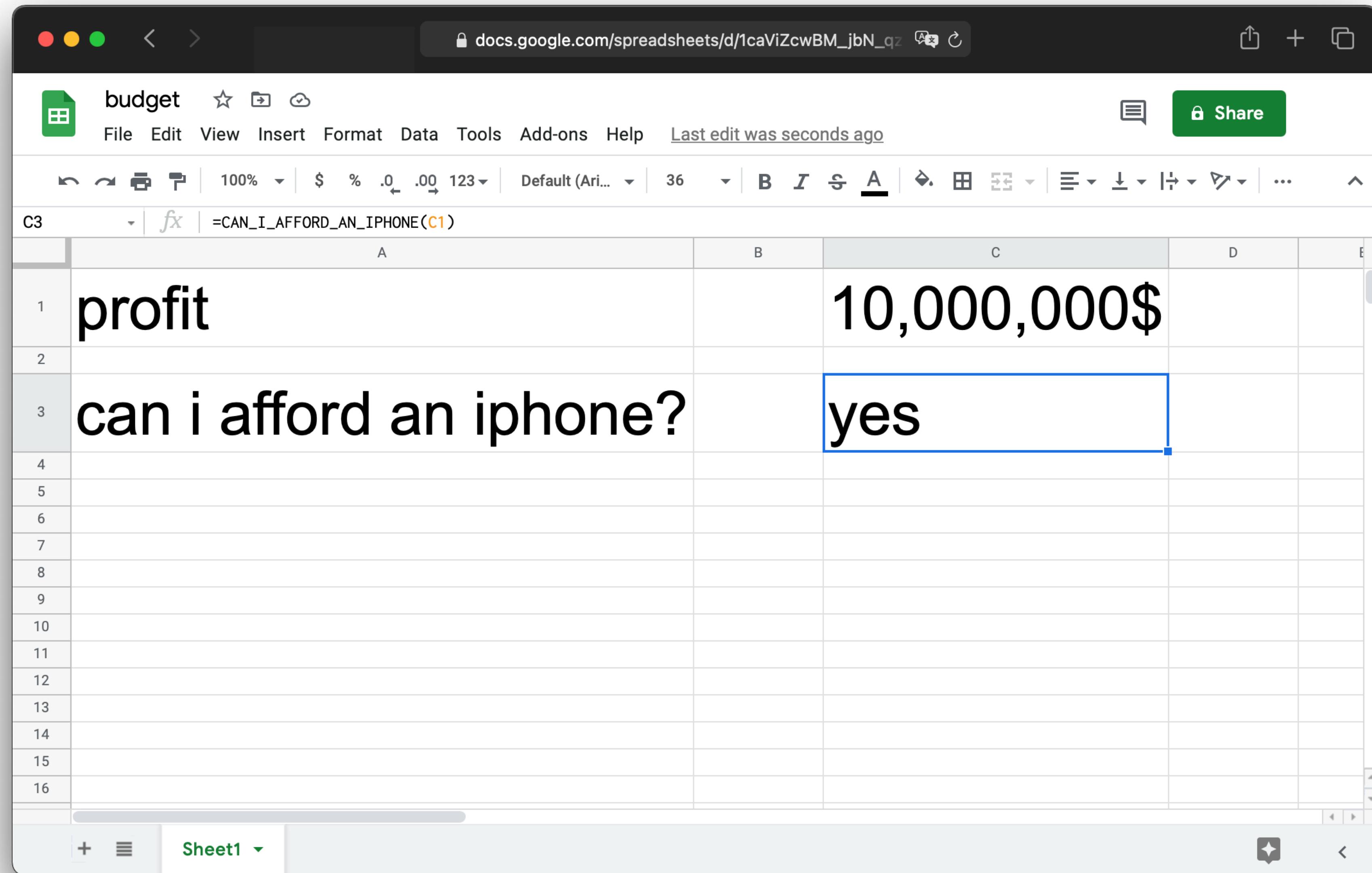
File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

100% | \$ % .0 .00 123 Default (Ari... 36 B I S A

C3 | fx |

	A	B	C	D	E
1	profit		10,000,000\$		
2					
3	can i afford an iphone?				
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

+ Sheet1 star <

A screenshot of a Google Sheets document titled "budget". The document contains a single cell with the formula `=CAN_I_AFFORD_AN_IPHONE(C1)`. The cell displays the result "yes", which is highlighted with a blue border. The spreadsheet has a single sheet named "Sheet1".

