INFO 151
Web Systems and Services

Week 7 (T1)

Dr Philip Moore
Dr Zhili Zhao

Overview Week #7 (T1)

- In this tutorial we extend our introduction to arrays and will introduce:
  - Date and time formatting
  - Generating random numbers
  - JavaScript math methods and math methods for arrays
  - `this` keyword
  - Multi-dimensional arrays

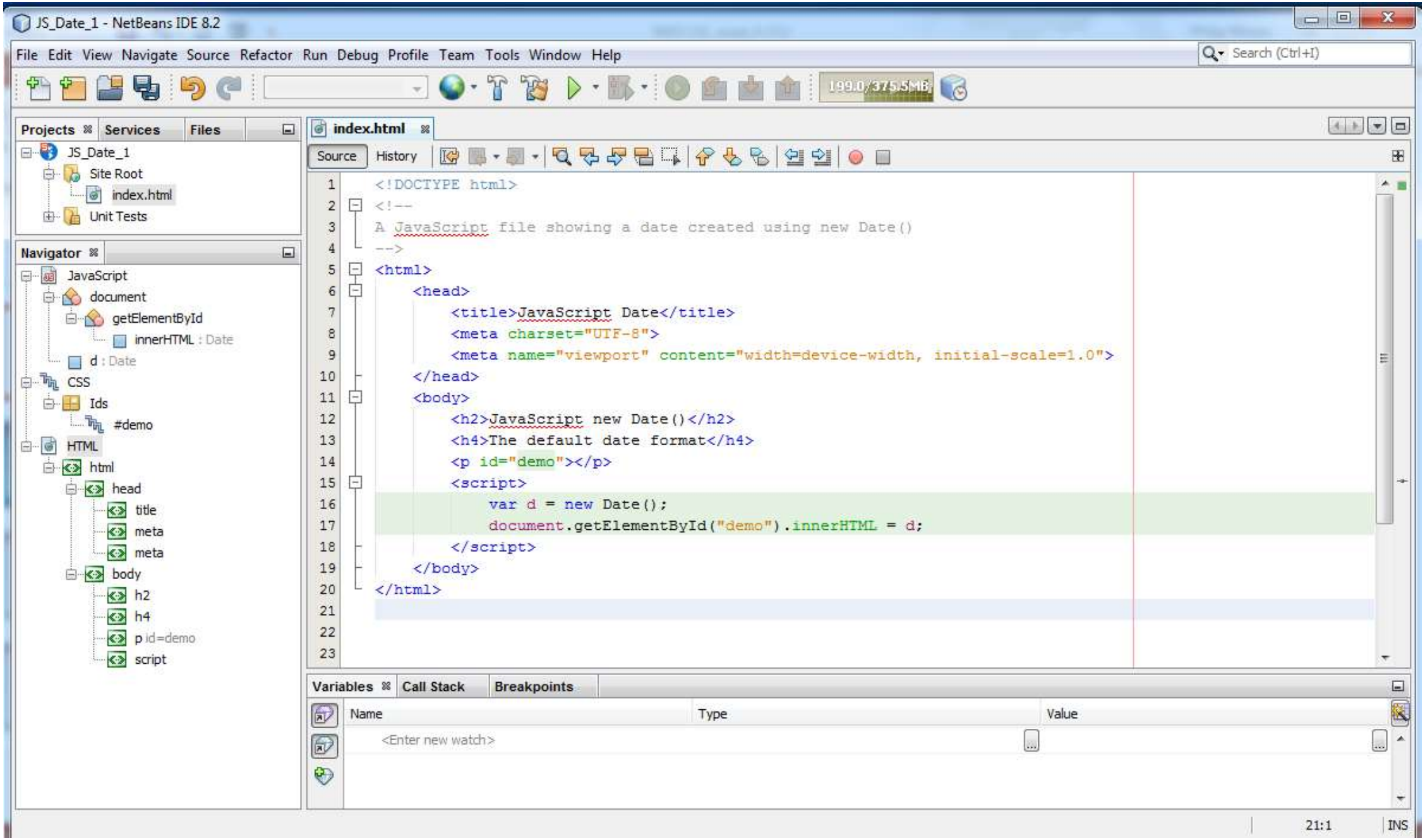
Date and Time Formatting

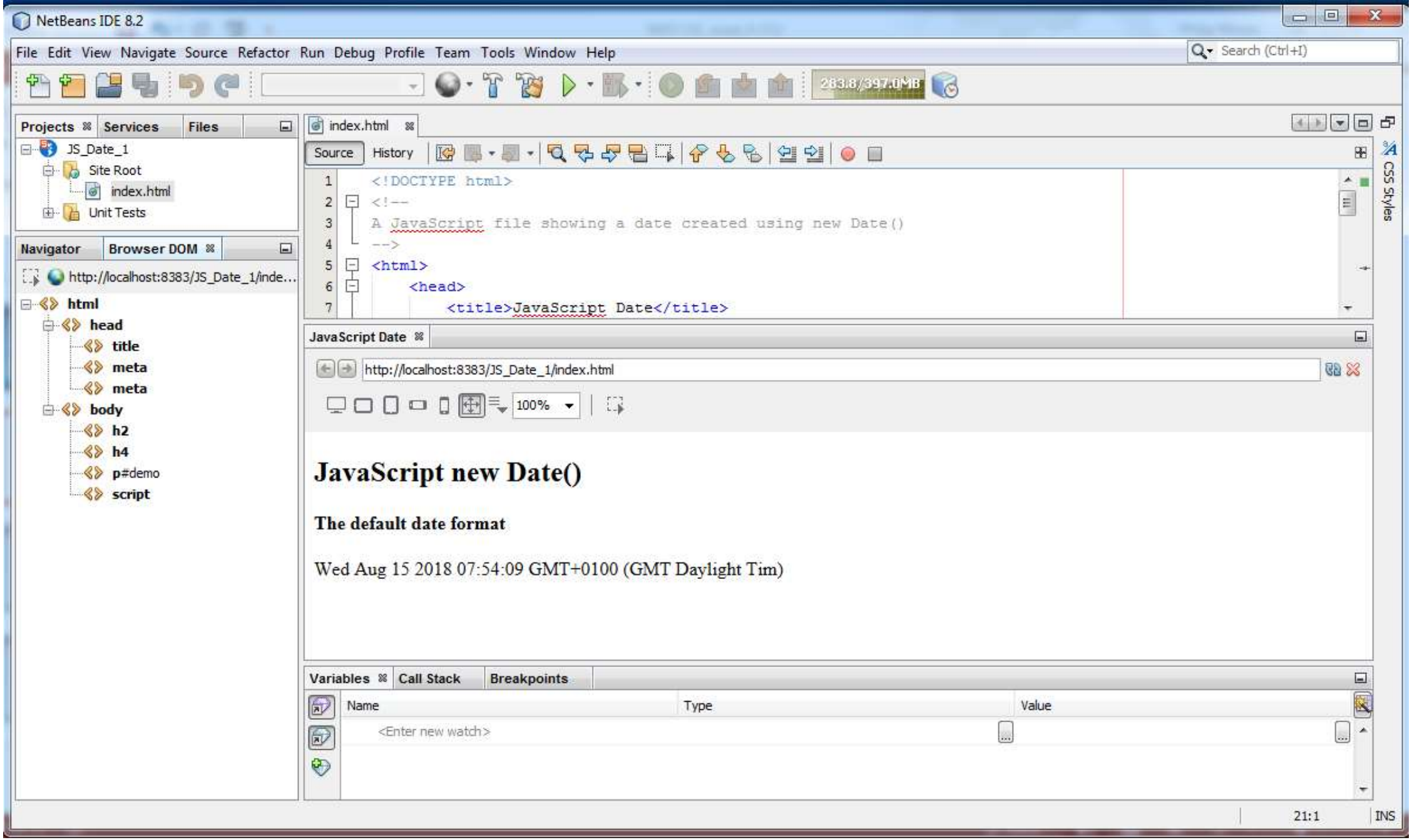
Dates and Date Format

