CSC 470 – Section 3

Topics in Computer Science: Advanced Browser Technologies

Mark F. Russo, Ph.D.

Spring 2016

Lecture 1

The World Wide Web

- Invented by Sir Tim Berners-Lee, a British computer scientist at CERN in 1989 (Conseil Européen pour la Recherche Nucléaire)
- High-energy-physics community
- To meet the demand for automatic information-sharing between scientists in universities and institutes around the world
- First web site hosted on Berners-Lee's NeXT computer
- Software placed in the public domain in 1993

- Original design was to share static information

http://home.cern/topics/birth-web

First Web Site

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.

Help
on the browser you are using

Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical
Details of protocols, formats, program internals etc

Bibliography
Paper documentation on W3 and references.

People
A list of some people involved in the project.

History
A summary of the history of the project.

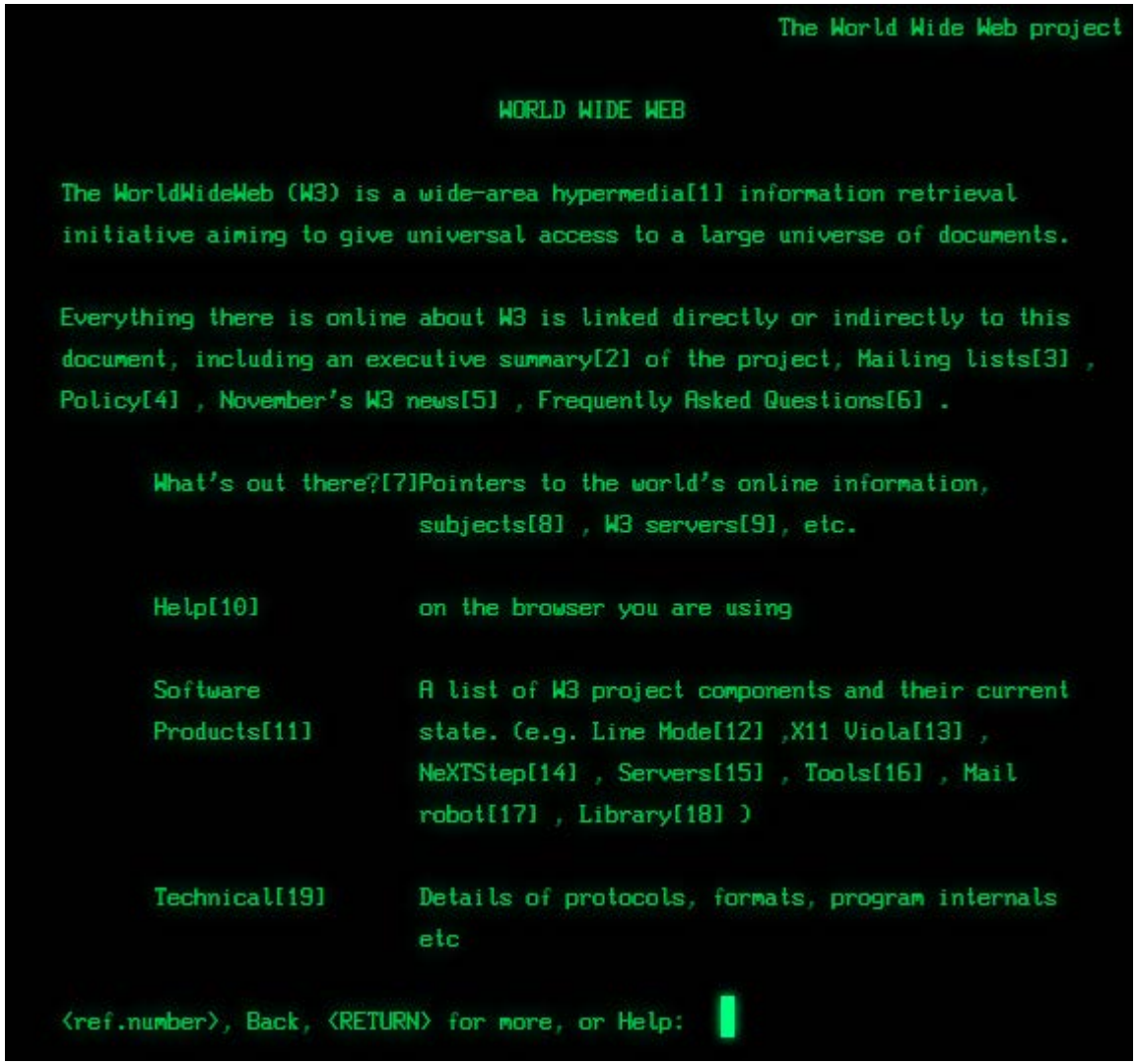
How can I help ?
If you would like to support the web..

Getting code
Getting the code by anonymous FTP , etc.

