

BOOKMARKLETS

Simplify Work with JavaScript

By Veronica Eulenberg

JavaScript is my favorite language.

Because I'm good at it.

I know you probably hate JavaScript, but by the end of this, you might hate it a little less :)

STORY TIME!

1. Tedious QA queue
2. Clicking the same sequence
3. Annoying elements everywhere

WHAT IS A BOOKMARKLET?

- Small JavaScript program saved as a bookmark
- Solves repetitive tasks with a single click
- Customized actions for specific websites

WHY USE A BOOKMARKLET?

- Web scraping: Quickly extract info
- Productivity: Automate repetitive web tasks
- Customization: Improve usability with CSS changes

You probably hate CSS as well.

Yes, CSS is also my favorite.
Because I'm good at it.

WHEN TO USE A BOOKMARKLET?

- Waiting in queue and want to easily track your place
- Repeatedly clicking sequence you want automated
- Annoyed by page elements and want them removed

HOW TO CREATE A BOOKMARKLET:

1. Write JavaScript code (or ask AI to generate it)
2. Create a new browser bookmark
3. Paste the code into the **URL field**
4. Click the bookmark on any page to run your script

SIMPLE TEST

```
1 javascript:(function(){  
2     alert("Hello, world!");  
3 })();
```

Let's start with a simple hello world :)

WHAT IT LOOKS LIKE IRL:

Edit bookmark

Name	Hello World
URL	javascript:(function(){alert('Hello, world!');})();

THINGS TO KEEP IN MIND:

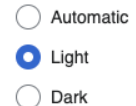
1. Modern browsers limit url size to 2k-8k characters
2. Test your code in the browser console
3. Comments need to be multi-line (block comments)
4. Always use semicolons to avoid potential errors

```
1 javascript:(function(){
2     // I will break your code >:)
3     alert("I will also break your code!!")
4
5     /* I'm a nice block comment! */
6     alert("I have a semicolon.");
7 })();
```

OR ELSE

```
Uncaught SyntaxError: Unexpected end of input
```

EXAMPLES!





DARK MODE TOGGLE



[Search](#)

[Donate](#) [Create account](#) [Log in](#) ...

Bookmarklet

🌐 17 languages

Contents

hide

(Top)

History

Concept

Usage

Installation

Method 1: Creating a New Bookmark

Method 2: Dragging and Dropping

Running the Bookmarklet

Tips

Example

References

External links

Article

Talk

Read

Edit

View history

Tools

Appearance

hide

Text

Small

Standard

Large

Width

Standard

Wide

Color (beta)

Automatic

Light

Dark

From Wikipedia, the free encyclopedia

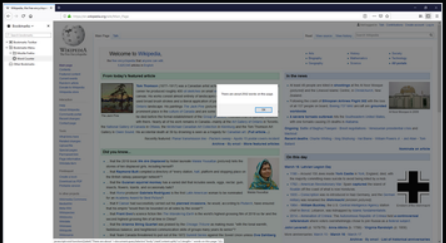
A **bookmarklet** is a **bookmark** stored in a **web browser** that contains JavaScript commands that add new features to the browser. They are stored as the **URL** of a bookmark in a **web browser** or as a **hyperlink** on a **web page**. Bookmarklets are usually small snippets of **JavaScript** executed when user clicks on them. When clicked, bookmarklets can perform a wide variety of operations, such as running a search query from selected text or extracting data from a table.

Another name for *bookmarklet* is **favelet** or **favlet**, derived from *favorites* (synonym of bookmark).^[1]

History

[edit]

Steve Kangas of bookmarklets.com coined the word *bookmarklet*^[2] when he started to create short scripts based on a suggestion in [Netscape's JavaScript guide](#).^[3] Before that, [Tantek Çelik](#) called these scripts *favelets* and used that word as early as on 6 September 2001 (personal email^[clarification needed]). [Brendan Eich](#), who developed JavaScript at Netscape, gave this account of the origin of bookmarklets:



Demonstration of a bookmarklet that counts the number of words on the page. The browser shown is Firefox 65.0.2 running on Windows 10.