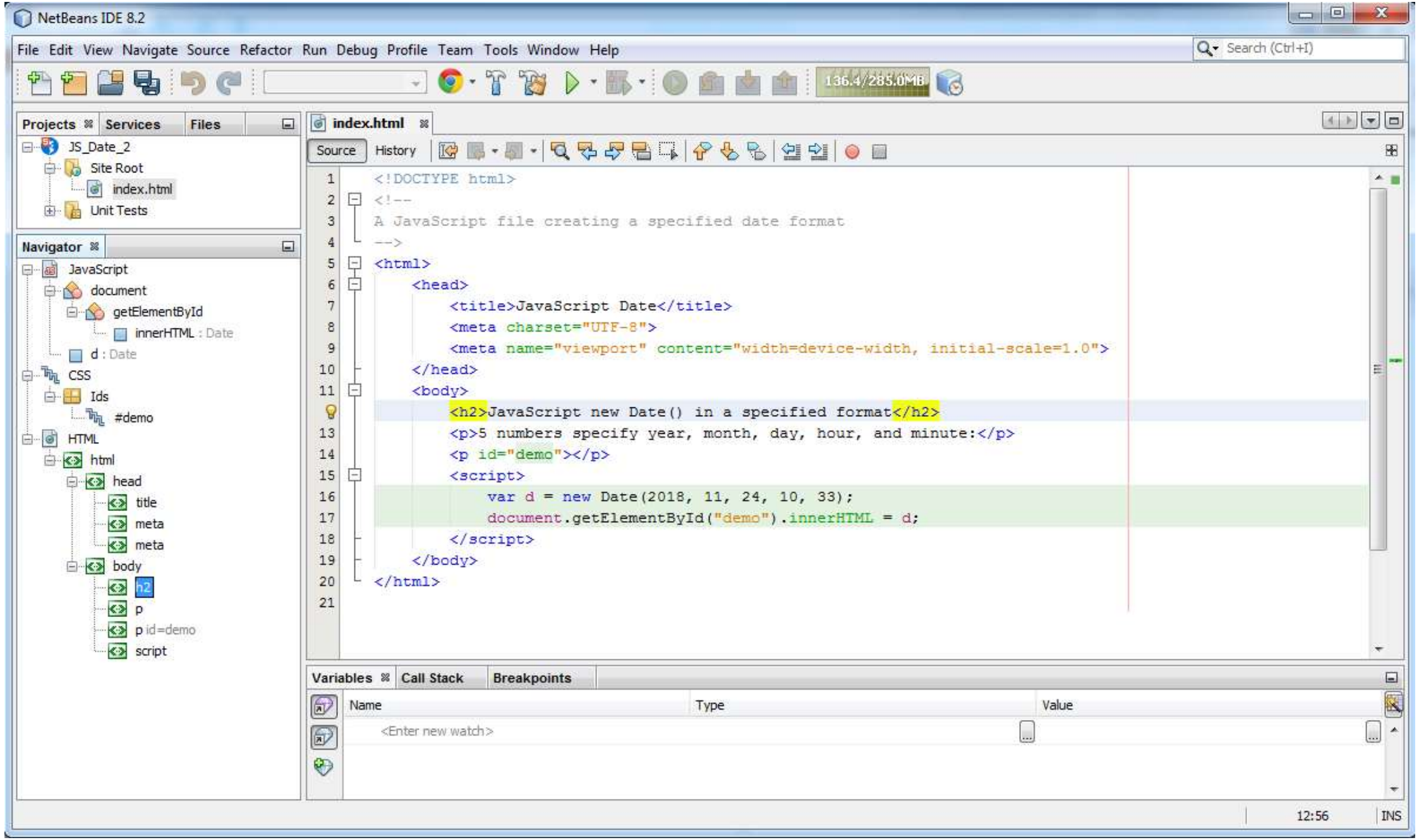
- The JavaScript Date Object operates with dates and time
- The following slides demonstrate dates and the date formatting
- Details of the date methods and formatting may be found in the course resources
- In summary date methods can:
  - Create date objects
  - Format dates and times in a range of formats
  - Create a date String object
  - Create a time in milliseconds
  - Display dates in a String format

