Debugging JavaScript

In this lesson, we will commence our debugging journey in JavaScript. Let's begin!

WE'LL COVER THE FOLLOWING

- Debugging JavaScript with Chrome dev tools
 - Step 1:
 - Step 2:
 - Step 3:
 - Step 4:
 - Step 5:

Debugging JavaScript with Chrome dev tools

To discover the simplicity of JavaScript debugging in Chrome, follow these steps:

Step 1:

Display your page in Chrome (the page you created in the previous exercise), and turn on the **Developer Tools** % pane (with F12 (press fn key on Mac) or Ctrl+Shift+I or Cmd+Opt+J for Mac). Select the Source tab and in the source file hierarchy, click (index). To the right from the source file hierarchy, the index.html file is displayed. Scroll down to the var mark line, and click the line number to set a breakpoint, as shown in the figure below.

```
Elements
                   Console
                              Sources Network
                                                 Timeline Profiles
                                                                              Security
                                                                                       Audits
                                                                   Resources
                                   (index) x
Sources Content scr... Snippets
                                    27
                                              $('h1').click(
▼ 🔲 top
                                    28
                                                  function () {
 ▼ △ 127.0.0.1:8080
                                    29
                                                    var node = $(this).children('.node');
                                    30
                                                    $(this).next().fadeToggle(500, 'swing',
   node_modules/jquery/dist
                                                         function () (
     (index)
                                                          var mark = node.text();
                                                          mark = mark ===
                                    33
     style.css
                                    34
                                                          node.text(mark);
                                    35
                                    36
                                                  })
                                    37
                                            });
                                    38
                                          </script>
                                    39
                                    40
                                   {}
                                        Line 1, Column 1
```

Setting a breakpoint on JavaScript code

Step 2:

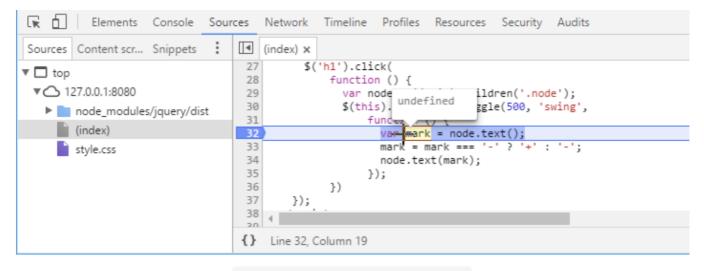
Click one of the first-level headings. As soon as the nested content is faded out, Chrome suspends the code execution because the execution flow reaches the breakpoint you set. This is indicated by the border around the shaded code line, as shown in the figure below.

```
Elements Console
                              Sources Network
                                                  Timeline Profiles Resources Security Audits
Sources Content scr... Snippets
                                    (index) x
                                     27
                                               $('h1').click(
▼ □ top
                                                    function () {
                                     28
 ▼ △ 127.0.0.1:8080
                                     29
                                                      var node = $(this).children('.node');
                                     30
                                                      $(this).next().fadeToggle(500, 'swing',
   node_modules/jquery/dist
                                                          function () {
  var mark = node.text();
                                      31
      (index)
                                                            mark = mark === '-' ? '+'
                                     33
        style.css
                                     34
                                                            node.text(mark);
                                     35
                                                          });
                                     36
                                                   })
                                     37
                                             });
                                     38
                                     20
                                     {}
                                         Line 32, Column 19
```

The execution flows reaches the breakpoint

Step 3:

Move the mouse over the mark variable name in the highlighted line. Chrome shows in a tooltip that its value is undefined (Figure below) because the breakpoint stopped right before the value of the variable has been set.



The value of mark is undefined.

Step 4:

Press F10 (press fn key on Mac) to execute the current line. Now the debugger stops at the next code line. Move the mouse over the mark variable name, and you can see, its initial value is – (Figure below), indicating the expanded state of the first-level heading.

```
Elements Console Sources Network Timeline Profiles Resources Security Audits
                                     (index) x jquery.js
Sources Content scr... Snippets
                                                $('h1').click(
                                      27
▼ 🔲 top
                                      28
                                                     function () {
 ▼ △ 127.0.0.1:8080
                                      29
                                                       var node = $(this).children('.node');
                                                       $(this).nex "_"
                                                                                le(500, 'swing',
                                      30
   node_modules/jquery/dist
                                       31
                                                            functio
                                                             var mark = mark = "-"
mark = mark === '-' ? '+' : '-';
node.text(mark);
      (index)
                                      33
      style.css
                                      34
                                      35
                                                            });
                                                     })
                                      36
                                      37
                                              });
                                      38
                                          Line 33, Column 19
```

The value of mark is set to "-"

Step 5:

Press F10 again and check the value of mark again. Now, it is toggled from - to +.

Amazing work! Now that we have had our short tour of the Holy Trinity that is HTML, CSS, and JavaScript, let's summarize what we've learned in the *next*

lessoit.

See you there!:)