Turn any website into dark mode.
Or if you hate dark mode, turn it light!

```
1 javascript:(function(){
2   let css = document.getElementById('darkModeToggle');
3
4   if (!css) {
5     css = document.createElement('style');
6     css.id = 'darkModeToggle';
7     css.textContent = 'html, img, video { filter: invert(1)';
8     document.head.appendChild(css);
9   } else {
10    css.remove();
11  }
12 })();
```

Turn any website into dark mode.
Or if you hate dark mode, turn it light!

```
1 javascript:(function(){
2   let css = document.getElementById('darkModeToggle');
3
4   if (!css) {
5     css = document.createElement('style');
6     css.id = 'darkModeToggle';
7     css.textContent = 'html, img, video { filter: invert(1)';
8     document.head.appendChild(css);
9   } else {
10    css.remove();
11  }
12 })();
```

Turn any website into dark mode.
Or if you hate dark mode, turn it light!

```
1 javascript:(function(){
2   let css = document.getElementById('darkModeToggle');
3
4   if (!css) {
5     css = document.createElement('style');
6     css.id = 'darkModeToggle';
7     css.textContent = 'html, img, video { filter: invert(1)';
8     document.head.appendChild(css);
9   } else {
10    css.remove();
11  }
12 })();
```

Turn any website into dark mode.
Or if you hate dark mode, turn it light!

```
1 javascript:(function(){
2   let css = document.getElementById('darkModeToggle');
3
4   if (!css) {
5     css = document.createElement('style');
6     css.id = 'darkModeToggle';
7     css.textContent = 'html, img, video { filter: invert(1)';
8     document.head.appendChild(css);
9   } else {
10    css.remove();
11  }
12 })();
```