- Static text
- Basic formatting
- Hyperlinks

http://info.cern.ch/hypertext/WWW/TheProject.html

First Browser

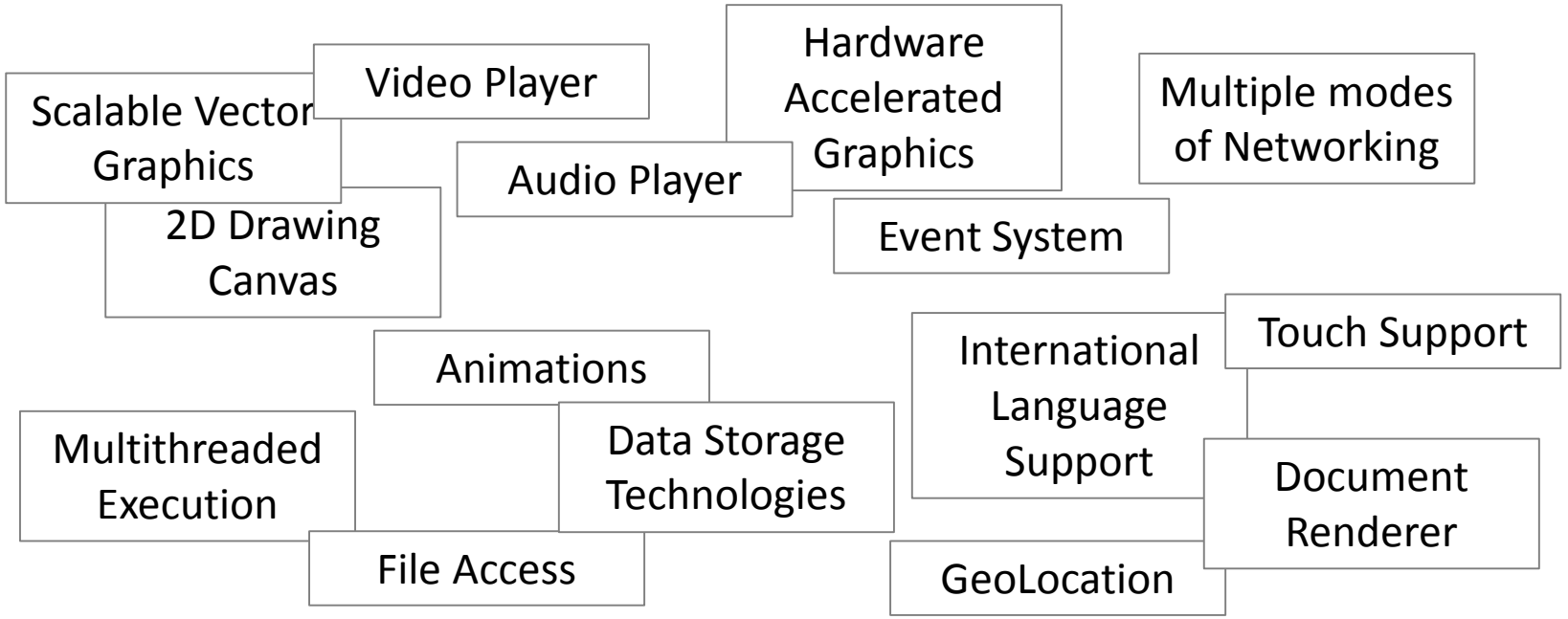


- Line mode terminal
- Commands entered at prompt
- Graphical browsers came later

http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html

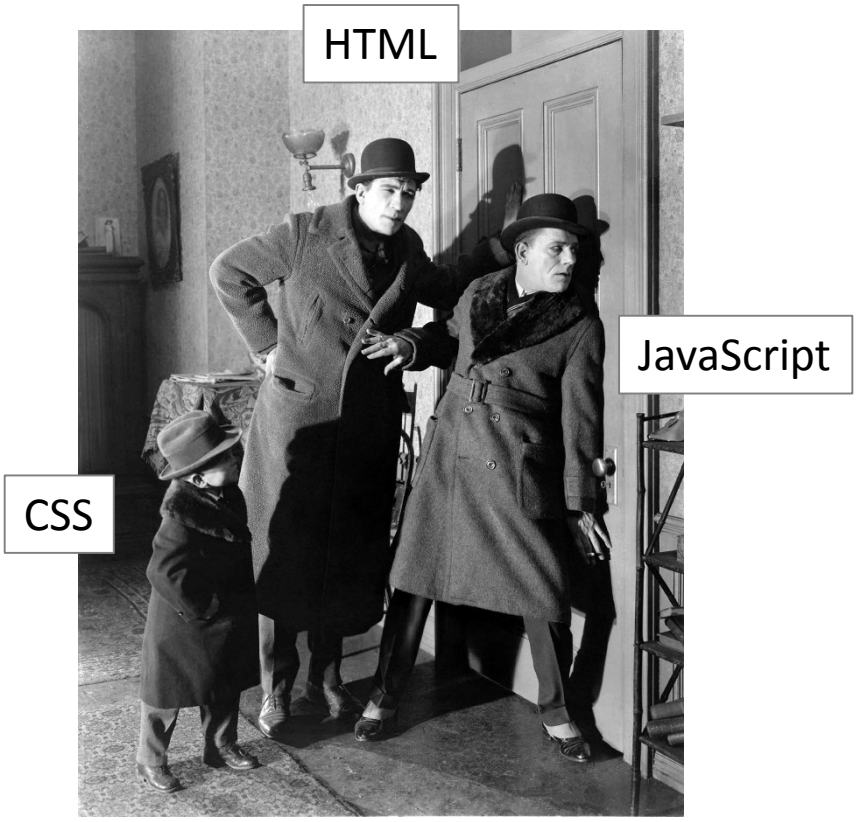
The Modern Web Browser

- Transformed from a simply document render into a full-blown application development platform
- See standard browser API
  - https://developer.mozilla.org/en-US/docs/Web/API



The Unholy Three

- A 1925 American silent film involving a crime spree?
- The three essential technologies implemented by modern web browsers.



The Unholy Three: HTML, CSS, JavaScript

HyperText Markup Language (HTML)

- The standard markup language used to create web pages
- Defines the structure and makeup of a web page
- Read and processed by a web browser
- Organized hierarchically into the Document Object Model (DOM)
  - starting from a single root node `<html>` and branching out

Cascading Style Sheets (CSS)

- A language used for describing the presentation
- The first release of CSS (CSS1) was in 1996 by the World Wide Web Consortium (W3C)
- Used to set the visual style
- Enables separation of document content from presentation, including layout, colors, and fonts.