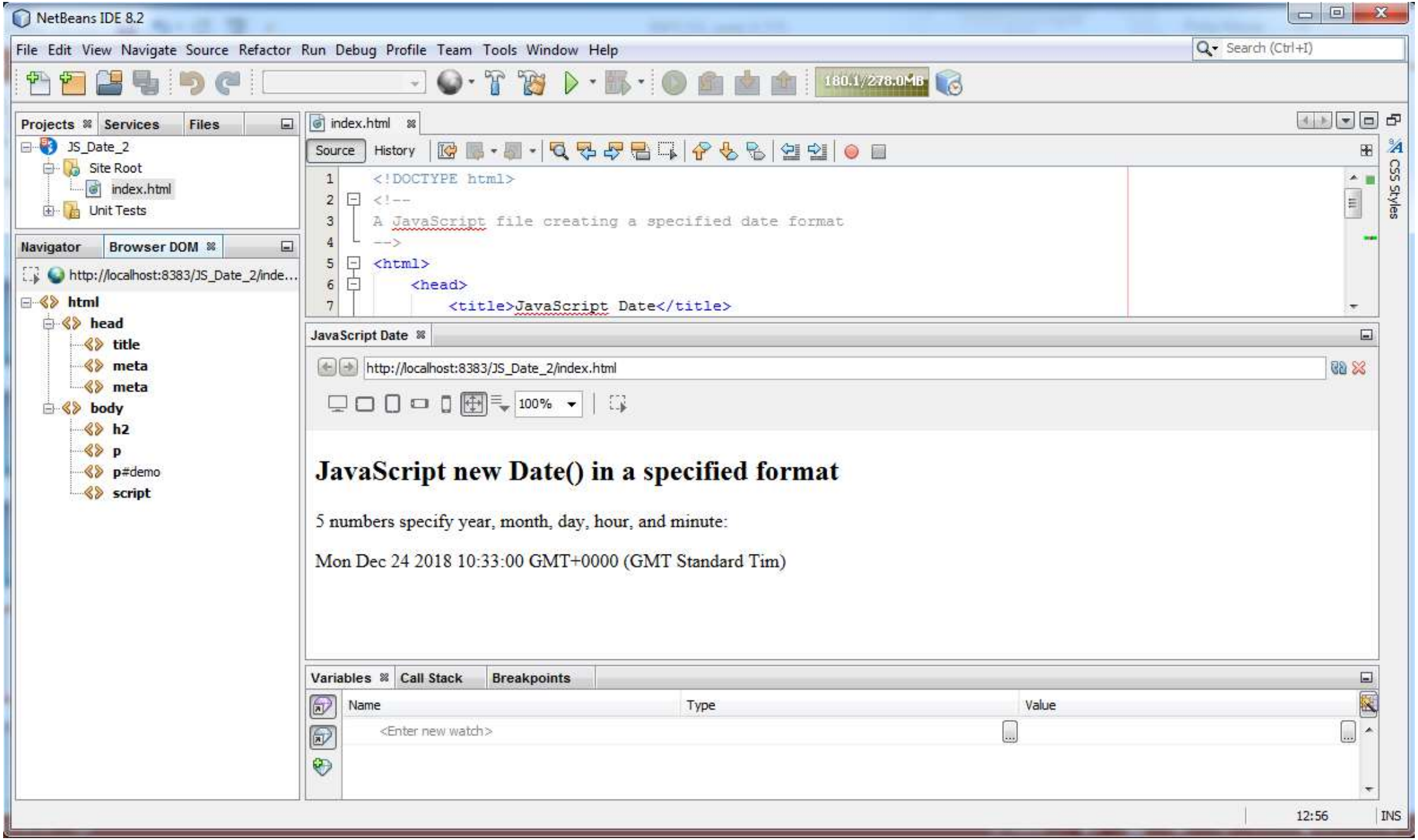
Creating Date Objects

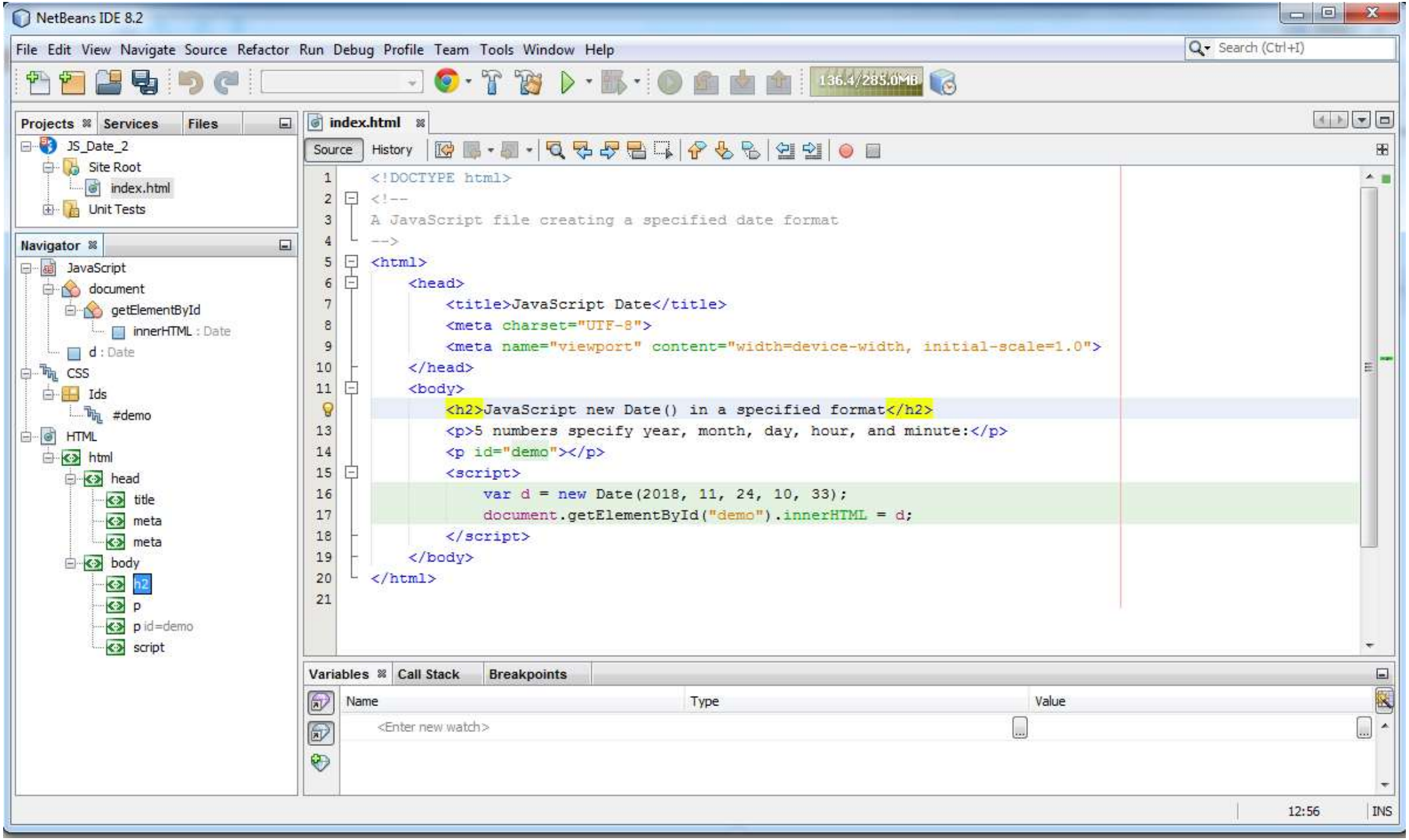
- There are two ways to create a Date Object:
  - Construct an empty date object with the new Date() method
  - Construct a date object with parameters
    - new Date(*year, month, day, hours, minutes, seconds, milliseconds*)
    - new Date(*milliseconds*)
- To create a date object and convert it to a string
  - Var d = new Date();
  - Var n = d.toString();
  - Note: This method is automatically called by JavaScript whenever a Date object needs to be displayed as a string

