

JavaScript & The Browser

- Scripts
- Chrome Developer Tools
- Firefox Developer Tools

- Allows a program directly in html
- Bad idea to place JavaScript in html
- Instead use source tags

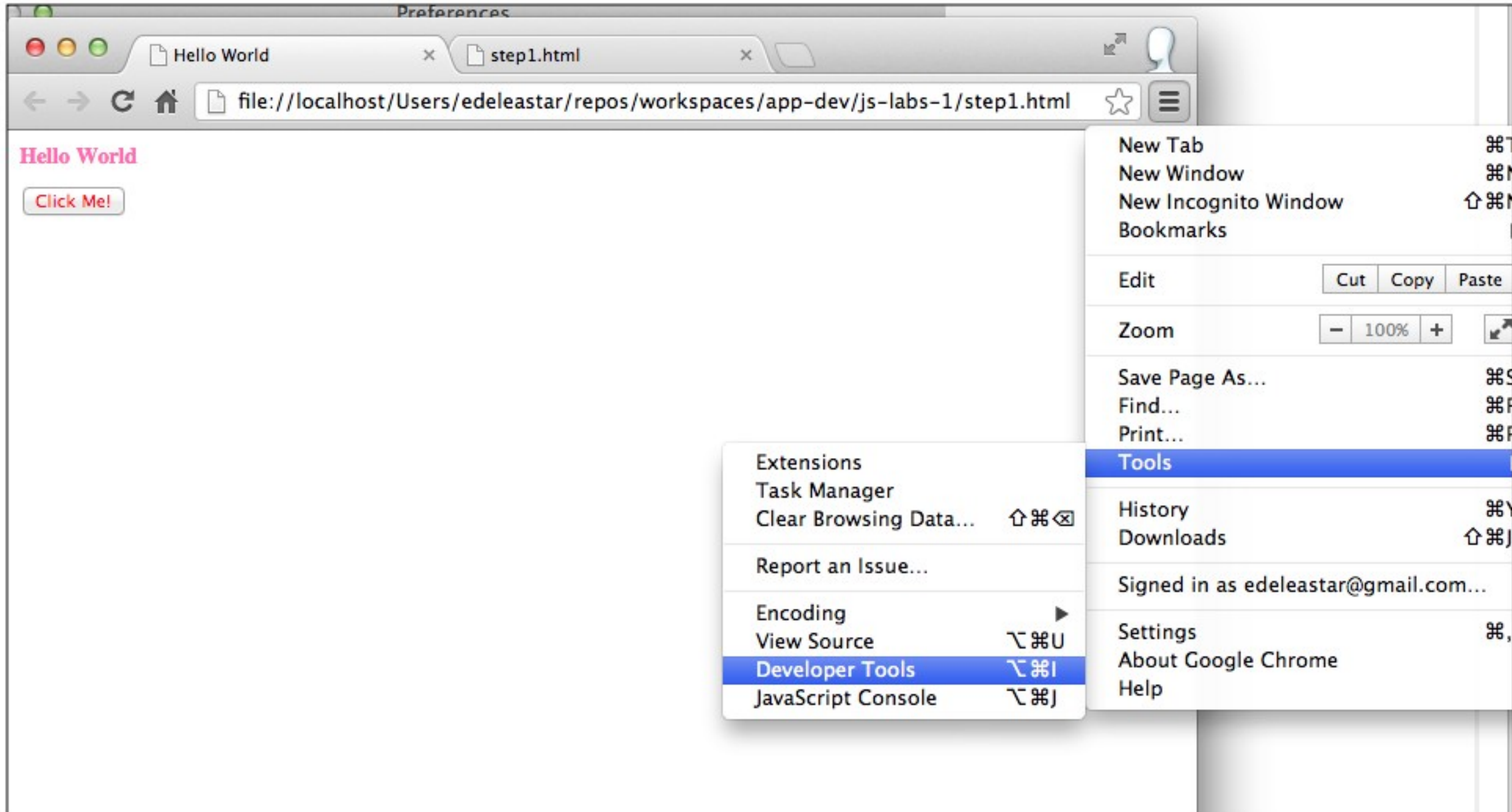
```
<!DOCTYPE html>
<html>
  <head>
</head>
  <body>
    ...
    ...
    <script src="js/jquery-2.0.0.js"></script>
    <script src="js/reportMap.js"></script>
  </body>
</html>
```