YOUTUBE THEATER MODE

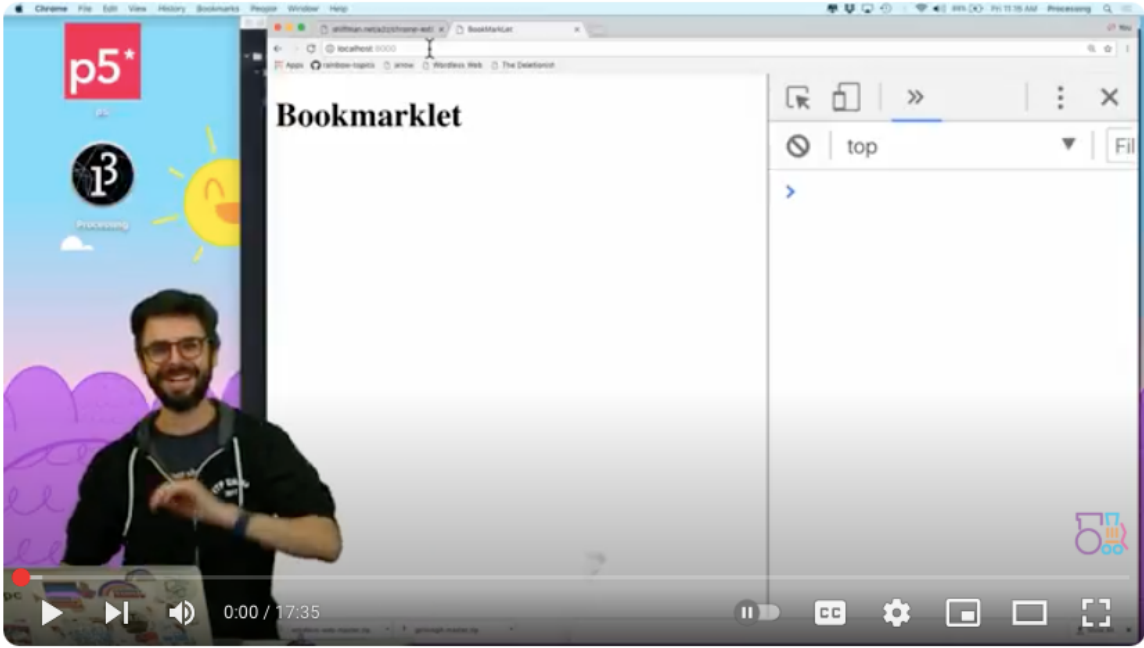


Search



+ Create





11.2: Bookmarklets - Programming with Text

The Coding Train 1.74M subscribers

92K views 7 years ago Programming with Text - All Videos

AllFrom the seriesFrom The Coding Train



11.3: Chrome Extensions: Content Scripts - Programming with Text

The Coding Train 228K views • 7 years ago



ml5.js 1.0 and Guest Conductor Patt Vira

The Coding Train 26K views • 8 months ago



How To Learn Any Skill So Fast It Feels Illegal

Justin Sung 1.8M views • 10 months ago



Programming with Text - All Videos

The Coding Train • Course

View full course



For-Profit (Creative) Software

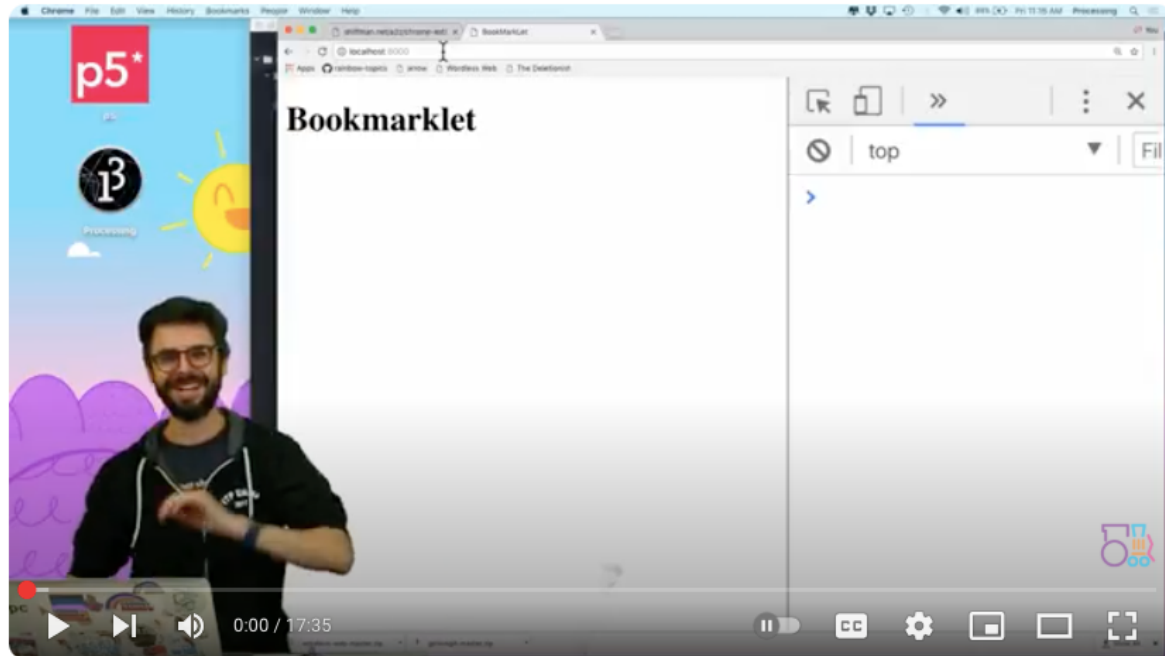
EndVertex 344K views • 2 weeks ago



Upbeat Lo-fi - Deep Focus & Study



YOUTUBE THEATER MODE



Turn up the speed and remove distractions!

```
1 javascript:(function(){  
2     document.querySelector("video").playbackRate = 1.5;  
3     masthead.remove();  
4     secondary.remove();  
5     below.remove();  
6 })();
```



