

**Chrome DevTools – Quick Guide**



Juha Hinkula, (Juhani Välimäki)  
 Haaga-Helia  
 Information Technology  
 2015-2016

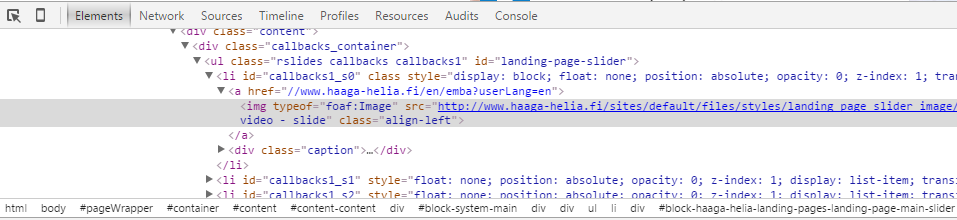
# **Introduction**

Chrome DevTools allow you to inspect and debug HTML, CSS and JavaScript. DevTools are built into Chrome browser.

Start the Chrome DevTools by clicking **F12** button when Chrome browser is running. Or select ‘More Tools 🡪 Developer Tools’ from the Chrome menu.

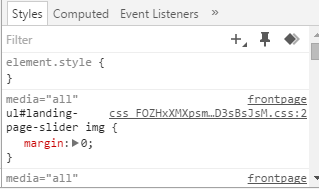
# **Inspect & Edit Elements**

Elements tab can be used to inspect and edit HTML/CSS of the web page.



Press ‘Inspect Element’- icon  and then click on element to inspect it. The other way is to right click any element on the page and select ‘Inspect Element / Tarkastele elementtiä’.

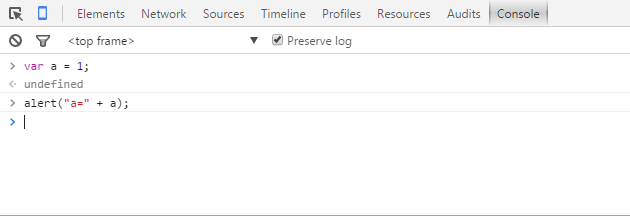
The Styles tab shows the CSS selectors that match and the style rules that apply to the selected element. You can edit property names or property values by clicking on it and see the changes at live.

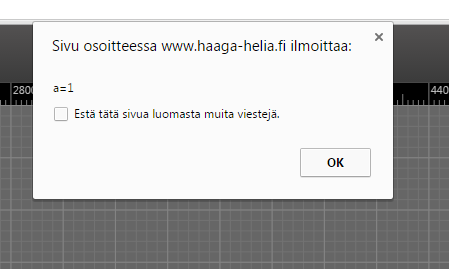


All live edits are **lost** when you reload the page.

# **Console**

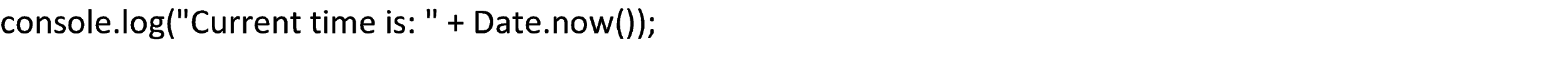
Console tab can be used to log diagnostic information during the development and enter commands that interact with the web page. The Console lets you use standard JavaScript statements and Console-specific commands while a page is live in the browser to help you debug the page.



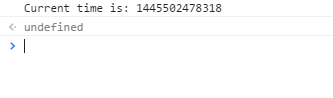


Console logging is excellent way to inspect your JavaScript program functionalities.

For basic logging to the console use console.log(). For example:



Output in the console is:

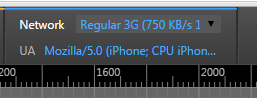


# **Device  Mode**

With the Device Mode (Press the icon:  or Ctrl+Shift+M) you can simulate different devices and check that your web pages are fully responsive to screen sizes and orientations.



You can also evaluate your web site performance with different network conditions.



# **Debugger**

(By JV)

Open the sources tab (means JavaScript source code files). Use the following actions:

* Add/remove breakpoints by clicking on the left margin beside the code lines. You need at least one breakpoints otherwise the code will never stop for you to see what happens.
* Resume / Continue to the next breakpoint by F8
* Step over means run some function without going into it to see what happens there
* Step in / Step into means run the function and also go inside of it to see what happens there.
* Step out. You are in some function call, but are not any more interested in the rest of the function and want to run those lines fast and stop after the function call.
* Start over (self-explanatory, starts from beginning without any changes that the code did earlier)
* Hovering on top of some variable will show what is the value of it at this point.
* Add Watch – You are interested in some variable or collection. Adding a watch for it will make its value to be visible all the time, even without hovering on the top of it

# **Reference & more info**

https://developers.google.com/web/tools/chrome-devtools