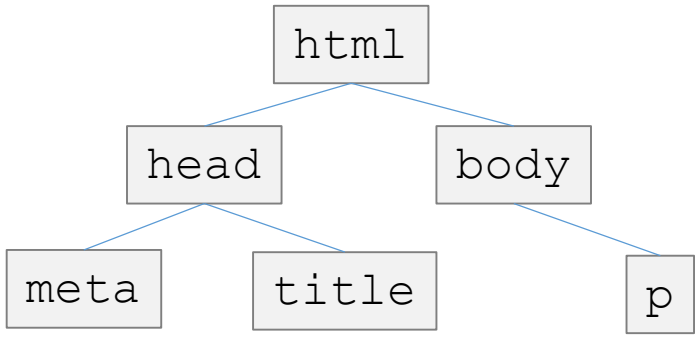
JavaScript

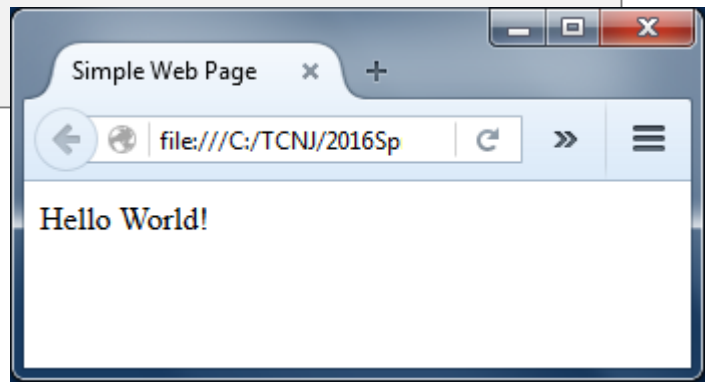
- Dynamic programming language for general-purpose application
- Supports object-oriented, imperative, and functional programming styles
- Embedded in nearly all modern web browsers
- Means by which all web browser technologies can be accessed

HTML

- A “language” based on nested tags `<tag> … </tag>`

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Simple Web Page</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```





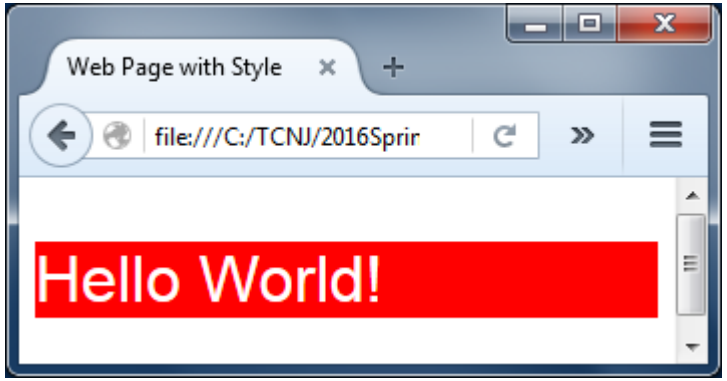
01/simple.html

http://www.w3schools.com/html/html_examples.asp

CSS

- Separates content style from structure and content
- Provides a central place to the look of several web pages
- May be set within `<style>` tags.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Web Page with Style</title>
    <style>
      p {
        font-family : arial;
        font-size : 20pt;
        color : white;
        background-color: red;
      }
    </style>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```



