

JavaScript Introduction

Topics discussed

- Scripts
- Chrome Developer Tools
- Firefox Developer Tools

Script Tags

Inserts program in html document

```
<script> </script>
```

- Allows program 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>
```

Script Tags

Where to locate

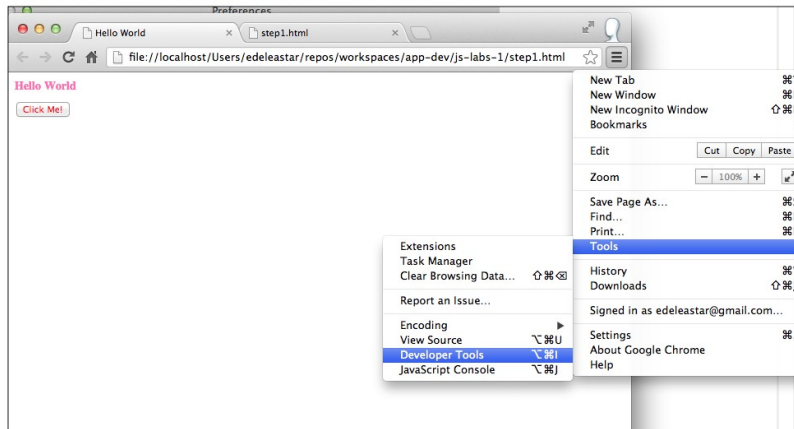
`<script></script>`

- Script files may have big impact on page load
- Place tags close as possible to bottom of body
- Place css `<link>` high as possible in head
- Reduce number of script files as much as possible
- Minify script files in release versions
 - Has big impact on load time

Chrome Dev Tools

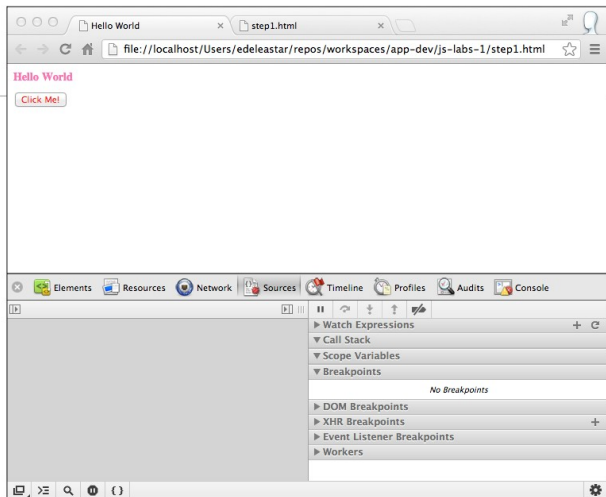
Web authoring & debugging tools

View->Developer Tools



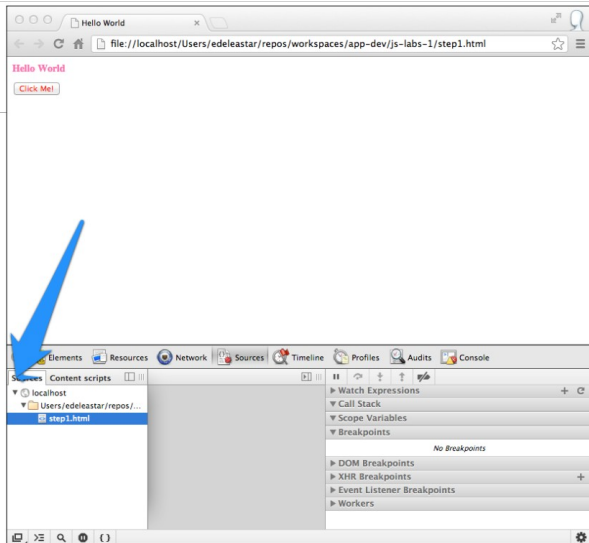
Chrome Dev Tools

Web authoring & debugging tools



Chrome Dev Tools

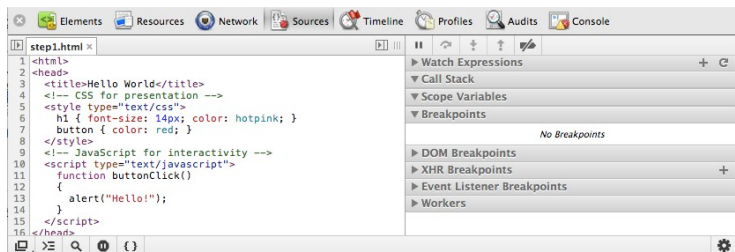
Web authoring & debugging tools



Chrome Dev Tools

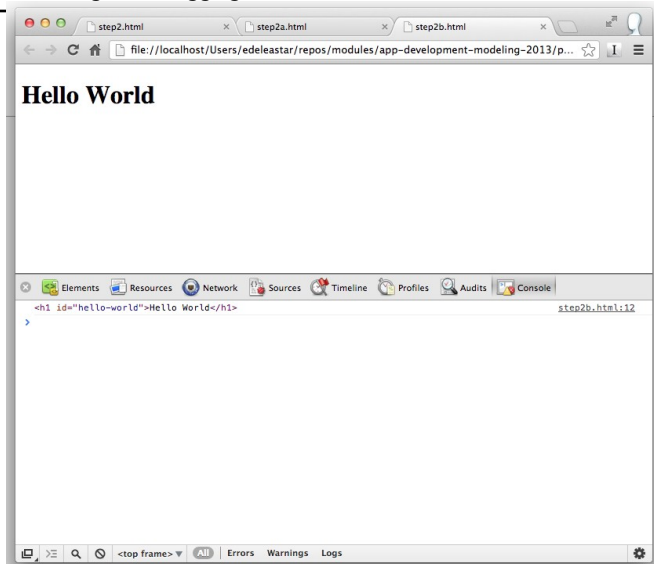
Web authoring & debugging tools

Source view in Developer Tools



Chrome Dev Tools

Web authoring & debugging tools



Example

Button press causes invocation JavaScript function

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>
```