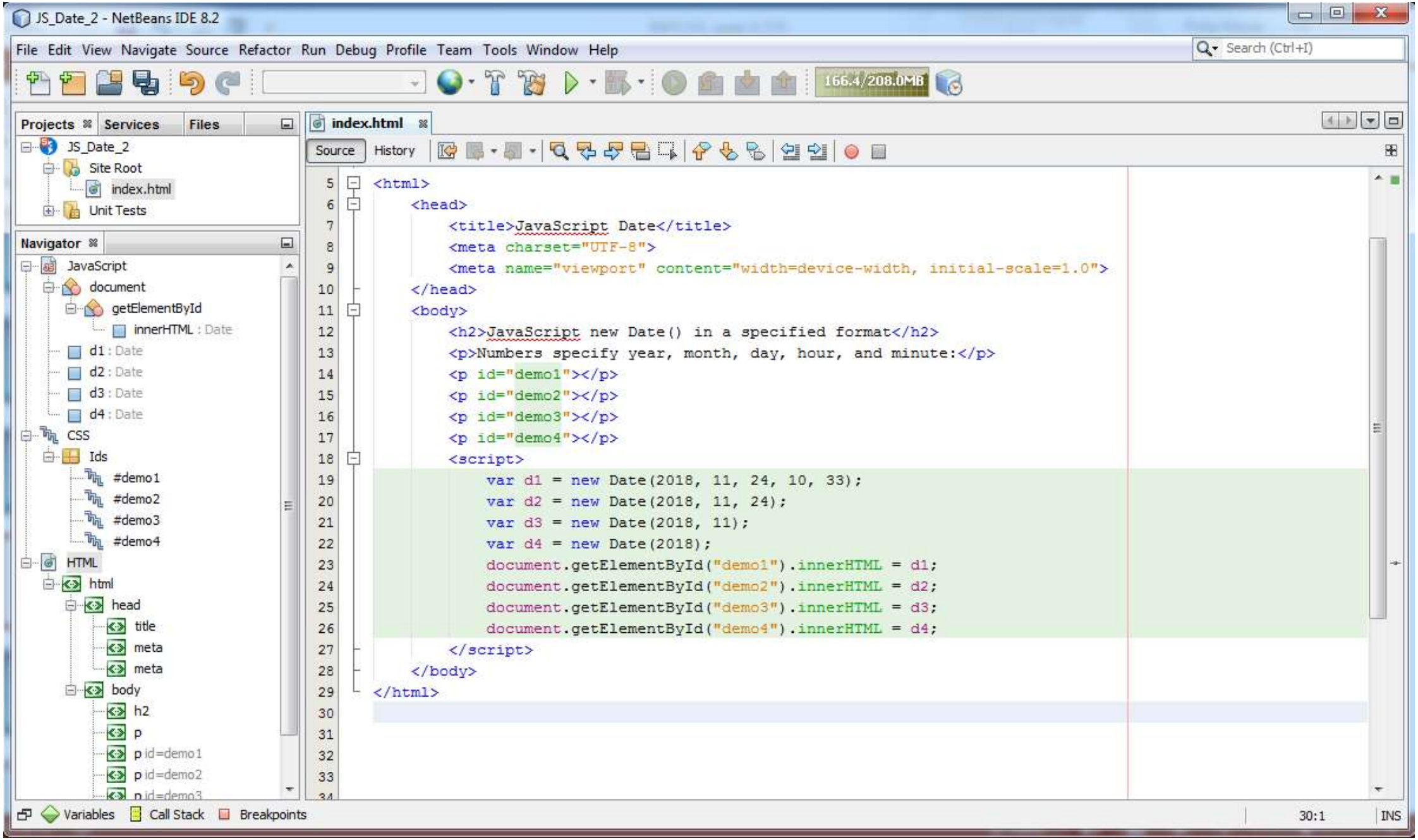

JS_Date_1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
199.0/375.5 MB
Projects
Services
Files
JS_Date_1
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Date
d : Date
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
h4
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file showing a date created using new Date()
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <h2>JavaScript new Date()</h2>
13 <h4>The default date format</h4>
14
15 <script>
16 var d = new Date();
17 document.getElementById("demo").innerHTML = d;
18 </script>
19 </body>
20 </html>
21
22
23
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
21:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Date_1
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_Date_1/inde...
html
head
title
meta
meta
body
h2
h4
p#demo
script
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file showing a date created using new Date()
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date</title>
CSS Styles
JavaScript Date
http://localhost:8383/JS_Date_1/index.html
100%
JavaScript new Date()
The default date format
Wed Aug 15 2018 07:54:09 GMT+0100 (GMT Daylight Tim)
Variables Call Stack Breakpoints
Name Type Value
<Enter new watch>
21:1
INS

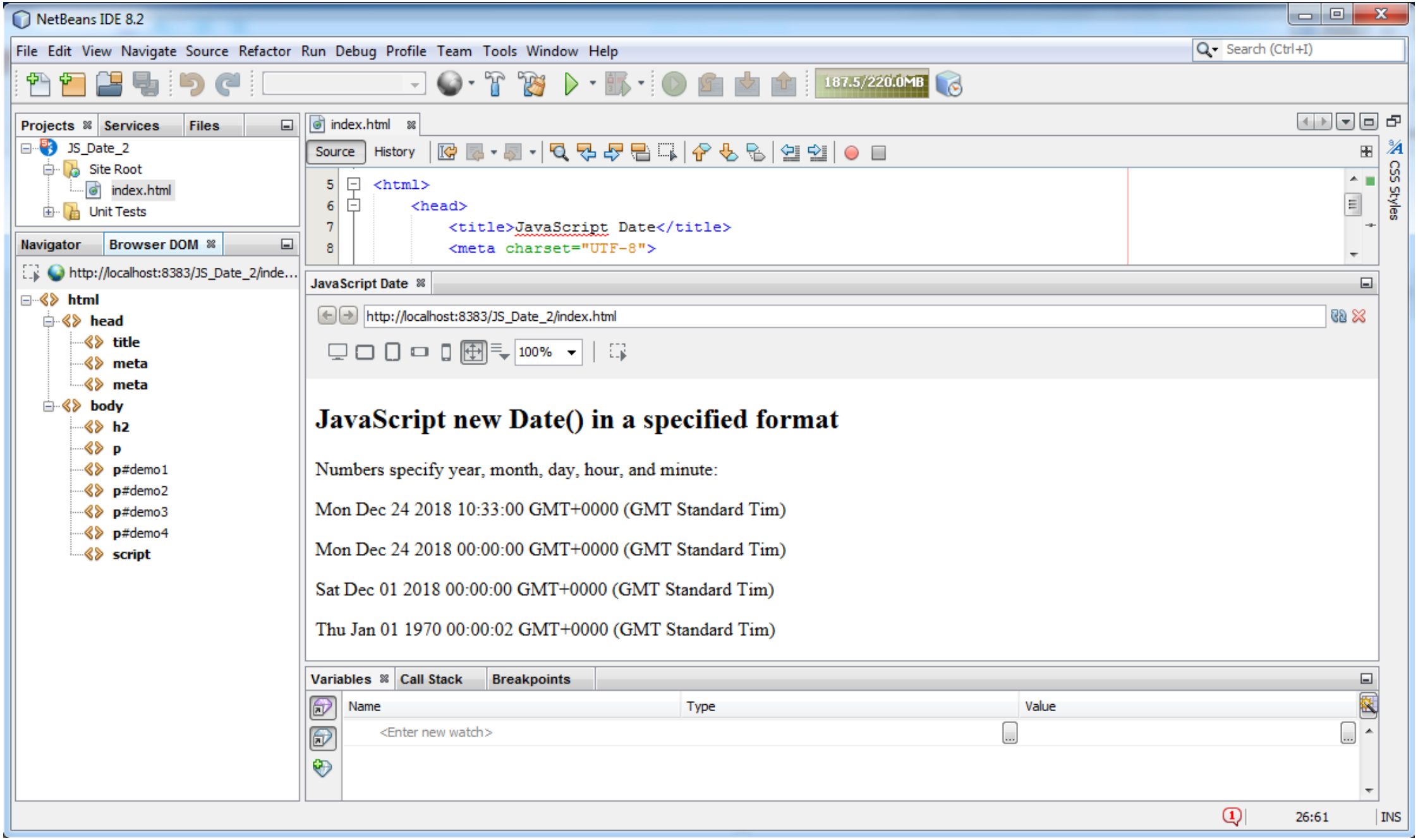

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
136.4/285.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Date
d : Date
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file creating a specified date format
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
<h2>JavaScript new Date() in a specified format</h2>
13 5 numbers specify year, month, day, hour, and minute:
14
15 <script>
16 var d = new Date(2018, 11, 24, 10, 33);
17 document.getElementById("demo").innerHTML = d;
18 </script>
19 </body>
20 </html>
21
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
12:56
INS