	A	B	C	D	E
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					

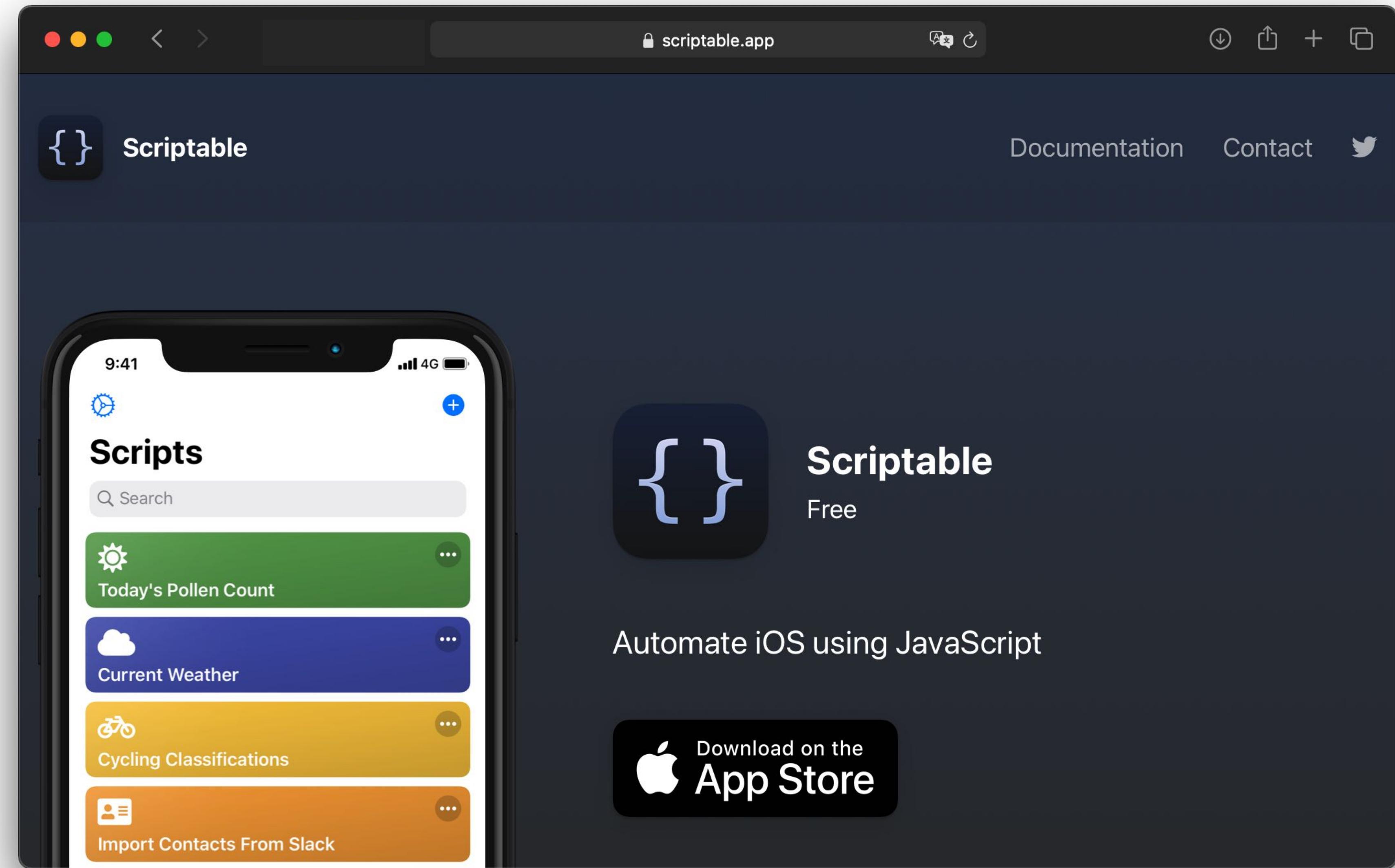
The screenshot shows a Mac OS X desktop environment with a window open to the Apple Developer documentation website at [developer.apple.com/documentation/javascriptcore](https://developer.apple.com/documentation/javascriptcore). The window title bar includes standard Mac OS X controls (red, yellow, green buttons, back, forward, search, and others). The main content area is the Apple Developer site, specifically the **JavaScriptCore** framework page.

The navigation bar at the top of the developer site includes links for **Discover**, **Design**, **Develop**, **Distribute**, **Support**, **Account**, and a search icon. Below the navigation bar, the breadcrumb trail shows **Documentation > JavaScriptCore**. On the right side of the header, there are dropdown menus for **Language: Swift** and **API Changes: Show**.

The main content area has a heading **Framework** followed by the title **JavaScriptCore**. A descriptive paragraph states: "Evaluate JavaScript programs from within an app, and support JavaScript scripting of your app." To the right of this text, under the heading **Availability**, are listed system requirements: iOS 7.0+, macOS 10.5+, Mac Catalyst 13.0+, and tvOS 9.0+.

A horizontal line separates the main content from the **Overview** section. The **Overview** section contains a paragraph explaining the purpose of the framework: "The JavaScriptCore framework provides the ability to evaluate JavaScript programs from within Swift, Objective-C, and C-based apps. You can also use JavaScriptCore to insert custom objects into the JavaScript environment."

To the right of the **Overview** section, under the heading **On This Page**, are two links: [Overview](#) and [Topics](#).



The screenshot shows a terminal window with a dark theme. The title bar reads "core.js — scriptable — mo-dice". The window contains the following code:

```
1 const R = (n) => parseInt(Math.random() * n);
2
3 class Core {
4     static symbols = ["\u2680", "\u2681", "\u2682", "\u2683", "\u2684", "\u2685"];
5
6     constructor(options) {
7         options = options || {};
8
9         this.symbols = options.symbols || Core.symbols;
10        this.lastRoll =
11            typeof options.lastRoll === "number"
12            ? // 0 < options.lastRoll < this.symbols.length
13                Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
14            : R(this.symbols.length);
15        this.history = options.history || [];
16    }
17
18    roll() {
19        this.lastRoll = R(this.symbols.length);
20        this.history = [this.lastRoll, ...this.history].slice(0, 10);
21    }
22
23    pretty() {
24        return {
25            lastRoll: this.symbols[this.lastRoll],
26        };
27    }
28}
```

The code defines a `Core` class with methods `roll()` and `pretty()`. It uses a static array `symbols` containing six dice faces. The `roll()` method updates the `lastRoll` property and the `history` array. The `pretty()` method returns an object with `lastRoll` set to the current value of `lastRoll`.

core.js — scriptable — mo-dice

```
12     ? // 0 < options.lastRoll < this.symbols.length
13         Math.max(0, Math.min(options.lastRoll, this.symbols.length - 1))
14     : R(this.symbols.length);
15     this.history = options.history || [];
16 }
17
18 roll() {
19     this.lastRoll = R(this.symbols.length);
20     this.history = [this.lastRoll, ...this.history].slice(0, 10);
21 }
22
23 pretty() {
24     return {
25         lastRoll: this.symbols[this.lastRoll],
26         history: this.history.map((e) => this.symbols[e]),
27     };
28 }
29
30 clear() {
31     this.history = [];
32 }
33 }
34
35 module.exports = Core;
36
```

22 characters selected    main (3)    Spaces: 2    JavaScript

```
modice.siri.js — mo-dice
1 const Core = importModule("core.js");
2
3 const fileManager = FileManager.local();
4 const cachePath = fileManager.joinPath(
5   fileManager.documentsDirectory(), "modice.json"
6 );
7
8 function load() {
9   const symbols = [
10     "единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
11   ];
12
13   if (fileManager.fileExists(cachePath)) {
14     try {
15       return new Core({
16         ...JSON.parse(fileManager.readString(cachePath)),
17         symbols,
18       });
19     } catch (e) {
20       console.error(e);
21     }
22   }
23   return new Core({ symbols });
24 }
25
```

