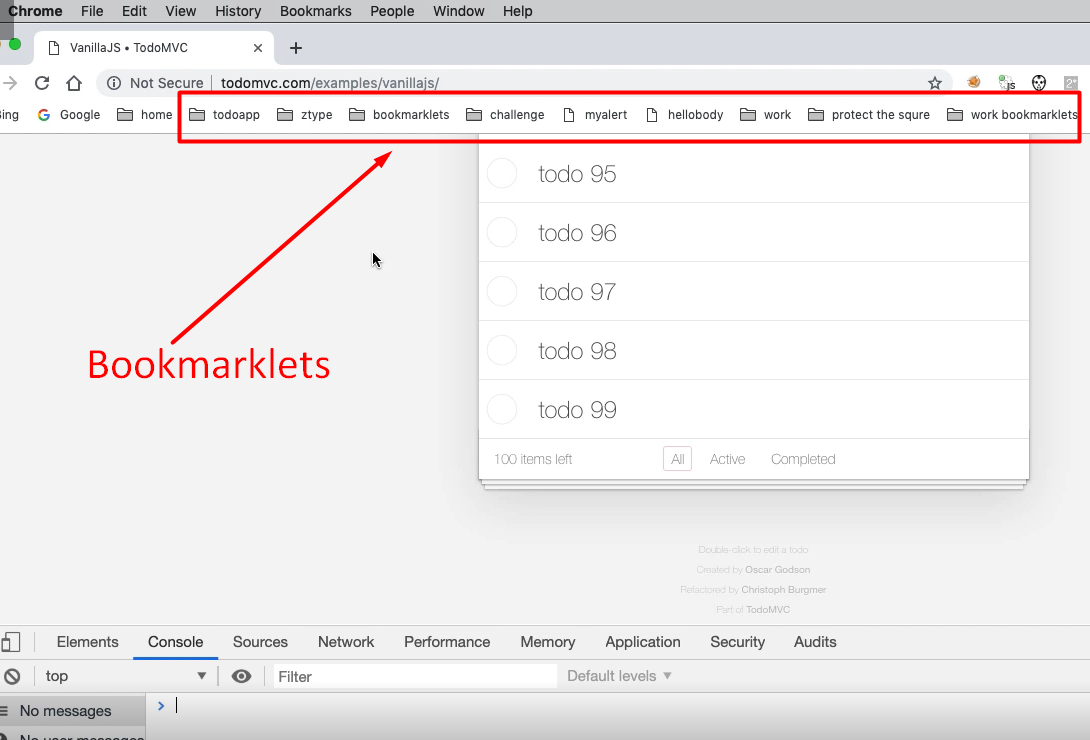
Now a lot of the helper tools that I use when I'm working with JavaScript from a console, I convert into bookmarklets.



I have a range of bookmarklets up here depending on what applications I'm working with and then I can start using functionality in there.

If I was testing todos, I'd have a set of bookmarklets for todoMVC which would allow me to create data, or clear down the data, or configure some users in the application. Whatever it is, whatever functionality I've written, if I can convert it into bookmarklets that's quite useful to me. Because in Chrome if I'm logged in and I have bookmarklets they're synchronized across all the different browsers, so I have the same bookmarklets on the different platforms that I'm testing.

**Let me just show you what a bookmarklet is.**

Imagine that I have functionality to, let’s just make it simple and obvious, so I'm going to have an alert that says, “hello”.

alert(“hello”) — there we go.

What I'm going to do is create a bookmarklet for that.

Now a bookmarklet is just a set of JavaScript, which is a function that runs immediately, which also has a javascript:on the front.

**Let me show you the steps for this.**

I would write a function that runs immediately, which is like this:



So, this doesn't do anything.

Now when we're working with bookmarklets all we really have to do is remember the pattern.

* We write some working code.
* We create an anonymous function that runs immediately.
* We put our code in there (this will still run as code here.)
* Then, in order to make a bookmarklet, we need to prefix it with javascript:

That then allows me to run it in the URL bar or as a bookmarklet.

javascript: (function() {alert(“hello”);})()

Let me take this code, run this up here.