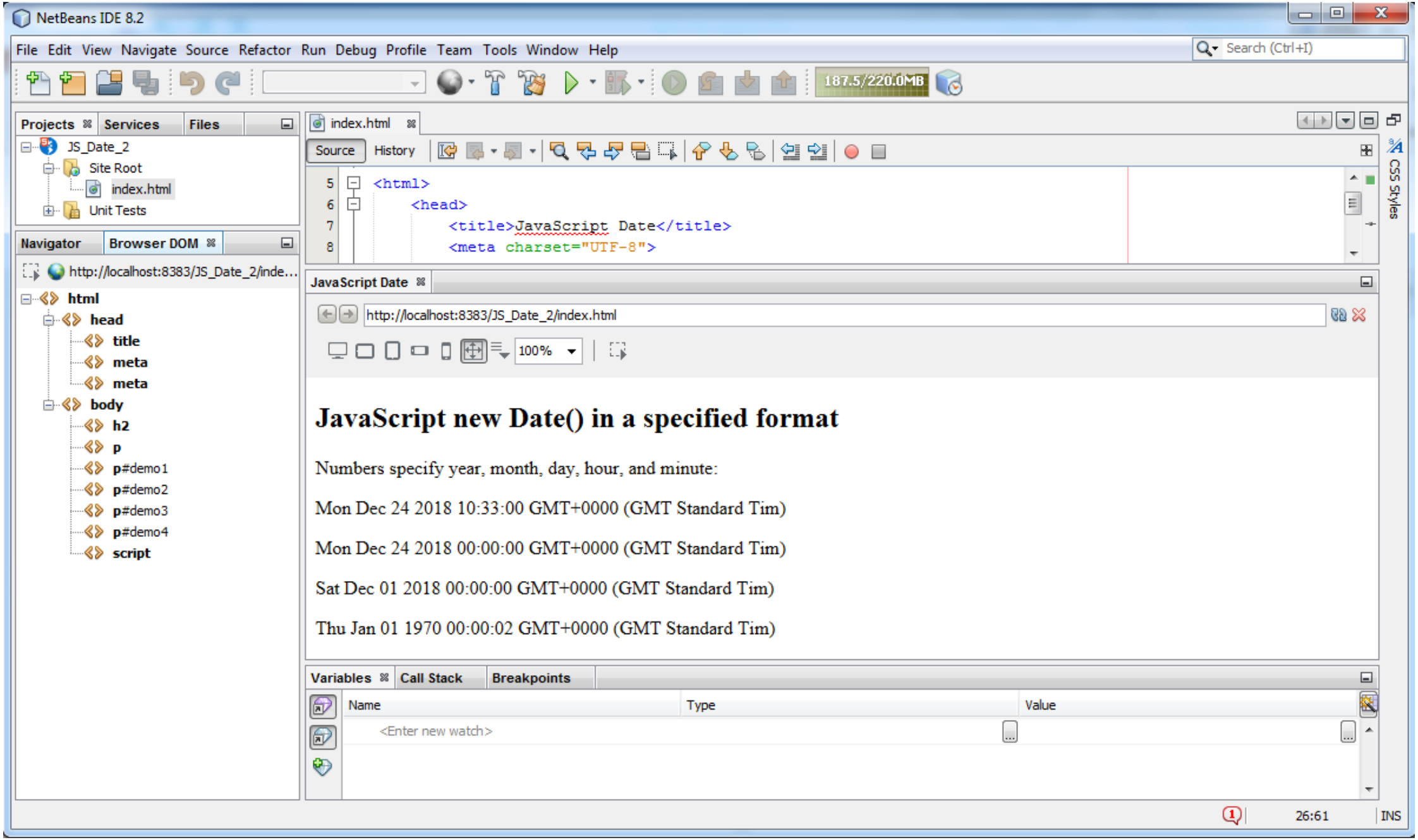

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
180.1/278.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Date_2/inde...
html
head
title
meta
meta
body
h2
p
p#demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file creating a specified date format
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date</title>
CSS Styles
JavaScript Date
http://localhost:8383/JS_Date_2/index.html
100%
JavaScript new Date() in a specified format
5 numbers specify year, month, day, hour, and minute:
Mon Dec 24 2018 10:33:00 GMT+0000 (GMT Standard Tim)
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
12:56
INS