37 characters selected      main [4]      Spaces: 2      JavaScript

```
modice.siri.js — mo-dice
1 const Core = importModule("core.js");
2
3 const fileManager = FileManager.local();
4 const cachePath = fileManager.joinPath(
5   fileManager.documentsDirectory(), "modice.json"
6 );
7
8 function load() {
9   const symbols = [
10     "единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
11   ];
12
13   if (fileManager.fileExists(cachePath)) {
14     try {
15       return new Core({
16         ...JSON.parse(fileManager.readString(cachePath)),
17         symbols,
18       });
19     } catch (e) {
20       console.error(e);
21     }
22   }
23   return new Core({ symbols });
24 }
25
```

4 lines, 133 characters selected

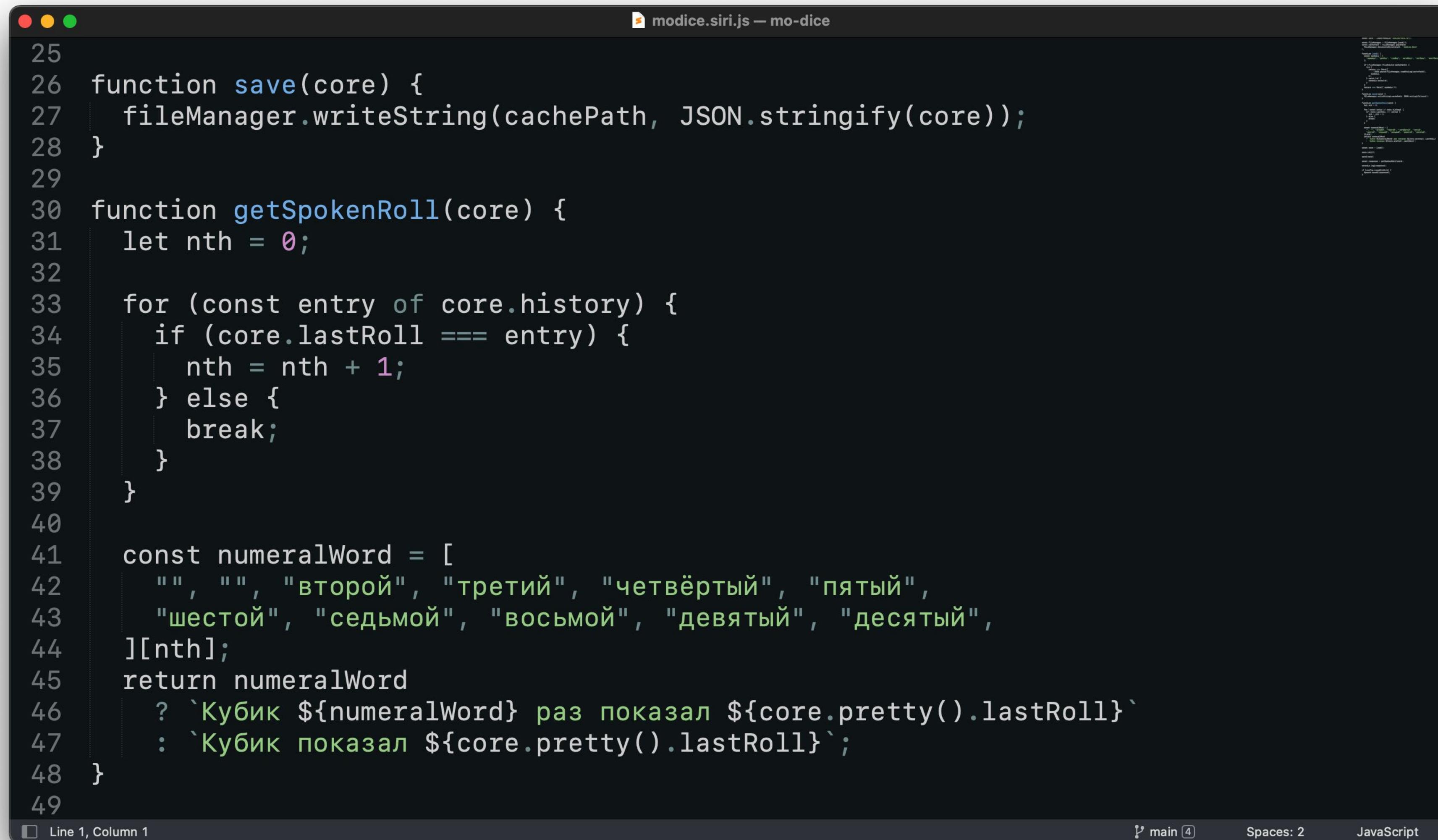
main [4] Spaces: 2 JavaScript

A screenshot of a code editor window titled "modice.siri.js — mo-dice". The code is written in JavaScript and defines a function `load` that reads symbols from a JSON file in local storage.

```
1 const Core = importModule("core.js");
2
3 const fileManager = FileManager.local();
4 const cachePath = fileManager.joinPath(
5   fileManager.documentsDirectory(), "modice.json"
6 );
7
8 function load() {
9   const symbols = [
10     "единицу", "двойку", "тройку", "четвёрку", "пятёрку", "шестёрку",
11   ];
12
13   if (fileManager.fileExists(cachePath)) {
14     try {
15       return new Core({
16         ...JSON.parse(fileManager.readString(cachePath)),
17         symbols,
18       });
19     } catch (e) {
20       console.error(e);
21     }
22   }
23   return new Core({ symbols });
24 }
25
```