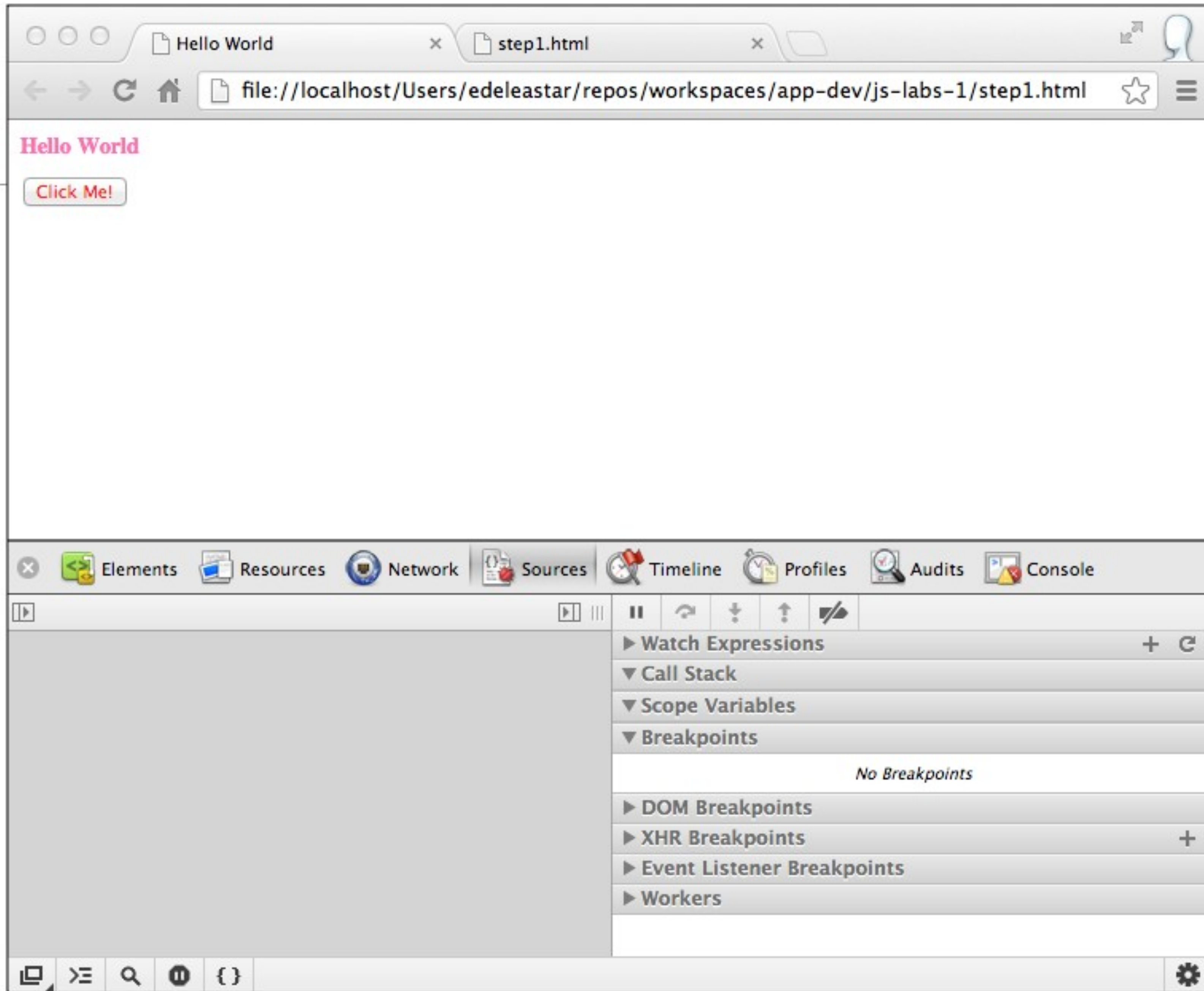
Script Tags

- Script files may have big impact on page load
- Place tags close as possible to bottom of body
- Reduce number of script files as much as possible
- Minify script files in release versions