HIGHLIGHT SEARCH TERMS

bookmarklet: 48

javascript: 30

Bookmarklet

🌐 17 languages ▼

Article [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#) ▼

Contents [hide](#)

(Top)

[History](#)

[Concept](#)

▼ [Usage](#)

[Installation](#)

[Method 1: Creating a New Bookmark](#)

[Method 2: Dragging and Dropping](#)

[Running the Bookmarklet](#)

[Tips](#)

[Example](#)

[References](#)

[External links](#)

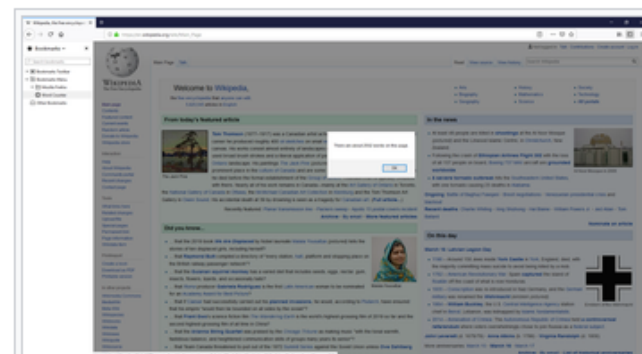
From Wikipedia, the free encyclopedia

A **bookmarklet** is a [bookmark](#) stored in a [web browser](#) that contains [JavaScript](#) commands that add new features to the browser. They are stored as the [URL](#) of a bookmark in a [web browser](#) or as a [hyperlink](#) on a [web page](#). **Bookmarklets** are usually small snippets of [JavaScript](#) executed when user clicks on them. When clicked, bookmarklets can perform a wide variety of operations, such as running a search query from selected text or extracting data from a table.

Another name for **bookmarklet** is **favelet** or **favlet**, derived from *favorites* (synonym of bookmark).^[1]

History [\[edit \]](#)

Steve Kangas of [bookmarklets.com](#) coined the word **bookmarklet**^[2] when he started to create short scripts based on a suggestion in [Netscape's JavaScript](#) guide.^[3] Before that, [Tantek Çelik](#) called these scripts *favelets* and used that word as early as on 6 September 2001 (personal email^[clarification needed]). [Brendan Eich](#), who developed JavaScript at Netscape, gave this account of the origin of bookmarklets:



Demonstration of a **bookmarklet** that counts the number of words on the page. The browser shown is Firefox 65.0.2 running on Windows 10.

Highlights all occurrences of the words you specify.
Automatically navigate through them with ease.

```
1 javascript:(function(){
2   let colors = ['#ffeb3b', '#4caf50', '#2196f3', '#f44336',
3   terms = prompt('Enter search terms:').split(','),
4   widget = document.body.appendChild(Object.assign(document
5
6   terms.forEach((term, i) => {
7     let color = colors[i % colors.length], regex = new RegExp
8
9     document.querySelectorAll('*:not(script,style)').forEach(
10      el.childNodes.forEach(n => n.nodeType === 3 && regex.
11        n.replaceWith(...n.textContent.split(regex).map(t =>
12          t.match(regex) ? Object.assign(document.createElement(
13            className: `term-${i}`, style: `background:${color}
14          }) : document.createTextNode(t)
15        ))
16    })
17  })
18})
```

Highlights all occurrences of the words you specify.
Automatically navigate through them with ease.

```
1 javascript:(function(){
2   let colors = ['#ffeb3b', '#4caf50', '#2196f3', '#f44336',
3   terms = prompt('Enter search terms:').split(','),
4   widget = document.body.appendChild(Object.assign(document
5
6   terms.forEach((term, i) => {
7     let color = colors[i % colors.length], regex = new RegExp
8
9     document.querySelectorAll('*:not(script,style)').forEach
10      el.childNodes.forEach(n => n.nodeType === 3 && regex.
11      n.replaceWith(...n.textContent.split(regex).map(t =>
12        t.match(regex) ? Object.assign(document.createElement
13          className: `term-${i}`, style: `background:${color}
14        }) : document.createTextNode(t)
15      ))
```