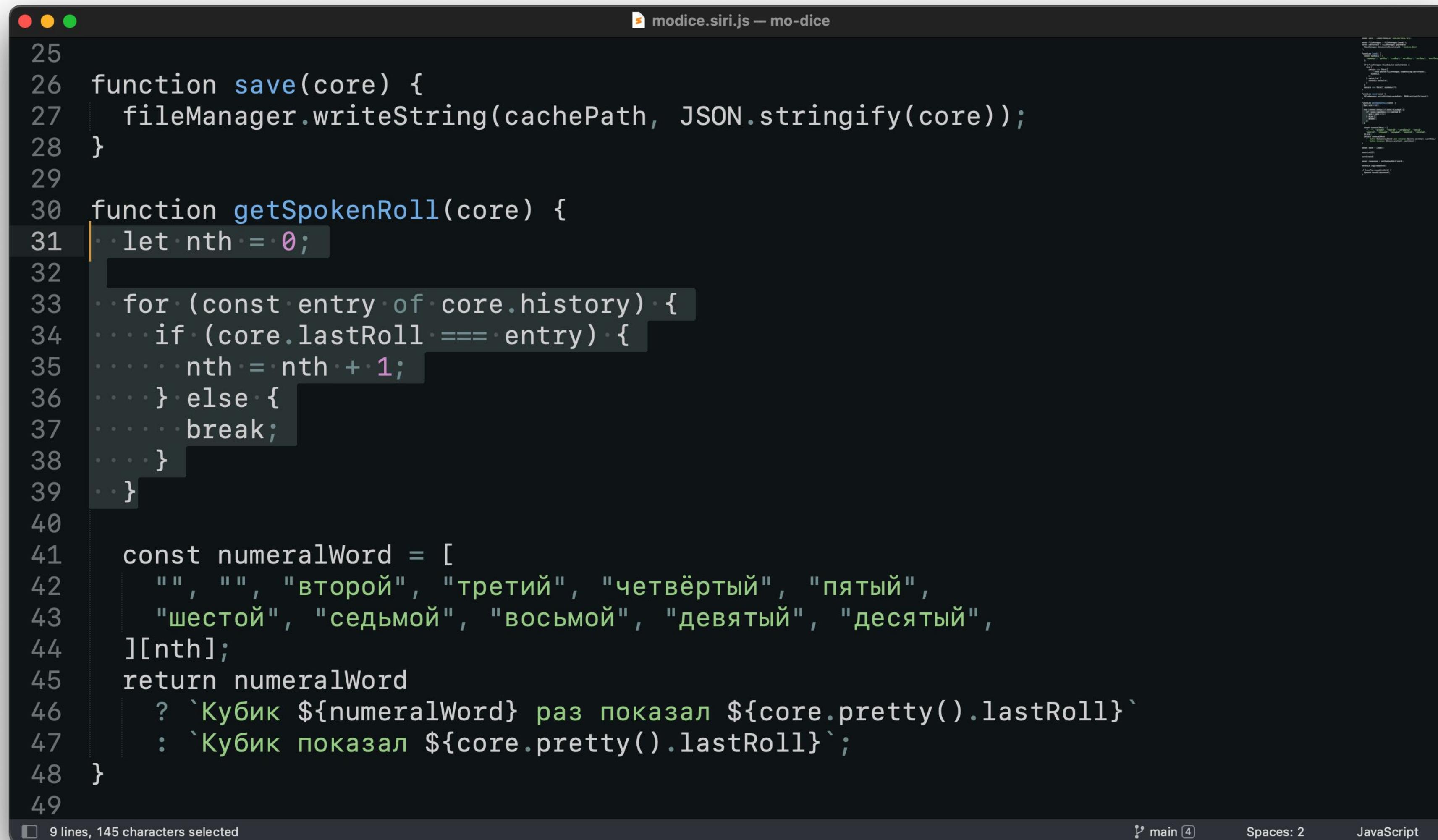
The code editor interface includes a status bar at the bottom with the following information:

- 3 lines, 94 characters selected
- main [4] (with a dropdown arrow)
- 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



A screenshot of a macOS terminal window titled "modice.siri.js — mo-dice". The window displays a block of JavaScript code. The code includes functions for saving core data and generating spoken rolls, utilizing a numeral word array. The code is annotated with line numbers from 25 to 49. A selection bar highlights lines 31 through 39. The status bar at the bottom shows "9 lines, 145 characters selected".

```
25 function save(core) {
26   fileManager.writeString(cachePath, JSON.stringify(core));
27 }
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
```

9 lines, 145 characters selected

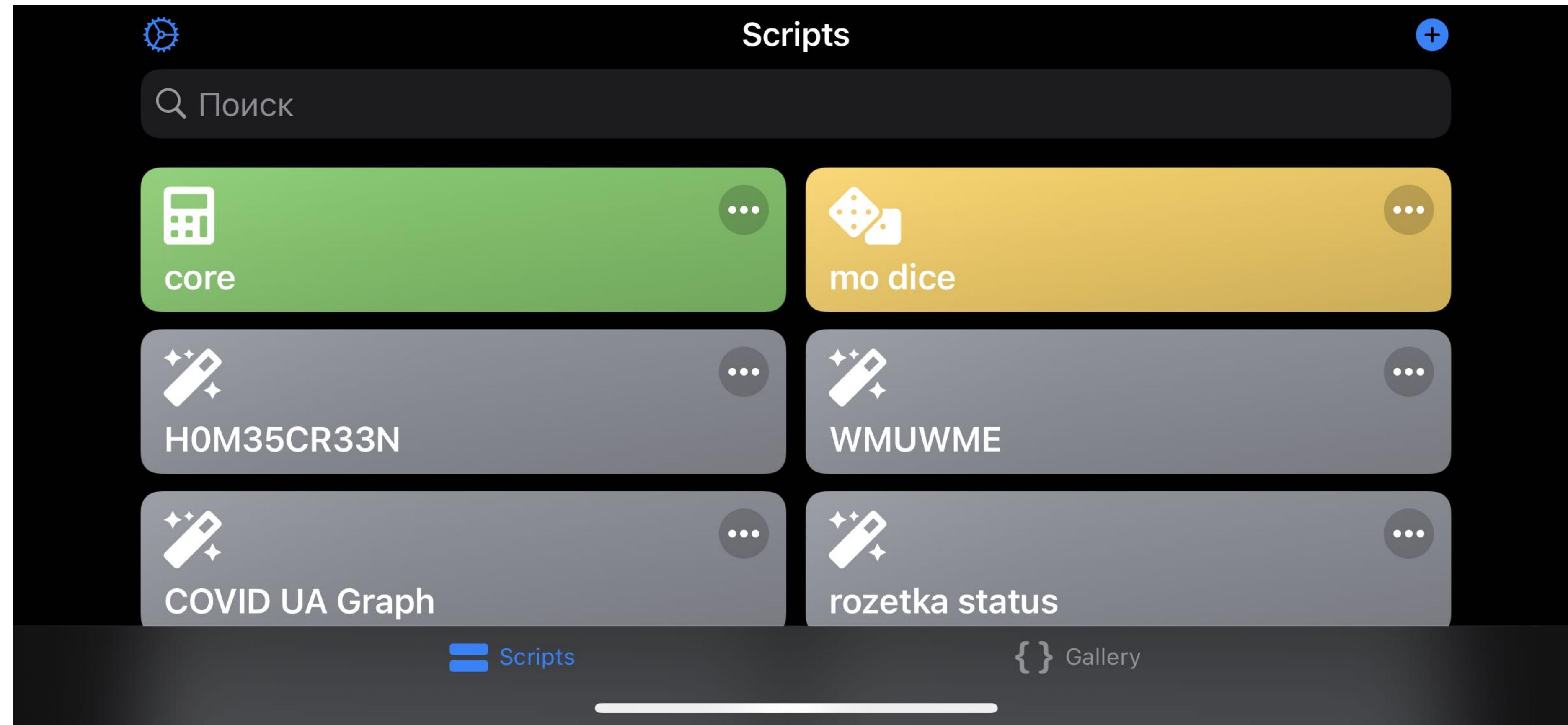
main [4] Spaces: 2 JavaScript

```
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
```