http://www.w3schools.com/css/css_examples.asp

01/simple2.html

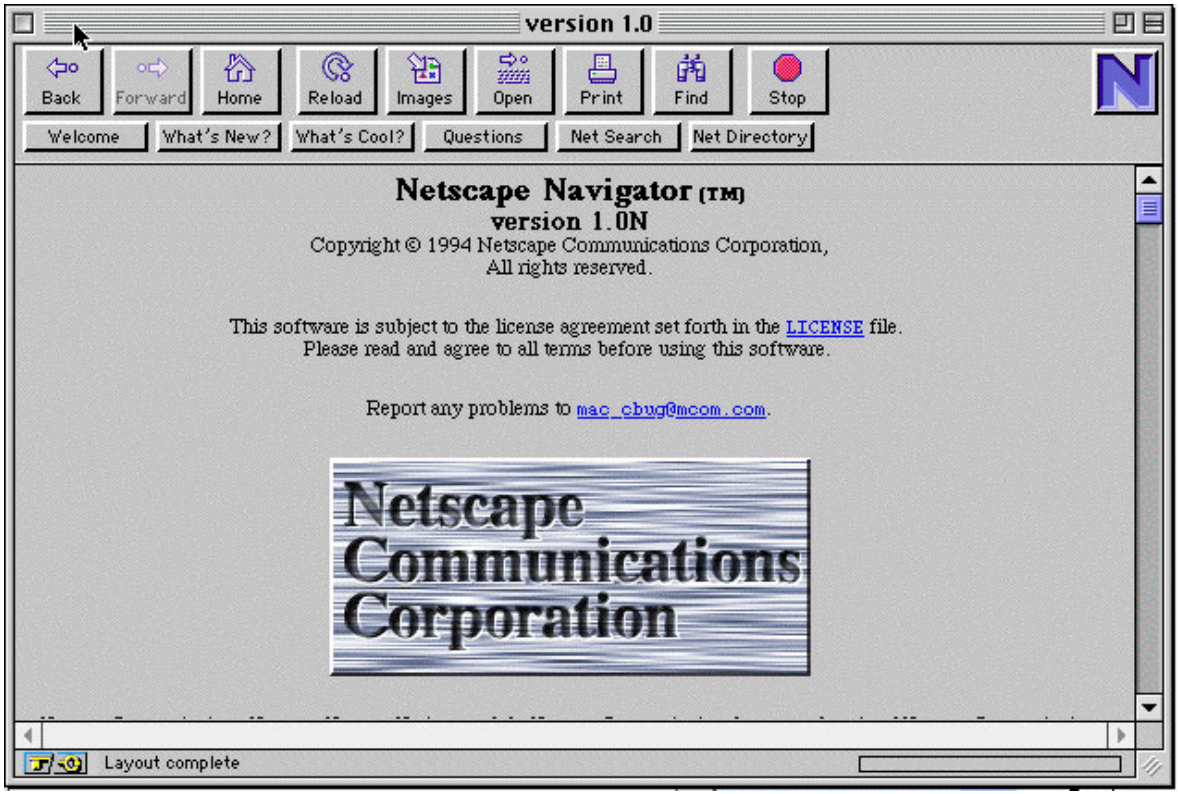
JavaScript History

- Created in 10 days in May 1995 by Brendan Eich, then working at Netscape
- Introduced as a way to make pages more dynamic
- Originally named Mocha, then LiveScript
- Ultimately named JavaScript when Sun acquired trademark n 1995
  - Marketing strategy to promote their Java language
  - Has virtually nothing to do with Java
- In 1996 - 1997 JavaScript was taken to ECMA to standardize
  - European Computer Manufacturers Association
- ECMA-262 Ed.1: ECMAScript is the name of the official standard
  - JavaScript is the most well known of the implementations
- ECMAScript 2 in 1998 and ECMAScript 3 in 1999
- ECMAScript 4 failed due to Microsoft's unwillingness to implement in Internet Explorer
- In 2009 ECMAScript 5 was adopted – the version in wide use today
- In 2015 ECMAScript 6 was completed – browser authors are actively implementing

https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript

Netscape Navigator

- First browser to include JavaScript

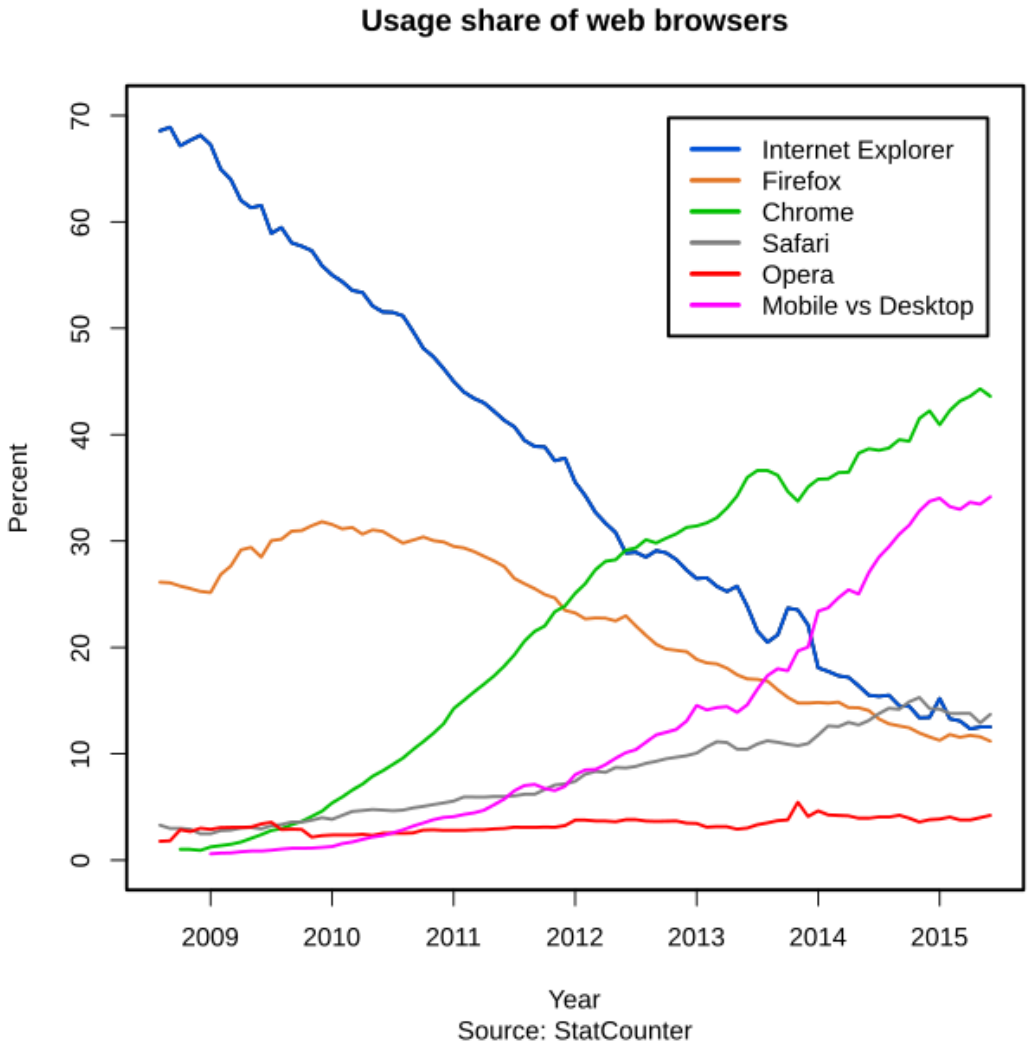


Major JavaScript Implementations

- SpiderMonkey
  - Mozilla's JavaScript engine written in C/C++
  - Used in various Mozilla products, including Firefox
  - Free and open source
- V8
  - Google's open source, high performance JavaScript engine
  - Written in C++ and is used in Google Chrome
  - Compiles JavaScript to native machine code, on-the-flly
- Chakra
  - Developed by Microsoft for its Internet Explorer 9 (IE9) web browser
  - Open Sourcing the ChakraCore engine in January of 2016 on GitHub
- Nitro (SquirrelFish Extreme)
  - Developed by Apple for Safari and other applications
- Nashorn
  - A standalone JavaScript Engine built in to Oracle Java JDK 8
  - Compiles JavaScript to Java bytecode
  - Has access to all Java libraries

Continuous JIT compiler improvements have improved JavaScript performance to the point that it can now run at "near native" speeds

Browser Market Share



Firefox Developer Tools

Web Console

- Logs information associated with a web page: network requests, JavaScript, CSS, security errors and warnings as well as error, warning and informational messages explicitly logged by JavaScript code running in the page context
- Enables you to interact with a web page by executing JavaScript expressions in the context of the page

JavaScript Debugger

- Enables you to step through JavaScript code and examine or modify its state to help track down bugs.