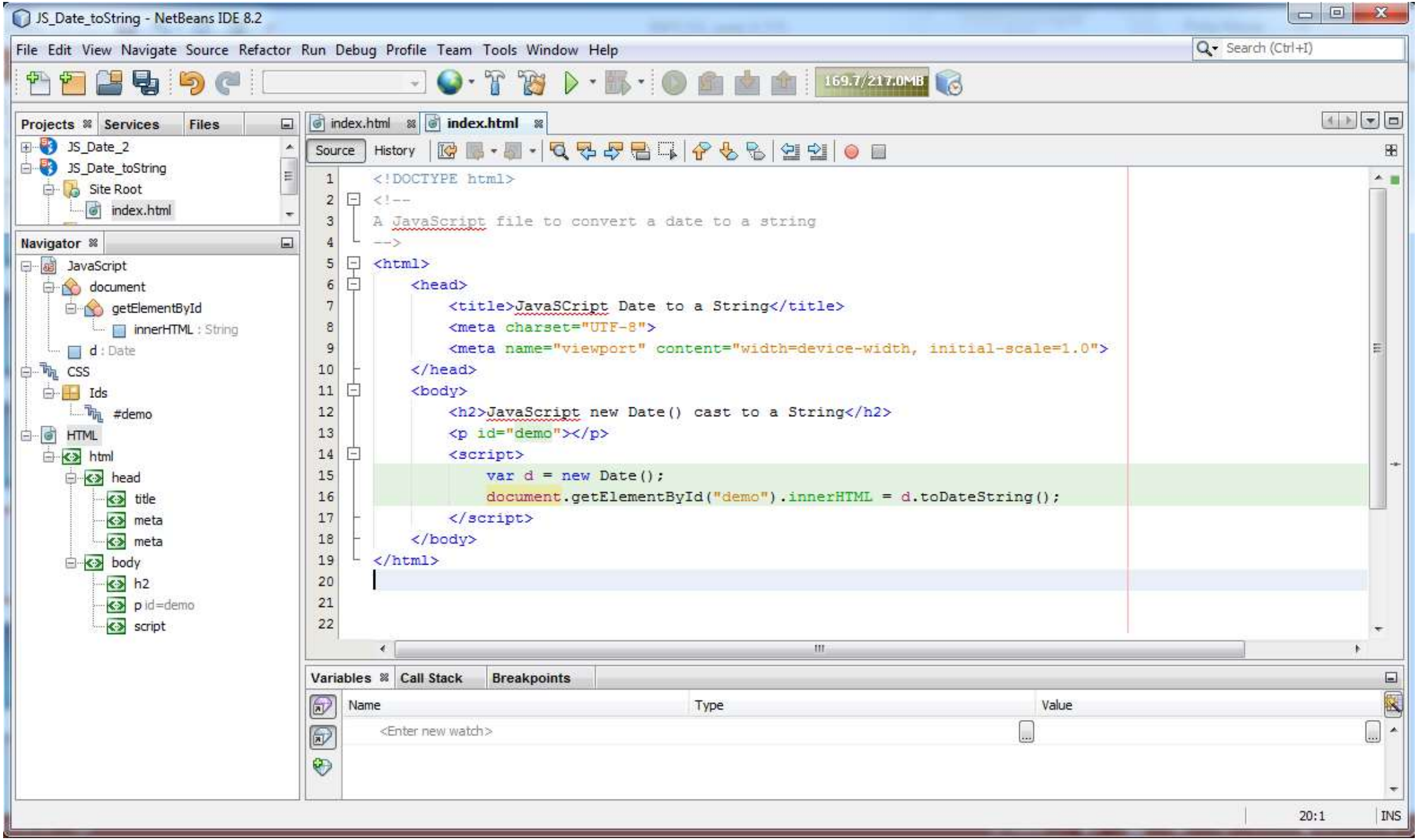

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
136.4/285.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Date
d : Date
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file creating a specified date format
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
<h2>JavaScript new Date() in a specified format</h2>
13 5 numbers specify year, month, day, hour, and minute:
14
15 <script>
16 var d = new Date(2018, 11, 24, 10, 33);
17 document.getElementById("demo").innerHTML = d;
18 </script>
19 </body>
20 </html>
21
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
12:56
INS

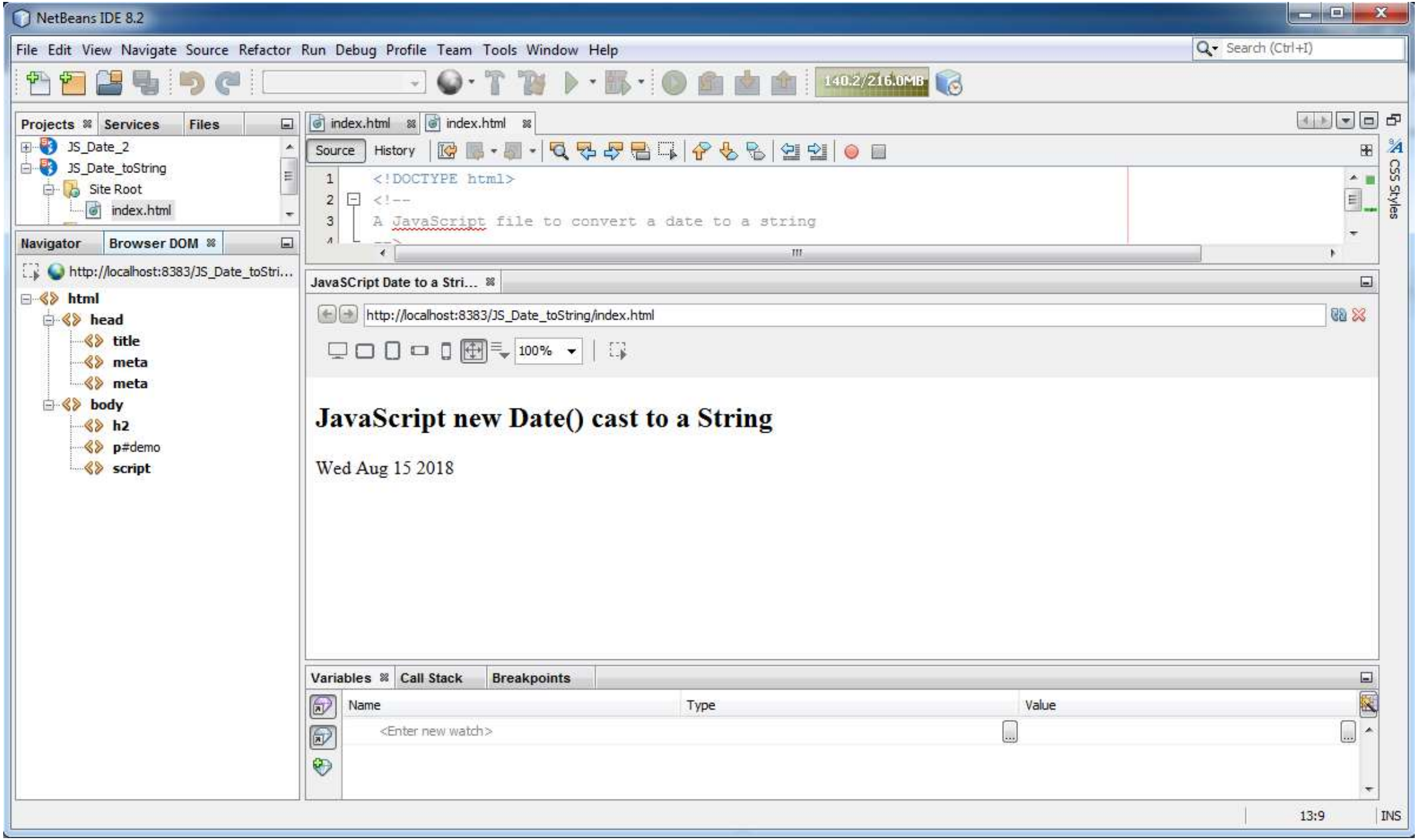

JS_Date_2 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
166.4/208.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Date
d1 : Date
d2 : Date
d3 : Date
d4 : Date
CSS
Ids
#demo1
#demo2
#demo3
#demo4
HTML
html
head
title
meta
meta
body
h2
p
p id=demo1
p id=demo2
p id=demo3
index.html
Source
History
<html>
<head>
<title>JavaScript Date</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h2>JavaScript new Date() in a specified format</h2>
Numbers specify year, month, day, hour, and minute:

<script>
var d1 = new Date(2018, 11, 24, 10, 33);
var d2 = new Date(2018, 11, 24);
var d3 = new Date(2018, 11);
var d4 = new Date(2018);
document.getElementById("demo1").innerHTML = d1;
document.getElementById("demo2").innerHTML = d2;
document.getElementById("demo3").innerHTML = d3;
document.getElementById("demo4").innerHTML = d4;
</script>
</body>
</html>
Variables
Call Stack
Breakpoints
30:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
187.5/220.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Date_2/inde...
html
head
title
meta
meta
body
h2
p
p#demo1
p#demo2
p#demo3
p#demo4
script
index.html
Source
History
5 <html>
6 <head>
7 <title>JavaScript Date</title>
8 <meta charset="UTF-8">
CSS Styles
JavaScript Date
http://localhost:8383/JS_Date_2/index.html
100%
JavaScript new Date() in a specified format
Numbers specify year, month, day, hour, and minute:
Mon Dec 24 2018 10:33:00 GMT+0000 (GMT Standard Tim)
Mon Dec 24 2018 00:00:00 GMT+0000 (GMT Standard Tim)
Sat Dec 01 2018 00:00:00 GMT+0000 (GMT Standard Tim)
Thu Jan 01 1970 00:00:02 GMT+0000 (GMT Standard Tim)
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
26:61
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
187.5/220.0MB
Projects
Services
Files
JS_Date_2
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Date_2/inde...
html
head
title
meta
meta
body
h2
p
p#demo1
p#demo2
p#demo3
p#demo4
script
index.html
Source
History
5 <html>
6 <head>
7 <title>JavaScript Date</title>
8 <meta charset="UTF-8">
CSS Styles
JavaScript Date
http://localhost:8383/JS_Date_2/index.html
100%
JavaScript new Date() in a specified format
Numbers specify year, month, day, hour, and minute:
Mon Dec 24 2018 10:33:00 GMT+0000 (GMT Standard Tim)
Mon Dec 24 2018 00:00:00 GMT+0000 (GMT Standard Tim)
Sat Dec 01 2018 00:00:00 GMT+0000 (GMT Standard Tim)
Thu Jan 01 1970 00:00:02 GMT+0000 (GMT Standard Tim)
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
26:61
INS