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



**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

```
1 const
2
3 const
4 const
5 file
6 );
7
8 function
9   cor
10   Пожалуйста >
11 ];
12
13 if
14   try {
15
16
17
18
19   } catch (e) {
20     console.error(e);
```



Run Script



Done

{ } SCRIPTABLE

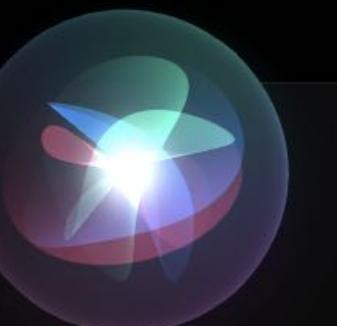
```
1 const  
2  
3 const  
4 const  
5 fil  
6 );  
7  
8 funct  
9 cor  
10 ,  
11 ];  
12  
13 if  
14 try {
```



Run Script

Пожалуйста >

Хорошо, готово. Кубик второй раз показал двойку



```
19 } catch (e) {  
20     console.error(e);
```



everywhere

A screenshot of a web browser window showing the Node.js homepage at [nodejs.org/en/](https://nodejs.org/en/). The browser interface includes standard controls (red, yellow, green buttons, back/forward arrows, search bar, tabs, and a menu icon) at the top.

The Node.js logo is prominently displayed at the top center. Below it is a navigation bar with links: HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS. To the right of the navigation bar is a small icon with a 'G' and a downward arrow.

The main content area features a large heading: "Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#)". Below this, there is a section titled "Download for macOS (x64)" with two prominent buttons:

- 14.17.0 LTS** (Recommended For Most Users)
- 16.2.0 Current** (Latest Features)

Below each button are links to "Other Downloads", "Changelog", and "API Docs".

Further down the page, there is a link to the "Long Term Support (LTS) schedule".

The footer contains the OpenJS Foundation logo, copyright information ("© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent."), trademark information ("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."), and links for reporting issues and getting help.

The image shows a Mac desktop with two browser windows open. The left window displays the Cypress.io homepage, featuring a large white banner with the text "The web has changed. Finally, testing has too." and "Fast, easy and reliable testing for anything". It includes a terminal-like interface with "\$ npm install cypress" and a download button. The right window shows the Puppeteer documentation on pptr.dev, with a sidebar containing links like API, Outline, Release Notes, Overview, puppeteer vs puppeteer-core, Environment Variables, Working with Chrome Extensions interface, CustomQueryHandler, Puppeteer, BrowserFetcher, Browser, BrowserContext, and Page. The main content area features a large heading "Puppeteer", a status bar with "run-checks passing npm v9.1.1", and a yellow box describing Puppeteer as a Node library for controlling Chrome or Chromium via the DevTools Protocol. It also lists what can be done with Puppeteer.

**Cypress**

Features How it works Dashboard Pricing Support Blog Docs Login Sign up GitHub

# The web has changed. Finally, testing has too.

Fast, easy and reliable testing for anything.

```
$ npm install cypress
```

or [Download now](#)

Install Cypress for Mac, Linux, or Windows,

[\\$ npm install cypress](#)

[Download now](#)

Install Cypress for Mac, Linux, or Windows,

**Puppeteer**

run-checks passing npm v9.1.1

[API](#) | [FAQ](#) | [Contributing](#) | [Troubleshooting](#)

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/](https://addons.mozilla.org/en-US/developers/)

**Add-on Developer Hub** Extension Workshop Documentation Support Blog

# Customize Firefox

Whether you're new to extension development, polishing up or porting an existing extension, creating a custom enterprise solution, or getting support.

[Learn how to make an extension](#)

## 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/](https://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, iPad, and iPhone with Safari Extensions. Learn more about what's available.