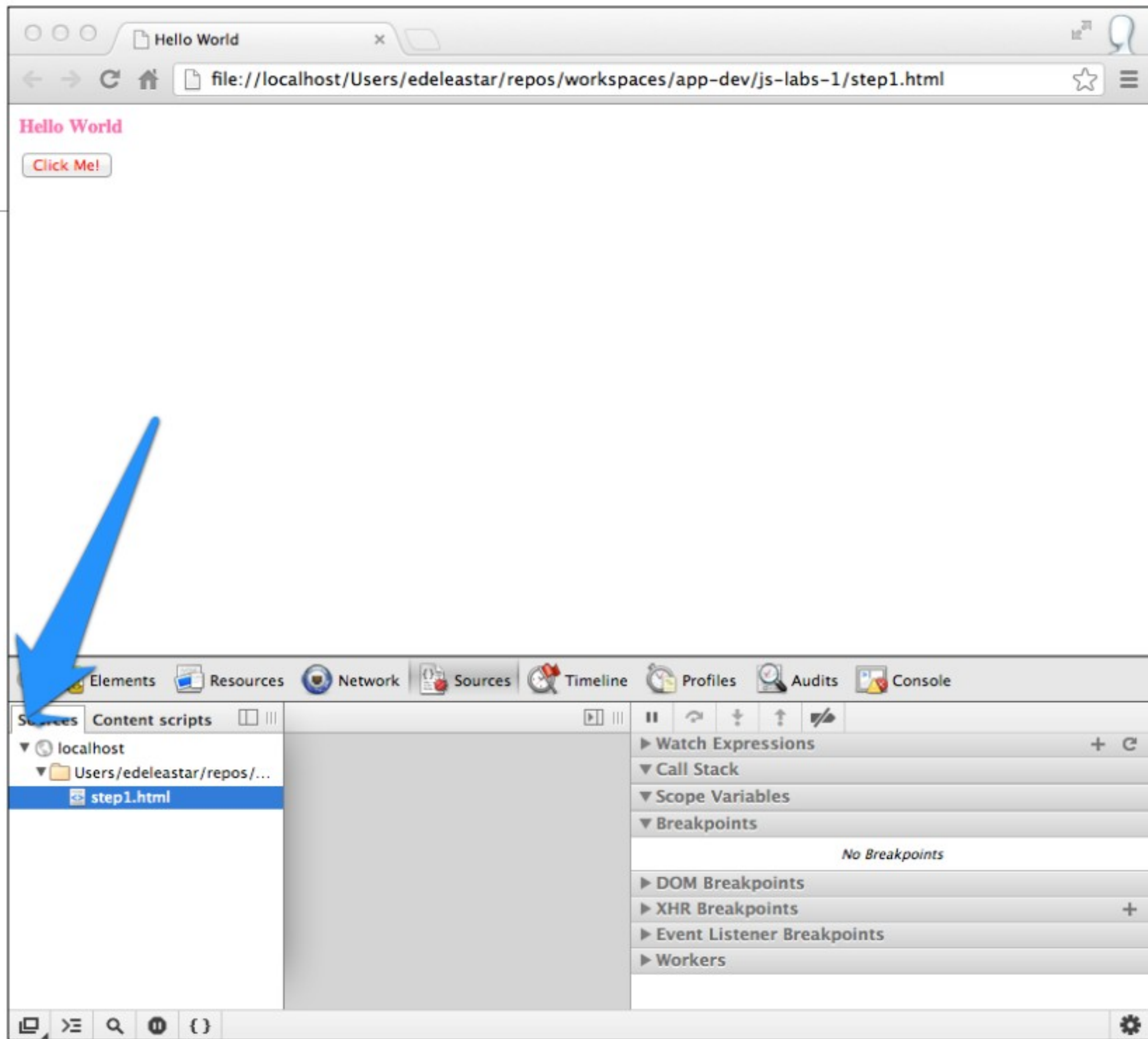
View->Developer Tools

Chrome



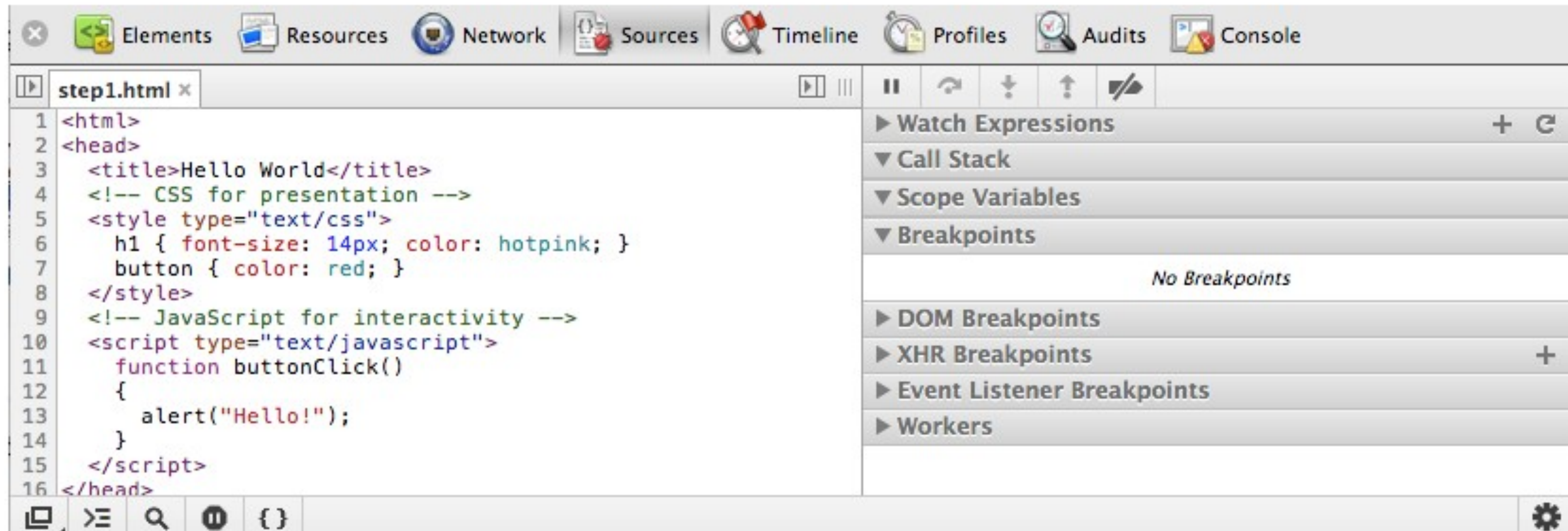


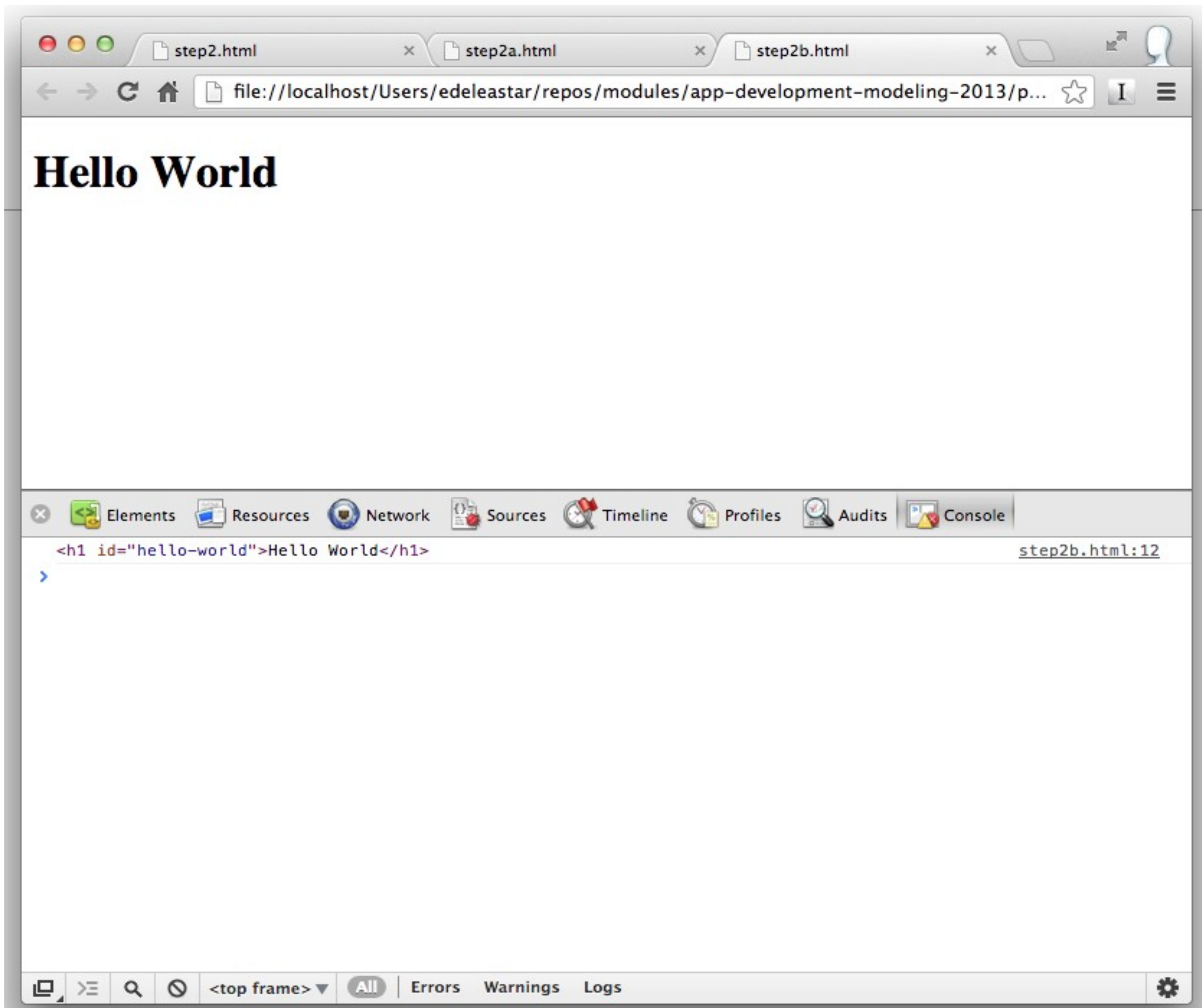
Chrome



Chrome

Source view in Developer Tools





Chrome

Page contains:

- paragraph <p>
- input elements <input>
 - text field
 - buttons
- list

This page contains a list, which will be modified by pressing the following button:

1. An Item

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Changing the DOM</title>
    <script type="text/javascript" src="script.js"></script>
  </head>
  <body>
    <p>
      This page contains a list, which will be modified by pressing the following button:
    </p>
    <input type="text" id="itemtext" />
    <input type="button" value="Add One" onclick="addElementById('itemtext')" />
    <input type="button" value="Clear All" onclick="clearList()" />
    <ol id="list">
      <li> An Item </li>
    </ol>
  </body>
</html>
```



```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Changing the DOM</title>
    <script type="text/javascript" src="script.js"></script>
  </head>
```

```
<input type="button" value="Add One" onclick="addElementById('itemtext')" />
<input type="button" value="Clear All" onclick="clearList()" />
```

- The script element identifies a file containing javascript functions
- The button elements identify the functions + parameters, to be called when the buttons are clicked
- The functions directly manipulate the DOM, changing the content of the current page

```
function addElementById(itemId)
{
  var list = document.getElementById('list');
  var itemText = document.getElementById(itemId);
  var newItem = document.createElement('li');
  newItem.innerHTML = itemText.value;
  list.appendChild(newItem);
}