Page Inspector

- To examine and modify the HTML and CSS of a page

Network Monitor

- Shows you all the network requests Firefox makes, how long each request takes, and details of each request.

Style Editor

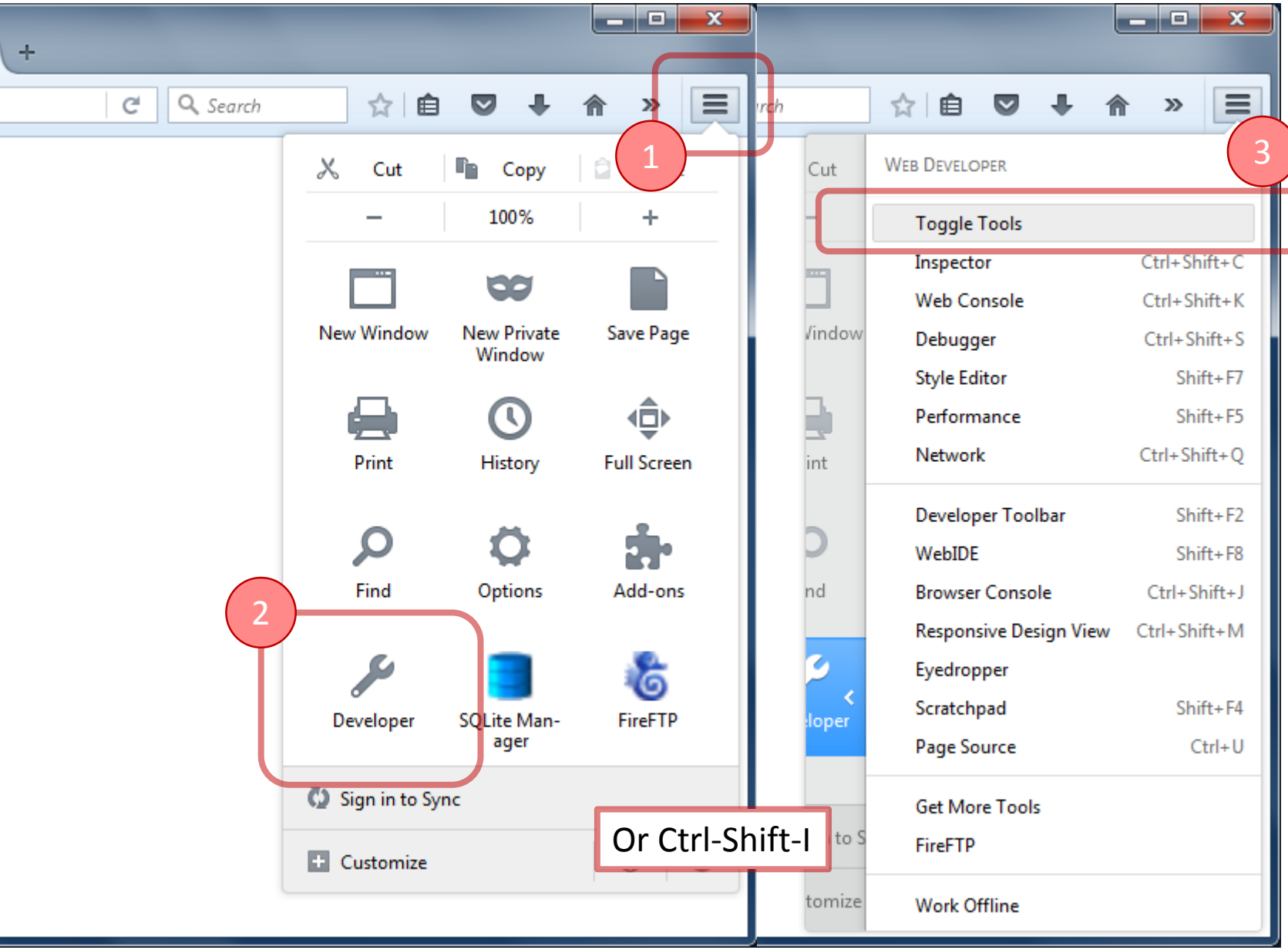
- View and edit all the stylesheets associated with a page
- Create new stylesheets from scratch and apply them to the page
- Import existing stylesheets and apply them to the page