 [developer.chrome.com/docs/extensions/](https://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

**abiliojr / sqlite-js**

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**

- **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

The screenshot shows a web browser window displaying the Cloudflare Workers homepage at [workers.cloudflare.com](https://workers.cloudflare.com). The page features a dark header with the Cloudflare logo and the word "Workers". Below the header, there's a navigation bar with links for "Home", "Pages", "Built with", "Docs", and "Discord", along with "Log in" and "Sign up" buttons. The main content area has a large heading "You write code. We handle the rest." followed by a subtext: "Deploy serverless code instantly across the globe to give it exceptional performance, reliability, and scale." Below this are two buttons: "Start building" (orange) and "Read docs" (light gray). To the right, there's a code block showing the steps to install Wrangler and publish a "Hello World" worker:

```
# 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
```

At the bottom left, there's a list of three orange bullet points:

- From signup to globally deployed in <5min
- Your code runs within **milliseconds** of your users worldwide
- Say goodbye to cold starts—support for **Oms worldwide NEW**

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

# Awesome Demos Done Quick

The screenshot shows a Sublime Text window with a presentation slide titled "Awesome Demos Done Quick". The slide features a background image of a pine tree branch against a blue sky. The text on the slide includes "#A#", "#D# #D#", "#Q#", "with JXA", "Awesome Demos Done Quick", "Anton Berinov @zemlanin", and "BeerJS Summit Minsk 2018". Below the slide, a terminal window displays the command `./present.js` being run in a directory named "click-yourself". The terminal output shows "TASBot" and "Introduction". The Sublime Text status bar indicates "Line 13, Column 9" and "Plain Text".

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/QEZON0rrbL0?t=...](https://youtu.be/QEZON0rrbL0?t=...)

Score: 4  
Lives: 2

!!Con 2020

1 37 119 I ⬆

Omar 🚛 @rsnous · May 11, 2020  
full source code (and working PDF!) at [github.com/osnr/horrifying-pdf-experiments](https://github.com/osnr/horrifying-pdf-experiments)

### 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 Documentation Support in YouTube Search...

## Contents

- Features
- Powering Pixl.js
  - Vin pins
- Power Consumption
- Resetting Pixl.js
- Hard Reset
- Tutorials
- Pinout
- Arduino Shields
  - Shield power
  - Shields
- Information
- LCD Screen
  - Graphics
  - Screen updates
  - Menus
  - Contrast
  - Terminal
- Power Usage
- Splash Screen

## 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

The image shows two separate browser windows side-by-side, each displaying a different embeddable JavaScript engine.

**Duktape (Top Window):**

- Title Bar:** ((o) Duktape
- Header:** Home Download API Guide Wiki REPL
- Section:** Duktape
- Text:** Duktape is an embeddable Javascript engine, with a focus on portability and compact footprint.
- Text:** 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.
- Section:** Main features

**QuickJS Javascript Engine (Bottom Window):**

- Title Bar:** bellard.org/quickjs/
- Header:** Fork
- Section:** QuickJS Javascript Engine
- Section:** News
- List:**
  - 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>
- Section:** Introduction
- Text:** 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.
- Section:** Main Features:
- List:**
  - 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?**