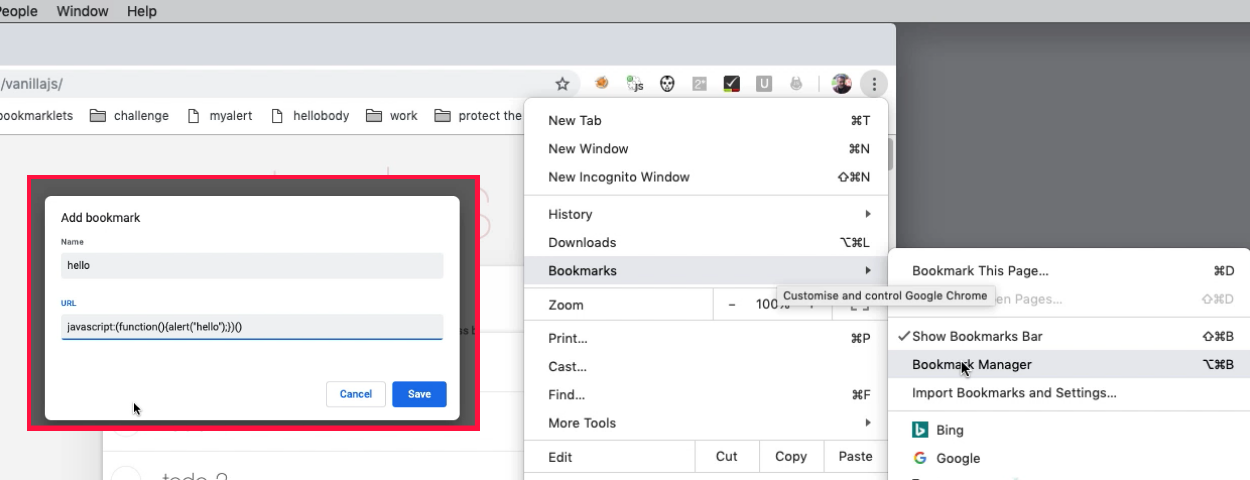
There we go, I've run my code from the browser.

What I'm going to do now is going to take this code and put it in as a bookmark.

If I go into *Bookmarks > Bookmark Manager*, I'm going to create a new folder for this [*Add Folder* is in the top menu pull down on the right]. I'm going to call this “vanillaJS todo” and *Save*.

Then let’s go into the folder: *double click > right click [Add new bookmark]*.

For name, I'll just call this “hello”, paste this code in the URL line and *Save*.



Now in my bookmarks I've got my “vanillaJS todo” folder, there's my 'hello” bookmarklet.

I say hello [click the bookmarklet, just like you would any other bookmark link] and it says 'hello'.

**What we need to do now is make something that is just a little bit more flexible than that.**

We already have some code for creating a hundred todos.

Let me create a new snippet and rename this “100 todos bookmarklet”. I'm just going to copy and paste this code here.

So, this is my pattern:



You can see that I've got my javascript: which essentially is what makes it a bookmarklet. I've got my “create an anonymous function” that runs immediately.

Now I need to add in the code.

We already have the code for creating a hundred todos, I'm just going to use the setTimeout one because its nice and simple.

I'll copy that code put it into my bookmarklet.

javascript: (function() {

for(x=1;x<=100;x++){

setTimeout(

function (name){

document.querySelector("input.new-todo").value=name;

document.querySelector("input.new-todo").dispatchEvent(new Event('change', { 'bubbles': true }))

}

, x\*100, "todo " + x)

}

})()

Now if I run this, we should see the 100 todos being created.

Let me just clear the console first — localStorage.clear() —refresh

Back to sources.

Run my snippet just to make sure it works. That's working.

Whilst that is doing its thing, I'm going to set up the bookmarklet.

I'm going to copy this, go into *Bookmarks > bookmark manager >double-click, Add new bookmark*.

This will say, “create 100 todos”; I'll paste in my “URL” and *Save* that.

Back to the application. That's finished running, so let’s just clear the local storage again, refresh.

Now if I go to bookmarks, “create 100 todos”, there we go.

Now I have essentially a little utility that I can run against this. So, if you're creating test data, any functionality you've created to help you, you can add into a bookmarklet and that will then share it across the different browser that you use.

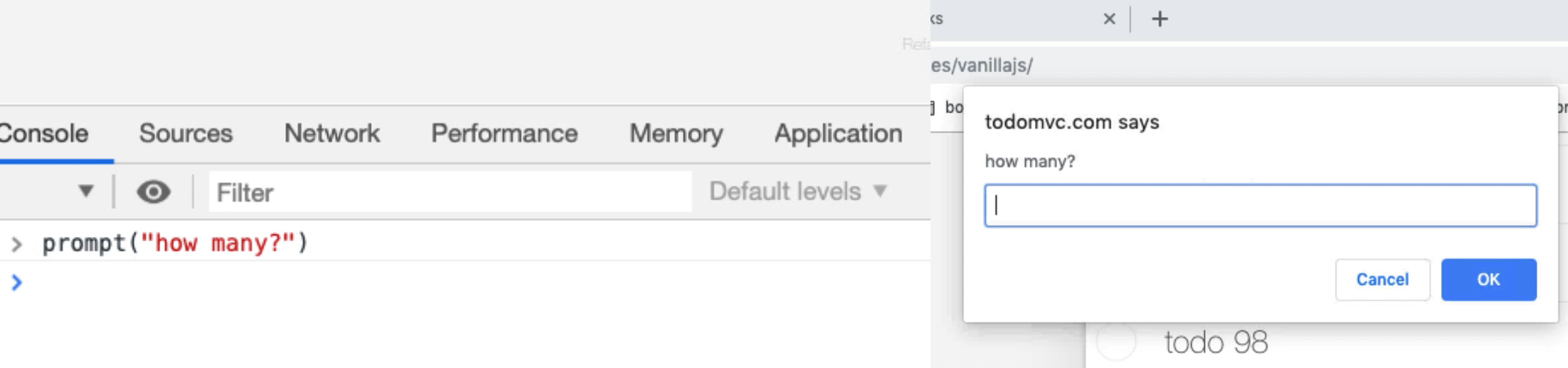
**But if it’s a tool, let’s make it even more useful to us.**

One of the things we can do in JavaScript, is we can prompt the user for input.