Highlights all occurrences of the words you specify.
Automatically navigate through them with ease.

```
5
6  terms.forEach((term, i) => {
7    let color = colors[i % colors.length], regex = new RegExp
8
9    document.querySelectorAll('*:not(script,style)').forEach(
10      el.childNodes.forEach(n => n.nodeType === 3 && regex.
11        n.replaceWith(...n.textContent.split(regex).map(t =>
12          t.match(regex) ? Object.assign(document.createElement(
13            className: `term-${i}`, style: `background:${color}
14            }) : document.createTextNode(t)
15          ))
16        )
17      );
18
19    let matches = document.querySelectorAll(`.term-${i}`),
```

Highlights all occurrences of the words you specify.
Automatically navigate through them with ease.

```
14         }) : document.createTextNode(t)
15     ))
16 )
17 );
18
19     let matches = document.querySelectorAll(`.term-${i}`),
20         btn = widget.appendChild(Object.assign(document.createElement('button'), {
21             style: `background:${color};cursor:pointer;`,
22             textContent: `${term.trim()}: ${matches.length}`
23         }));
24
25     let idx = 0;
26     btn.onclick = () => (
27         matches[idx].style = `background:${color};border:2px solid ${color};`,
28         matches[idx].scrollIntoView({ behavior: 'smooth' }),
```

Highlights all occurrences of the words you specify.
Automatically navigate through them with ease.

```
18
19     let matches = document.querySelectorAll(`.term-${i}`),
20       btn = widget.appendChild(Object.assign(document.createElement('button'), {
21         style: `background:${color};cursor:pointer;`,
22         textContent: `${term.trim()}: ${matches.length}`
23       }));
24
25     let idx = 0;
26     btn.onclick = () => (
27       matches[idx].style = `background:${color};border:2px solid ${color};`,
28       matches[idx].scrollIntoView({ behavior: 'smooth' }),
29       idx = (idx + 1) % matches.length
30     );
31   });
32 }());
```

SETUP

```
2 let colors = ['#ffeb3b', '#4caf50', '#2196f3', '#f44336', '#  
3 terms = prompt('Enter search terms:').split(','),  
4 widget = document.body.appendChild(Object.assign(document.cr
```

LOOP THROUGH TERMS

ADD COLOR TO TEXT ON PAGE

```
7 let color = colors[i % colors.length], regex = new RegExp(`
8
9 document.querySelectorAll('*:not(script,style)').forEach(el
10   el.childNodes.forEach(n => n.nodeType === 3 && regex.test
11     n.replaceWith(...n.textContent.split(regex).map(t =>
12       t.match(regex) ? Object.assign(document.createElement
13         className: `term-${i}`, style: `background:${color}
14       }) : document.createTextNode(t)
15     ))
16   )
17 );
```

CREATE BUTTONS FOR TEXT

```
19 let matches = document.querySelectorAll(`.term-${i}`),  
20 btn = widget.appendChild(Object.assign(document.createElement('button'),  
21   style: `background:${color};cursor:pointer;`,  
22   textContent: `${term.trim()}: ${matches.length}`  
23 ));
```

ADD ONCLICK TO BUTTONS

```
25 let idx = 0;
26 btn.onclick = () => (
27   matches[idx].style = `background:${color};border:2px solid
28   matches[idx].scrollIntoView({ behavior: 'smooth' }),
29   idx = (idx + 1) % matches.length
30 );
```

SECURITY CONCERNS

- Risks of running unknown bookmarklets
- Potential for data theft or malicious actions
- Use only trusted bookmarklets, review the code

CONCLUSION

Bookmarklets are simple and powerful tools that anyone can create with a bit of JavaScript magic. They enhance productivity, eliminate distractions, and improve your browsing experience!

CREATE YOUR OWN TODAY!

THANK YOU!

Simplify Work with JavaScript Bookmarklets

By Veronica Eulenberg