JS_Date_toString - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
169.7/217.0MB
Projects
Services
Files
JS_Date_2
JS_Date_toString
Site Root
index.html
Navigator
JavaScript
document
getElementById
innerHTML : String
d : Date
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p id=demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file to convert a date to a string
4 -->
5 <html>
6 <head>
7 <title>JavaScript Date to a String</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <h2>JavaScript new Date() cast to a String</h2>
13
14 <script>
15 var d = new Date();
16 document.getElementById("demo").innerHTML = d.toDateString();
17 </script>
18 </body>
19 </html>
20
21
22
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
20:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
140.2/216.0MB
Projects
Services
Files
JS_Date_2
JS_Date_toString
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_Date_toStri...
html
head
title
meta
meta
body
h2
p#demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file to convert a date to a string
CSS Styles
JavaSCript Date to a Stri...
http://localhost:8383/JS_Date_toString/index.html
100%
JavaScript new Date() cast to a String
Wed Aug 15 2018
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
13:9
INS

Generating Random Numbers

Generating Random Numbers

- In computer science:
  - Random number algorithms generate a *pseudorandom number*
  - In JavaScript a *pseudorandom number generator* is also termed a *deterministic random bit generator*
- Random number algorithms generate a sequence of numbers whose properties *approximate* the properties of sequences of *genuine random numbers*
- In JavaScript a random number may be generated with the following math method:
  - **`Math.random();`** (returns a 64 bit random number)

Generating Random Numbers

- A very useful and often used math method is the creation of random numbers – for example:
  - Consider building a casino application where a dice is rolled
    - How do you simulate it being unbiased?
  - Similarly, a game involving a deck of cards generally begins with the decks shuffled
    - How do you simulate a shuffled deck of cards?

Generating Random Integers

- To create random integers

  `Math.ceil(x)` and **`Math.floor(x)`**

  `Math.ceil(4.4);` //returns 5 – rounded ***up***)

  `Math.floor(4.7);` //returns 4 – rounded ***down***)

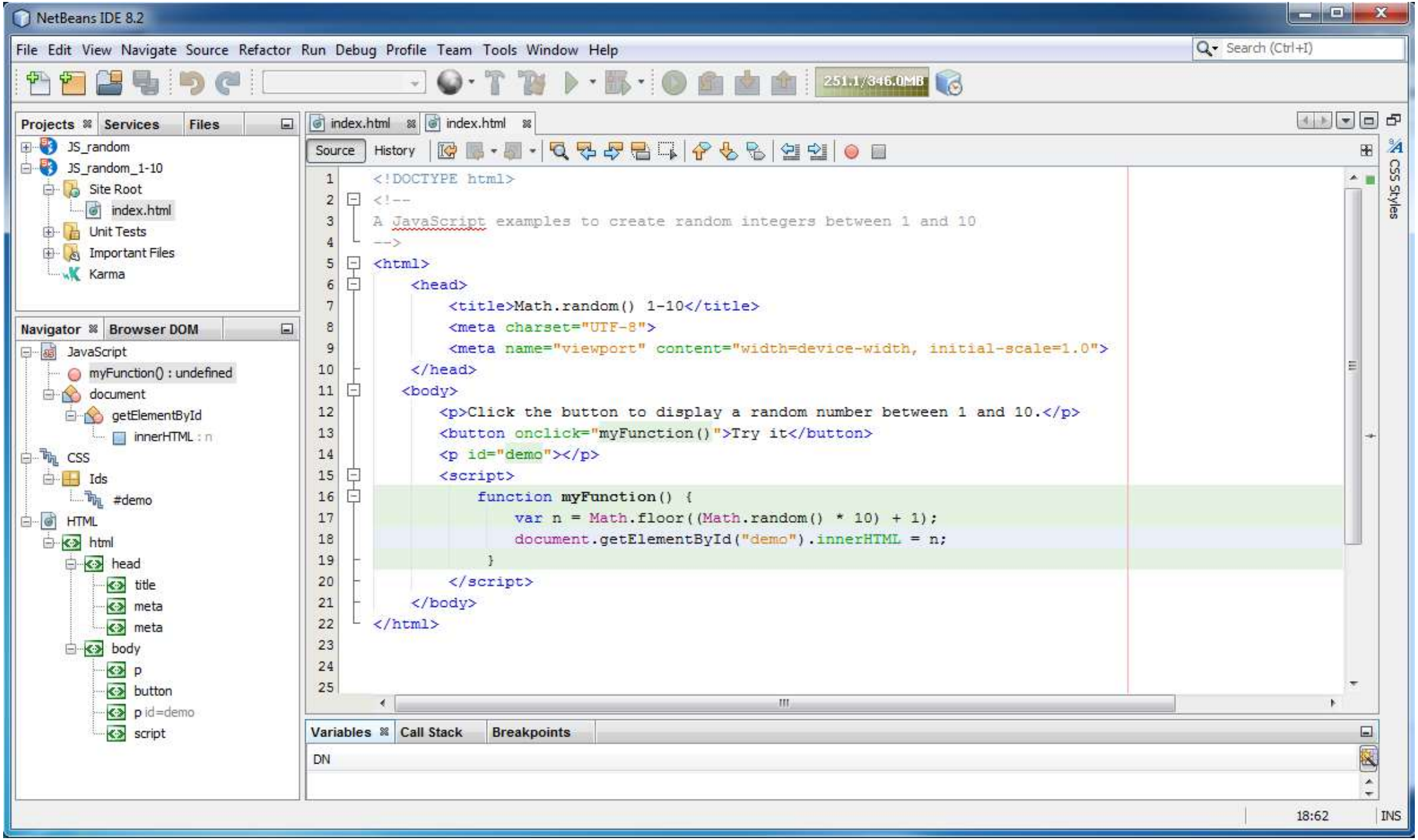
- If a random number within a certain range is required