So, if I say 'prompt' and then say 'how many todos do you want'.

prompt("how many?");

How many?



Well I want a 100.

* If I say 'okay' with a hundred then I'll get a hundred back.
* If say, how many and just cancel I'll get a “null”.
* If I say just “OK”, I get an empty String.

I can basically say if, and then if the prompt has a value then we can actually do this in a loop.

Let me just show you what I mean.

javascript: (function() {

for(x=1;x<=100;x++){

max = prompt("how many?");

if(max){

for(x=1;x<=max;x++){

setTimeout(

function (name){

document.querySelector("input.new-todo").value=name;

document.querySelector("input.new-todo").dispatchEvent(new Event('change', { 'bubbles': true }))

}

, x\*100, "todo " + x)

}

})()

So here, instead of a 100, what I could say the maximum I want is going to be defined by the user by a prompt, “how many”.

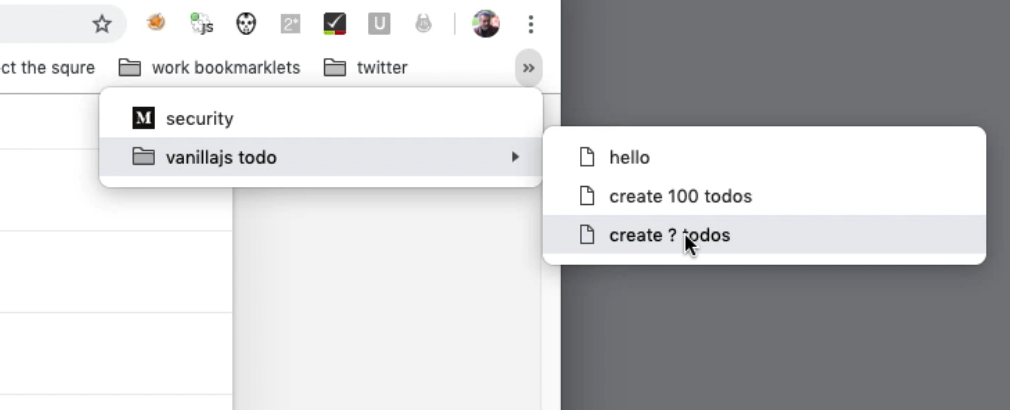
If max actually has a value, then rather than do it up to a 100, we'll do it up to or less than max.

If I run this — how many? — let’s say 3; and we should get 3 on the end there: 1, 2, 3.

Now we've got an actually very usable tool. Let me copy paste this.

Back to my *Bookmarks > Bookmark > Manager > Folder*. There we go. Then *Add new bookmark*, then [give it a name] “create ? todos”, [add code as] URL, *Save* that.

Now if I go into *“vanillasjs todo [folder] > “create ? todos”*:



[I can run it by clicking the link just like a bookmark].

Let’s have 5 todos, please. 1, 2, 3, 4, 5.

So, we can make something very, very useful that just sits in the bookmarklets and help us out with our tasking.

And this is just the extra buildup of all the techniques we've already had.

* We know how to write simple self-contained code.
* We know how to use for loops.
* We know how to do setTimeout.
* We know how to create our own functions.

What we're doing here is creating a function that runs immediately, prefixing it with javascript:, so we can copy it into a bookmarklet. And really all you have to do is remember that pattern of the “start” part — some code that does something — then “end” wrapper and then you'll be able to create bookmarklets very easily.

**Resources**

* [Quick Intro To BookMarkLets](https://www.eviltester.com/2017/04/a-quick-intro-to-bookmarklets.html)
* [Simple Bookmarklet Creation Utility](https://github.com/eviltester/bookmarklet)

**Quiz: +100 credits**

**1. What is a bookmarklet?**

JavaScript code that runs from a bookmark

a Chrome extension

a node.js application for managing book collections

**2. Which of the following is the pattern for a bookmarklet?**

(function(){alert("hello")})()

javascript:(function(){alert("hello")})()

function(){alert("hello")}()

javascript:({alert("hello")})()

**3. Which of the following is true for bookmarklets?**

we can run code that amends the DOM

bookmarklets can be released to the chrome store as extensions

bookmarklet code can only write to the console or create alerts

**4. The JavaScript function for capturing input from the user is?**

prompt

alert

input