function clearList()
{
  var list = document.getElementById('list');
  list.innerHTML = "";
}
```

script.js

Example

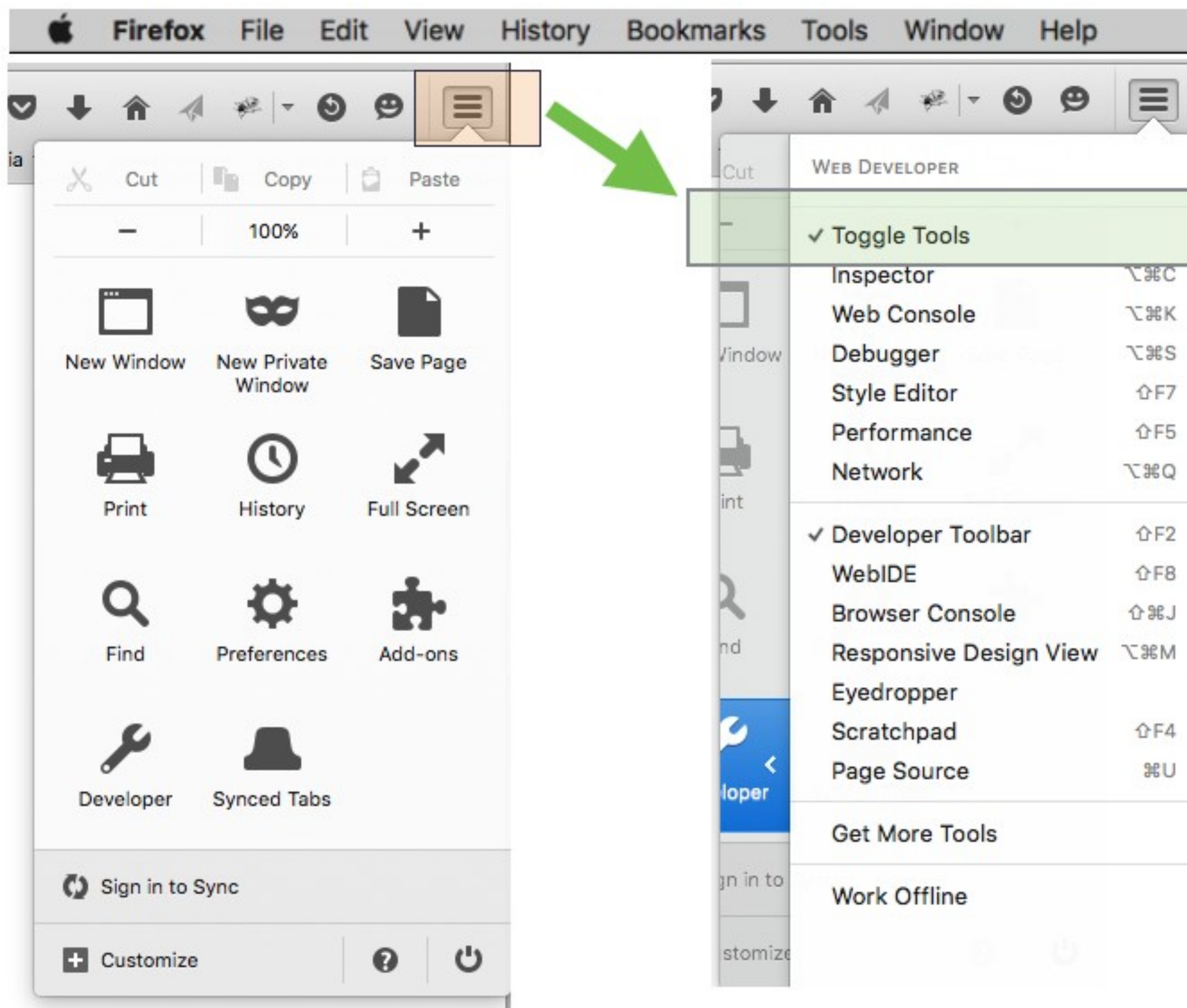
This page contains a list, which will be modified by pressing the following button:

1. An Item

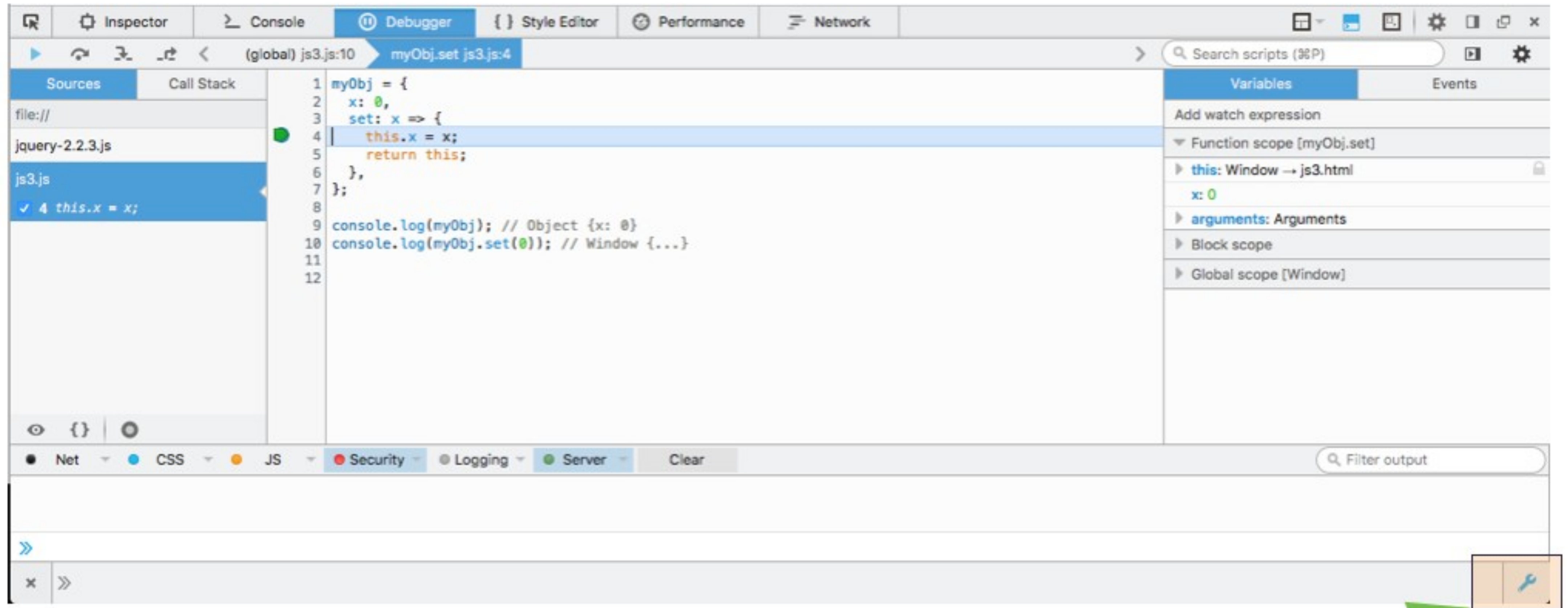
This page contains a list, which will be modified by pressing the following button:

1. An Item
2. test one
3. test two

- For a static page, clicking on a link/button takes the browser to a new page (new url)
- With a dynamic page (javascript enabled), clicking on a button may change the *current* pages structure, content or style



Firefox



Toggle Tools

Firefox

External javascript file js3.js

Debugger

myObj.set js3.js:5

Sources

Call Stack

file://

jquery-2.2.3.js

js3.js

4 this.x = x;