ScratchPad

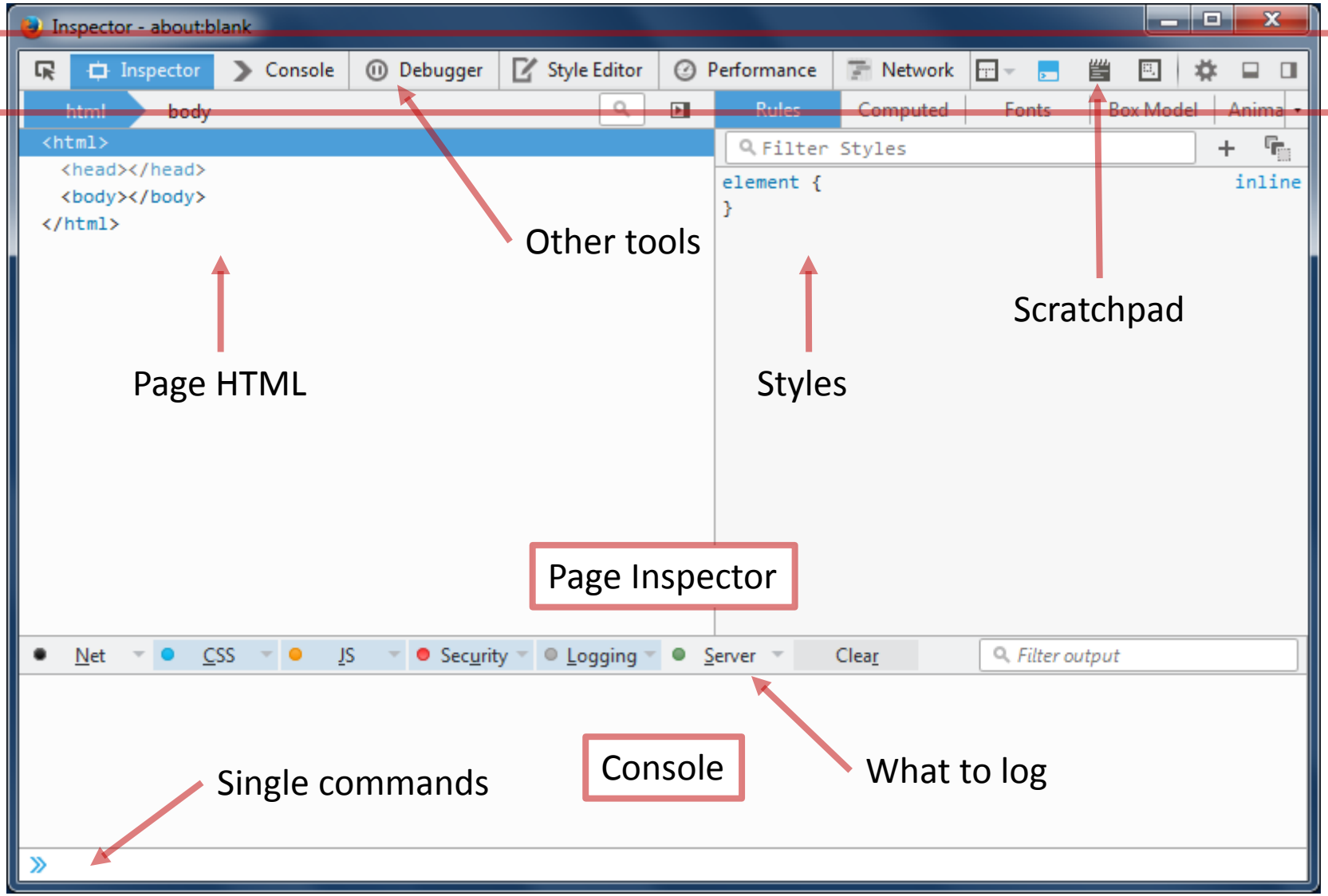
- Provides an environment for experimenting with JavaScript code.
- You can write, run, and examine the results of code that interacts with the web page.