シク

ジク

ジク

JS

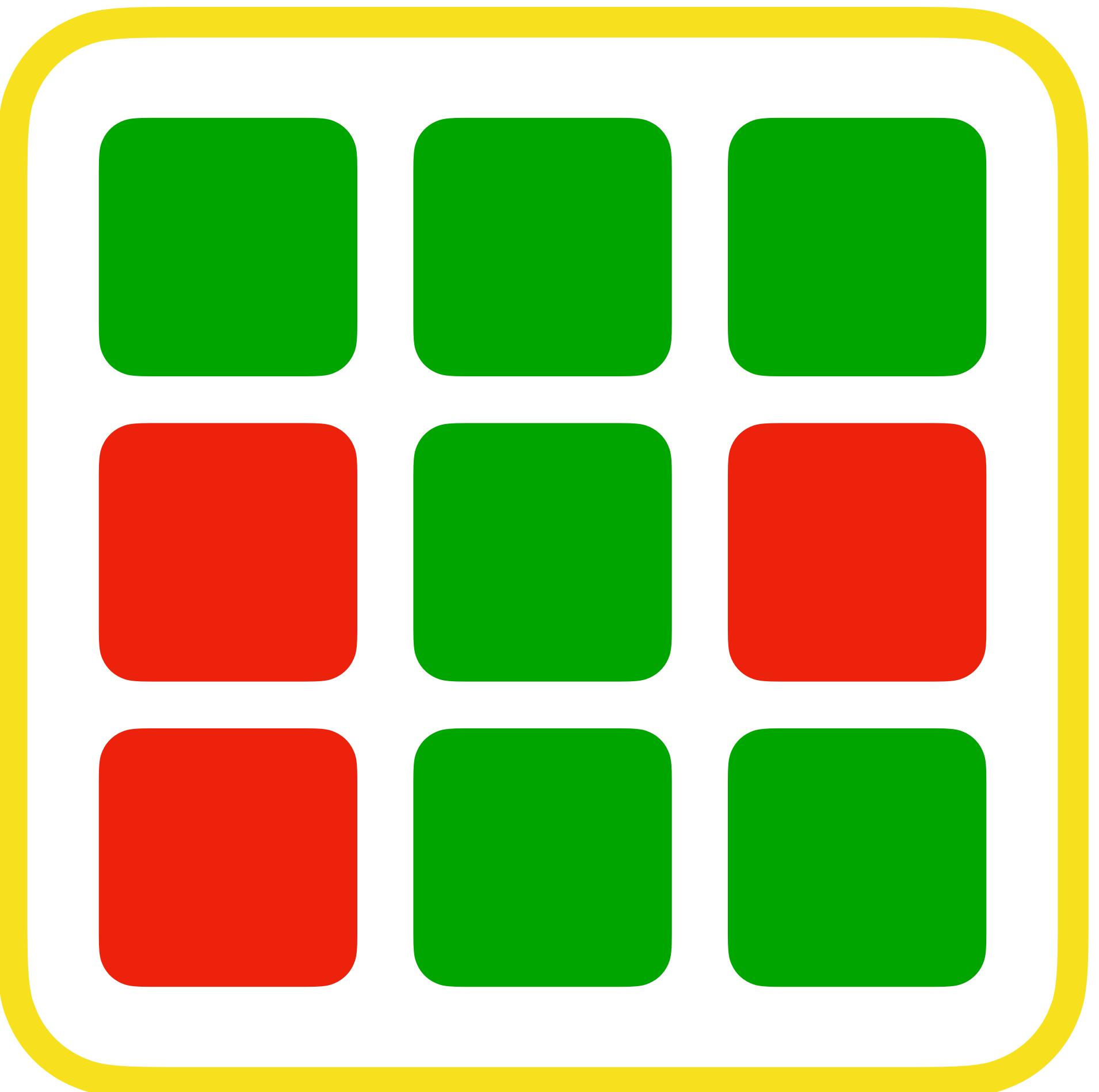
JS

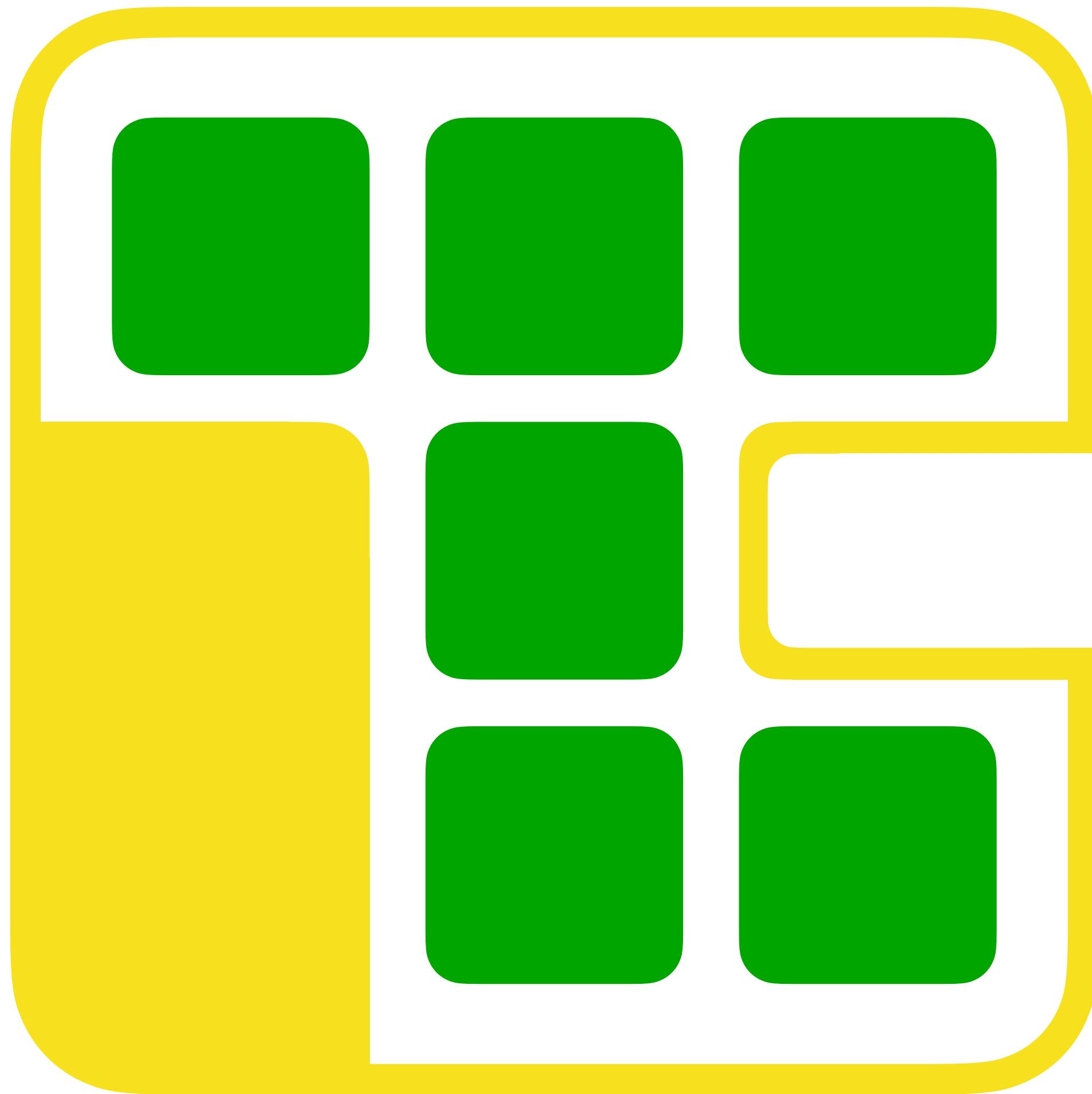
ЖС

