# Javascript Copy and Paste

[Fast, collaborative project management that's super easy to use. What a concept.](https://srv.carbonads.net/ads/click/x/GTND42JLCWYIE2J7FTSLYKQNCVBD55QIC6YIPZ3JCY7DV53ICE7IL53KC6BDV537CKYDP2QJFTSD65QLCTSIT53YHEYIKKJMC6SDL2JECTNCYBZ52K?segment=placement:paulundcouk;)[ads via Carbon](http://carbonads.net/?utm_source=paulundcouk&utm_medium=ad_via_link&utm_campaign=in_unit&utm_term=carbon)

In a previous tutorial we saw how you can create a copy and paste button using [jQuery and the ZeroClipboard flash extension](https://paulund.co.uk/jquery-copy-clipboard). The problem with this is that it requires flash. As most of the web is moving away from flash and some people don't even have flash installed on their browser we need another solution. The reason flash was used to copy and paste is mainly down to browser compatibility, now with most browsers now supporting the API to enable you to copy and paste directly using Javascript using the Web API document.execCommand. View the demo to see how the API works

## HTML

First we add the HTML to the page we need a textbox to enter the content we want to copy, a button that will have the click event to start the copy process and a textarea where we can test the copy is successful by pasting the contents into it.

<div>

<input type="text" id="text-to-copy" placeholder="Enter text" />

<button id="copy-text">Copy</button>

<span id="copied-text" style="display: none;">Copied!</span>

</div>

<div>

<textarea name="paste-area" id="paste-area" cols="30" rows="10" placeholder="Paste it here"></textarea>

</div>

## Copy Text

To copy the text we first need to add a click event to the button.

document.getElementById('copy-text').addEventListener('click', function(e){

// Click event

});

Inside the click event we need to select the text inside the textbox.

document.getElementById('text-to-copy').select();

With the text selected we can now use the execCommand('copy') to copy the text to the clipboard.

copied = document.execCommand('copy');

The return of this method will be a boolean on if the text was copied or not, you can then check this boolean value to display a message if it was able to be copied.

// Add click event

document.getElementById('copy-text').addEventListener('click', function(e){

e.preventDefault();

// Select the text

document.getElementById('text-to-copy').select();

var copied;

try

{

// Copy the text

copied = document.execCommand('copy');

}

catch (ex)

{

copied = false;

}

if(copied)

{

// Display the copied text message

document.getElementById('copied-text').style.display = 'block';

}

});

## Browser Compatibility

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature | Chrome | Firefox (Gecko) | Internet Explorer | Opera | Safari |
| `copy`/`cut` | 42 | [41](/en-US/Firefox/Releases/41 "Released on 2015-09-22.") (41) | 9 | 29 | 10 |

# Copy And Paste A Variable

The above example shows how you can copy and paste the contents of a HTML element. But what if you have a variable that you want to copy the contents to the clipboard.

The below code shows how you can take the contents of a variable, create a textarea element, set the value of the textarea and then copies the contents of the textarea onto your clipboard.

copyText (textToCopy) {

this.copied = false

// Create textarea element

const textarea = document.createElement('textarea')

// Set the value of the text

textarea.value = textToCopy

// Make sure we cant change the text of the textarea

textarea.setAttribute('readonly', '');

// Hide the textarea off the screnn

textarea.style.position = 'absolute';

textarea.style.left = '-9999px';

// Add the textarea to the page

document.body.appendChild(textarea);

// Copy the textarea

textarea.select()

try {

var successful = document.execCommand('copy');

this.copied = true

} catch(err) {

this.copied = false

}

textarea.remove()

}

https://paulund.co.uk/javascript-copy-and-paste

function copy() {

if(window.clipboardData) {

window.clipboardData.clearData();

window.clipboardData.setData("Text", document.getElementById('txToCopy').value);

}

}

function paste() {

if(window.clipboardData) {

document.getElementById('txToPaste').value = window.clipboardData.getData("Text");

}

}

var myClipboardVariable;

document.onkeyup = function(e){

if ((e.key == 'c') && e.ctrlKey){

// save data (you want to copy) into variable

myClipboardVariable = ....//some data

}

if ((e.key == 'v') && e.ctrlKey){

// paste saved data

.... paste your data from variable myClipboardVariable

}

}

0

have a look at [this MDN article](https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Interact_with_the_clipboard).

If you just want to copy user selected text you can do:

document.execCommand("copy");

if you need to select it previously:

document.getElementById('txToPaste').select();

* In my case this code didn't work - it turns out select() don't work for disabled inputs.
* you don't need any special permissions if you run it from an onClick event listener, if you want another event to trigger the copy you are a bit in trubbles.
* it works for me on Firefox and chrome, MDN says it won't work for safari but I haven't tested it.