https://developer.mozilla.org/en-US/docs/Tools

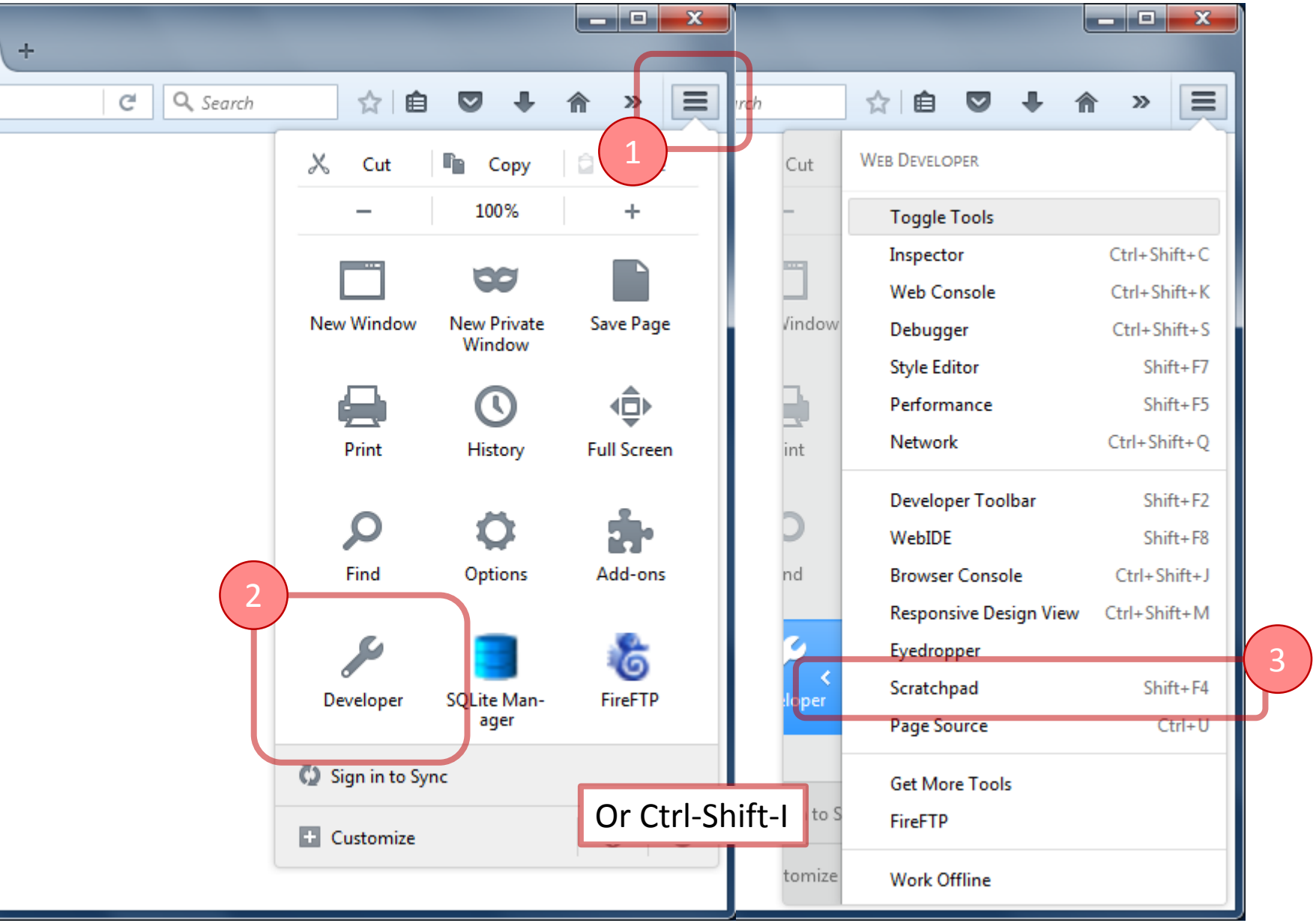
Firefox Developer Tools



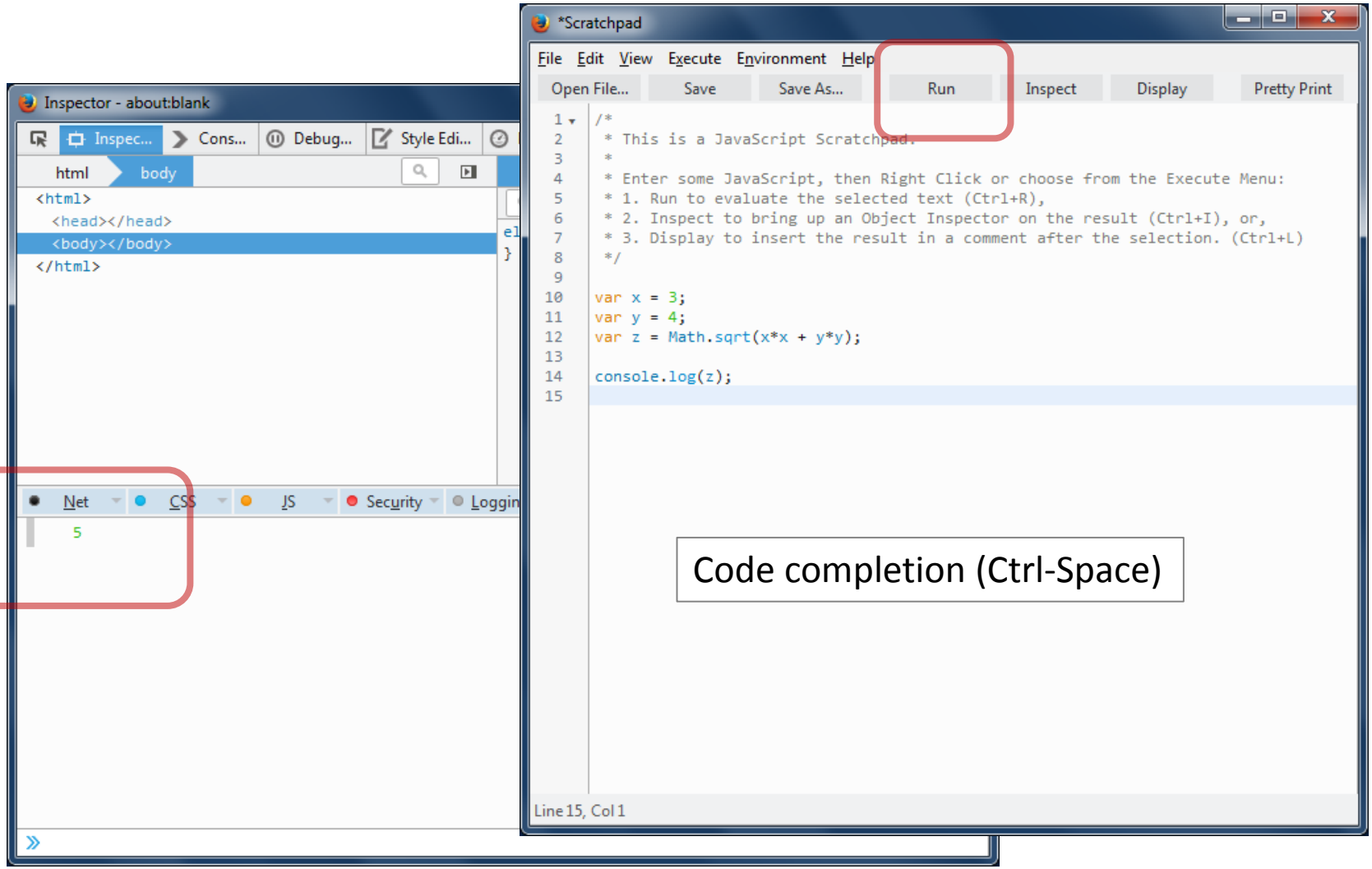
Firefox Developer Tools



Scratchpad


1
Cut
Copy
100%
New Window
New Private Window
Save Page
Print
History
Full Screen
Find
Options
Add-ons
2
Developer
SQLite Manager
FireFTP
Sign in to Sync
Customize
Or Ctrl-Shift-I
Web Developer
Toggle Tools
Inspector Ctrl+Shift+C
Web Console Ctrl+Shift+K
Debugger Ctrl+Shift+S
Style Editor Shift+F7
Performance Shift+F5
Network Ctrl+Shift+Q
Developer Toolbar Shift+F2
WebIDE Shift+F8
Browser Console Ctrl+Shift+J
Responsive Design View Ctrl+Shift+M
Eyedropper
Scratchpad Shift+F4
Page Source Ctrl+U
Get More Tools
FireFTP
Work Offline
3