```
1 myObj = {
2   x: 0,
3   set: x => {
4     this.x = x;
5     return this;
6   },
7 };
8
9 console.log(myObj); // Object {x: 0}
10 console.log(myObj.set(0)); // Window {...}
11
12
```

Variables

Add watch expression

Watch expressions

- x → 0
- this.x → 0
- this → Window → js3.html

Function scope [myObj.set]

- this: Window → js3.html
- x: 0
- arguments: Arguments
- Block scope
- Global scope [Window]

Execution halted at breakpoint on line 5

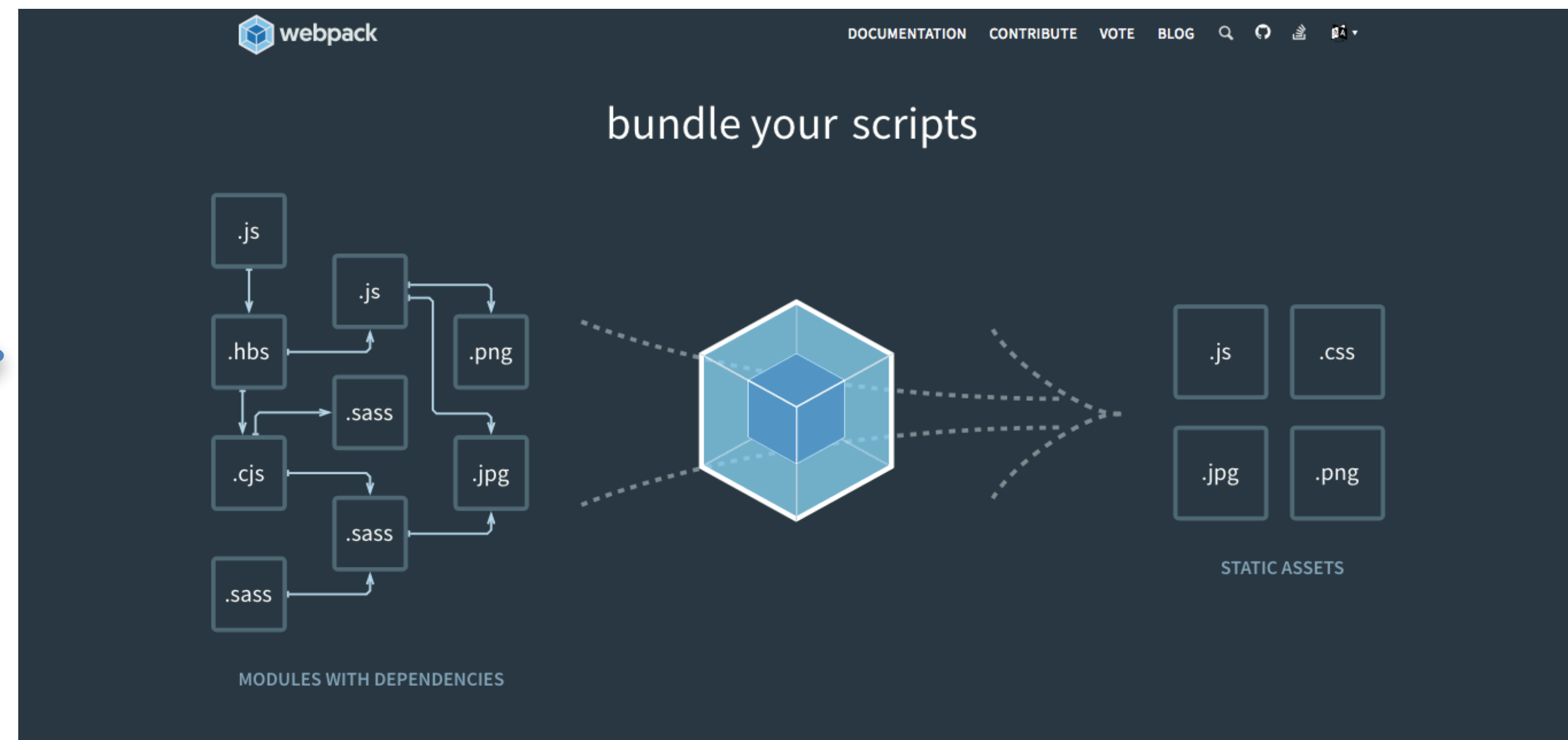
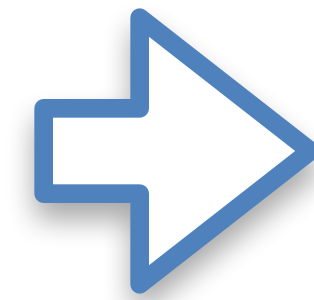
Values in scope viewable in Watch window

Summary

- Scripts
 - How to load - in html or external files
 - Strong recommendation to use external files only
 - Minification - use in production
- Developer tools
 - Chrome
 - Firefox
 - Regular use encouraged
 - May be sometimes advantageous to simultaneously use both

<https://webpack.js.org/>

- Script files may have big impact on page load
- Place tags close as possible to bottom of body
- Reduce number of script files as much as possible
- Minify script files in release versions



Bundle with webpack

Without `config` or provide custom `webpack.config.js`

```
const path = require('path');

module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js'
  }
};
```

page.html

```
<!doctype html>
<html>
  <head>
    ...
  </head>
  <body>
    ...
    <script src="dist/bundle.js"></script>
  </body>
</html>
```

Then run `webpack` on the command-line to create `bundle.js` .