Example

JavaScript functions

```
<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

Static v Dynamic

JavaScript enabled page

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:

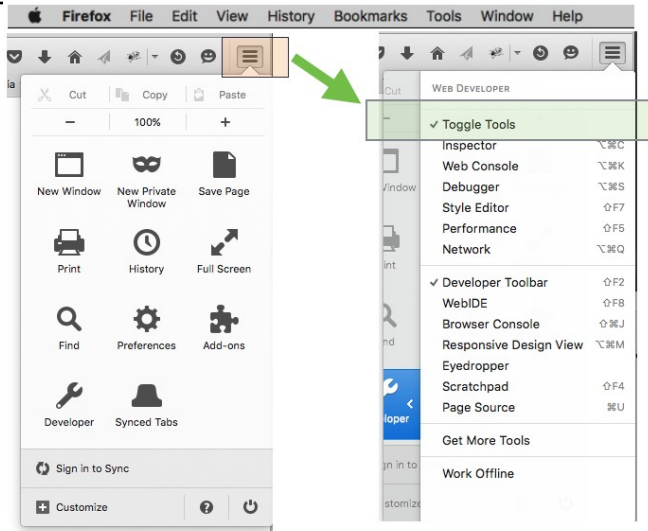
test two

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

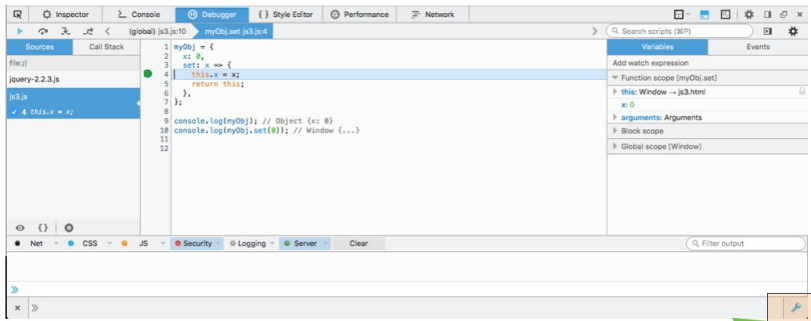
Firefox Dev Tools

Web authoring & debugging tools



Firefox Dev Tools

Web authoring & debugging tools



Toggle Tools

Firefox Dev Tools

Web authoring & debugging tools

External javascript file js3.js

The screenshot shows the Firefox DevTools Debugger interface. The Sources panel on the left shows the file structure with 'js3.js' selected. The main editor displays the code for 'myObj.set' in 'js3.js:5'. A breakpoint is set on line 5, and the execution has halted. The Watch window on the right shows the current scope, including 'x: 0' and 'this: Window -> js3.html'.

```
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
```

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
 - Number scripts
 - Minification - use in production

- Developer tools
 - Chrome
 - Firefox
 - Regular use encouraged
 - May be sometimes advantageous to simultaneously use both