ScratchPad



https://developer.mozilla.org/en-US/docs/Tools/Scratchpad

Scratchpad - Keyboard shortcuts

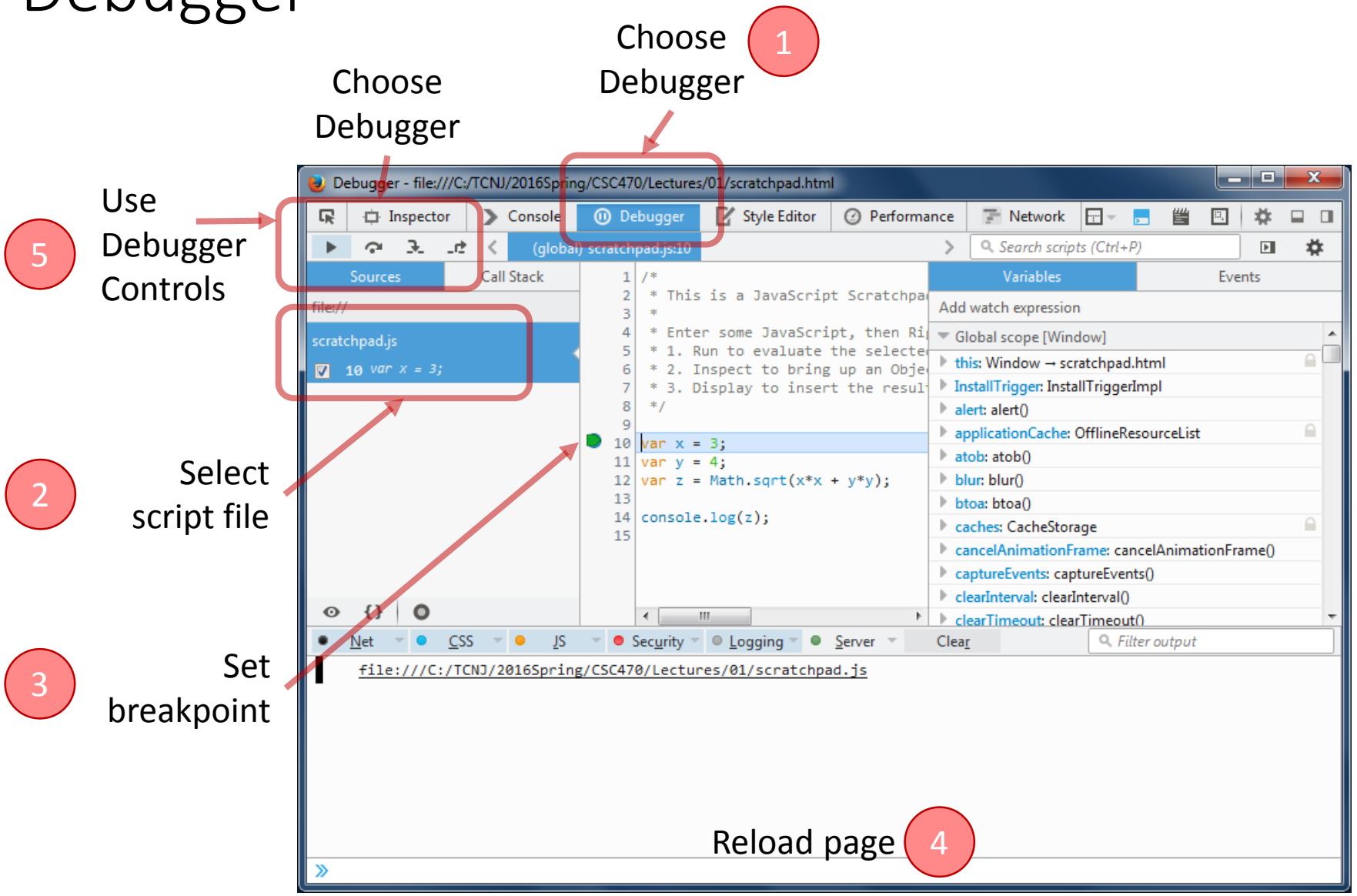
Command	Windows	OS X	Linux
Open the Scratchpad	Shift + F4	Shift + F4	Shift + F4
Run Scratchpad code	Ctrl + R	Cmd + R	Ctrl + R
Run Scratchpad code, display the result in the object inspector	Ctrl + I	Cmd + I	Ctrl + I
Run Scratchpad code, insert the result as a comment	Ctrl + L	Cmd + L	Ctrl + L
Re-evaluate current function	Ctrl + E	Cmd + E	Ctrl + E
Reload the current page, then run Scratchpad code	Ctrl + Shift + R	Cmd + Shift + R	Ctrl + Shift + R
Save the pad	Ctrl + S	Cmd + S	Ctrl + S
Open an existing pad	Ctrl + O	Cmd + O	Ctrl + O
Create a new pad	Ctrl + N	Cmd + N	Ctrl + N
Close Scratchpad	Ctrl + W	Cmd + W	Ctrl + W
Pretty print the code in Scratchpad	Ctrl + P	Cmd + P	Ctrl + P
Show autocomplete suggestions	Ctrl + Space	Ctrl + Space	Ctrl + Space
Show inline documentation	Ctrl + Shift + Space	Ctrl + Shift + Space	Ctrl + Shift + Space

Test Page

- In order to use the debugger, your script must be part of an HTML page.
- Save program in Scratchpad as `scratchpad.js`
- Create new text file and save in same place named `scratchpad.html`
- Open HTML page in browser and open developer tools.
- Select Debugger

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Scratchpad</title>
    <script type="text/javascript" src="scratchpad.js"></script>
  </head>
  <body>
  </body>
</html>
```

Debugger


Choose Debugger
1
Choose Debugger
Use Debugger Controls
5
Select script file
2
Set breakpoint
3
Reload page
4
Debugger - file:///C:/TCNJ/2016Spring/CSC470/Lectures/01/scratchpad.html
Inspector
Console
Debugger
Style Editor
Performance
Network
Search scripts (Ctrl+P)
Sources
Call Stack
Variables
Events
Add watch expression
Global scope [Window]
scratchpad.js
10 var x = 3;
var x = 3;
var y = 4;
var z = Math.sqrt(x*x + y*y);
console.log(z);
Net
CSS
JS
Security
Logging
Server
Clear
Filter output
file:///C:/TCNJ/2016Spring/CSC470/Lectures/01/scratchpad.js