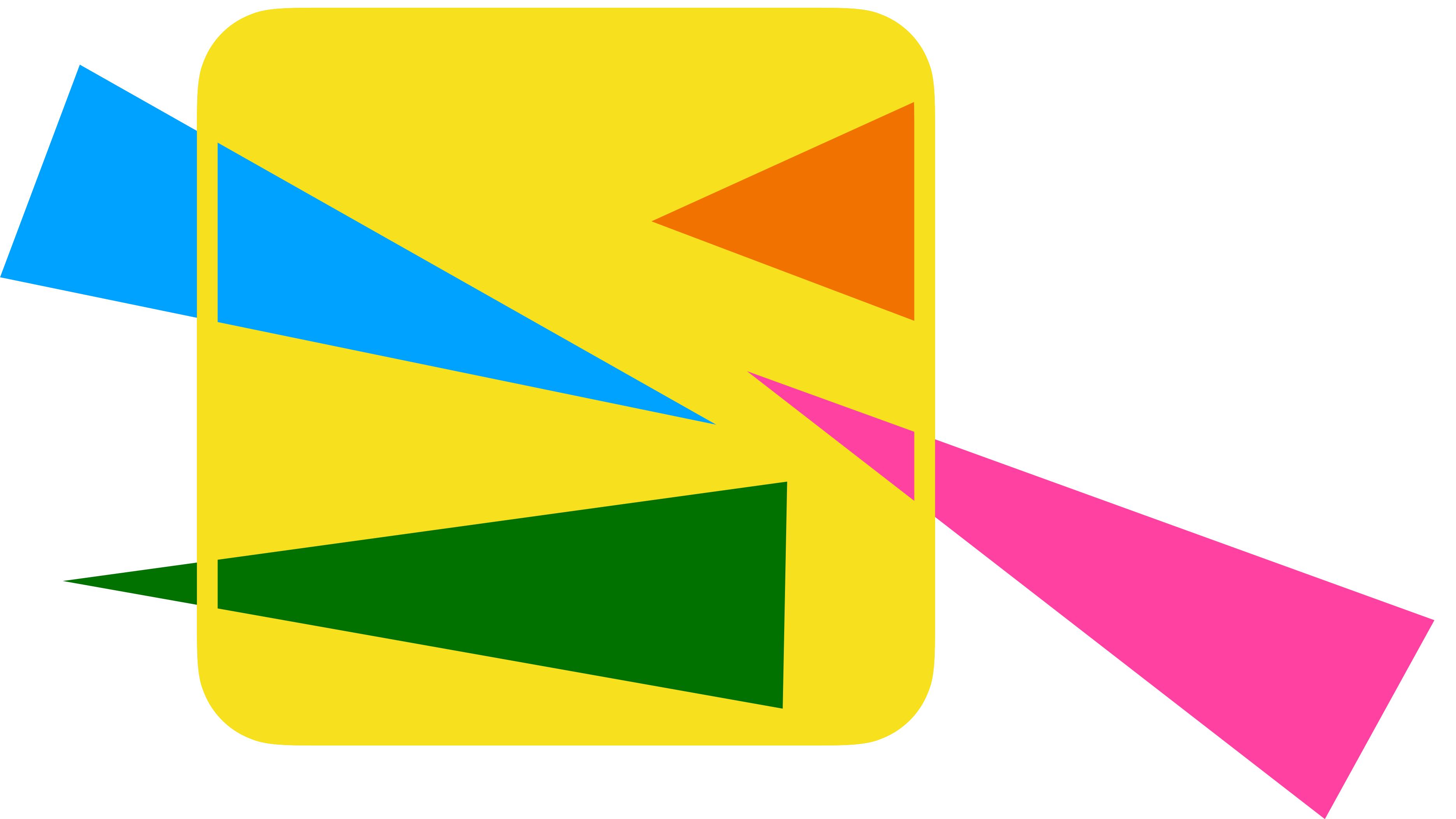
СС

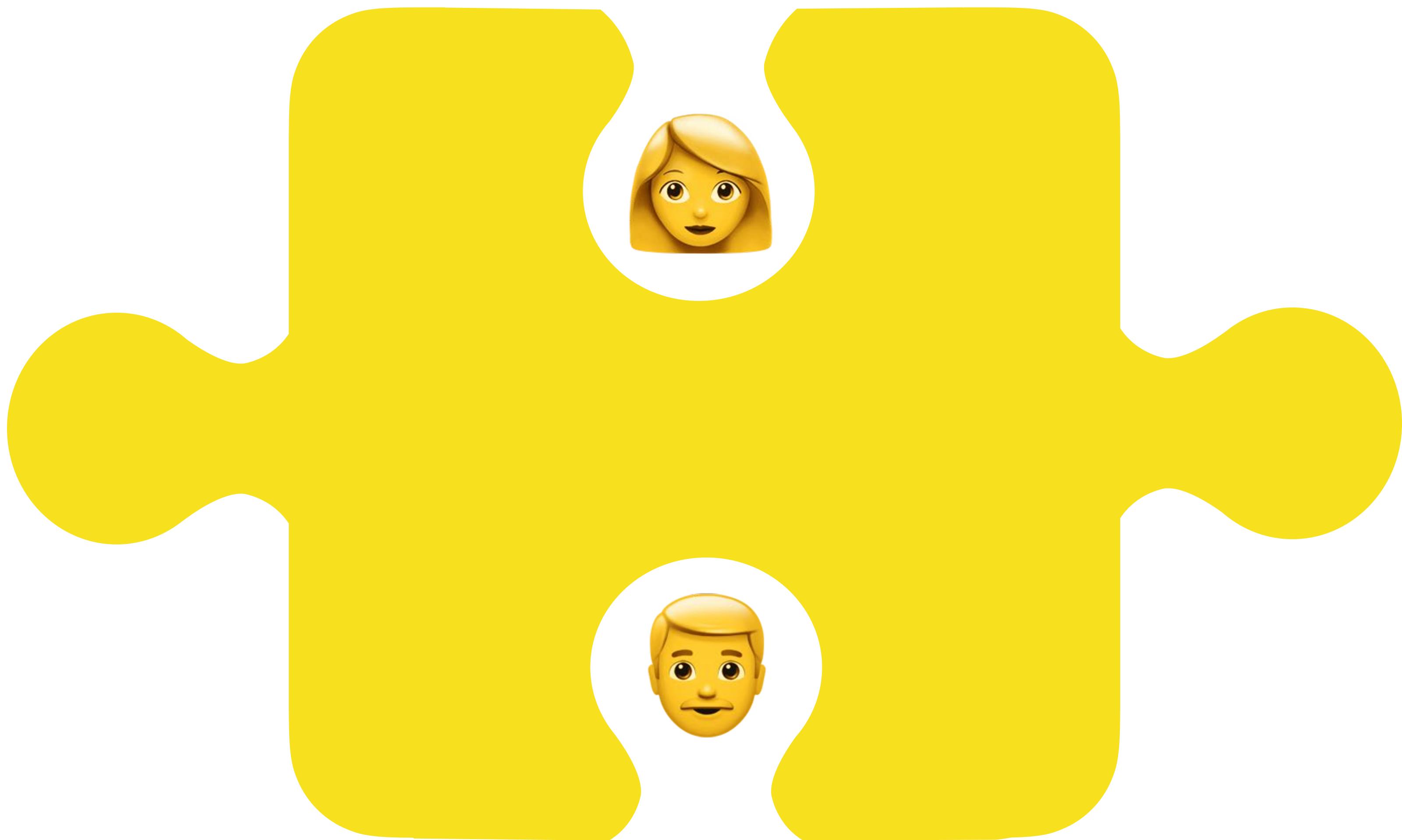
СС

ЖС











**anton.click/fwdays**

**@zemlanin**