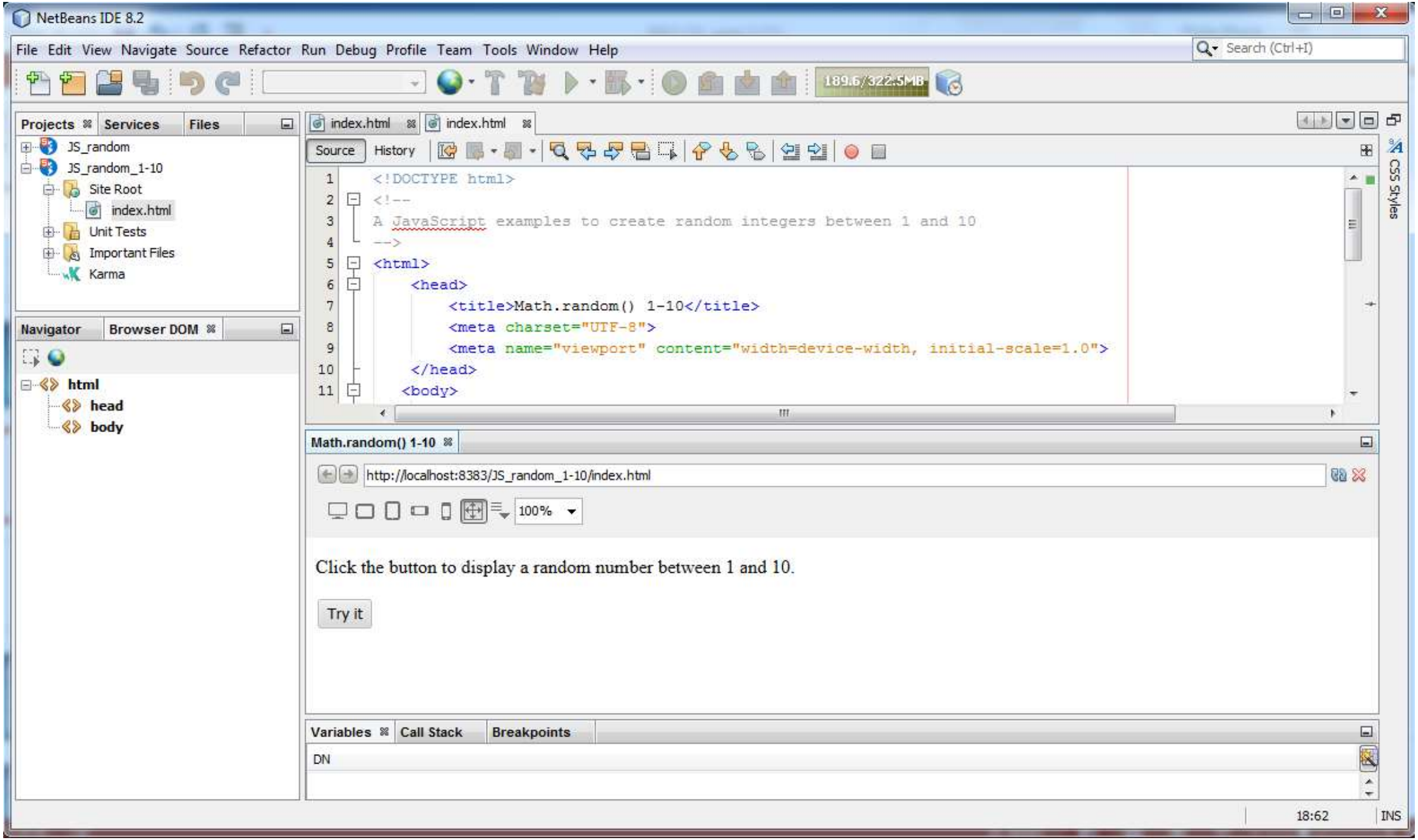
  `Math.floor(Math.random() * 10) + 1;`

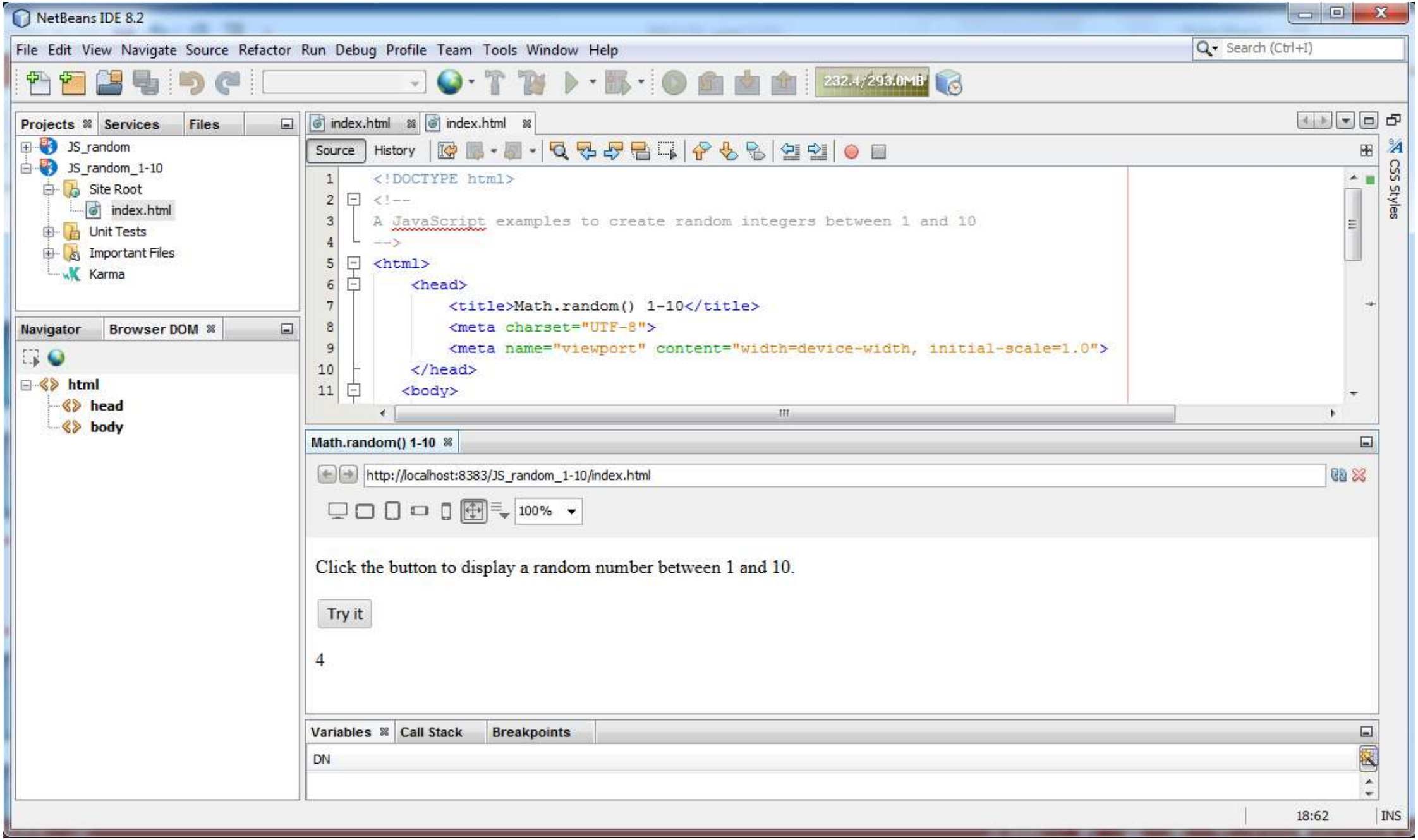
  - (returns a random integer in the range 1 … 10)

  `Math.floor(Math.random() * 10);`

  - Returns a random integer in the range 0 … 9

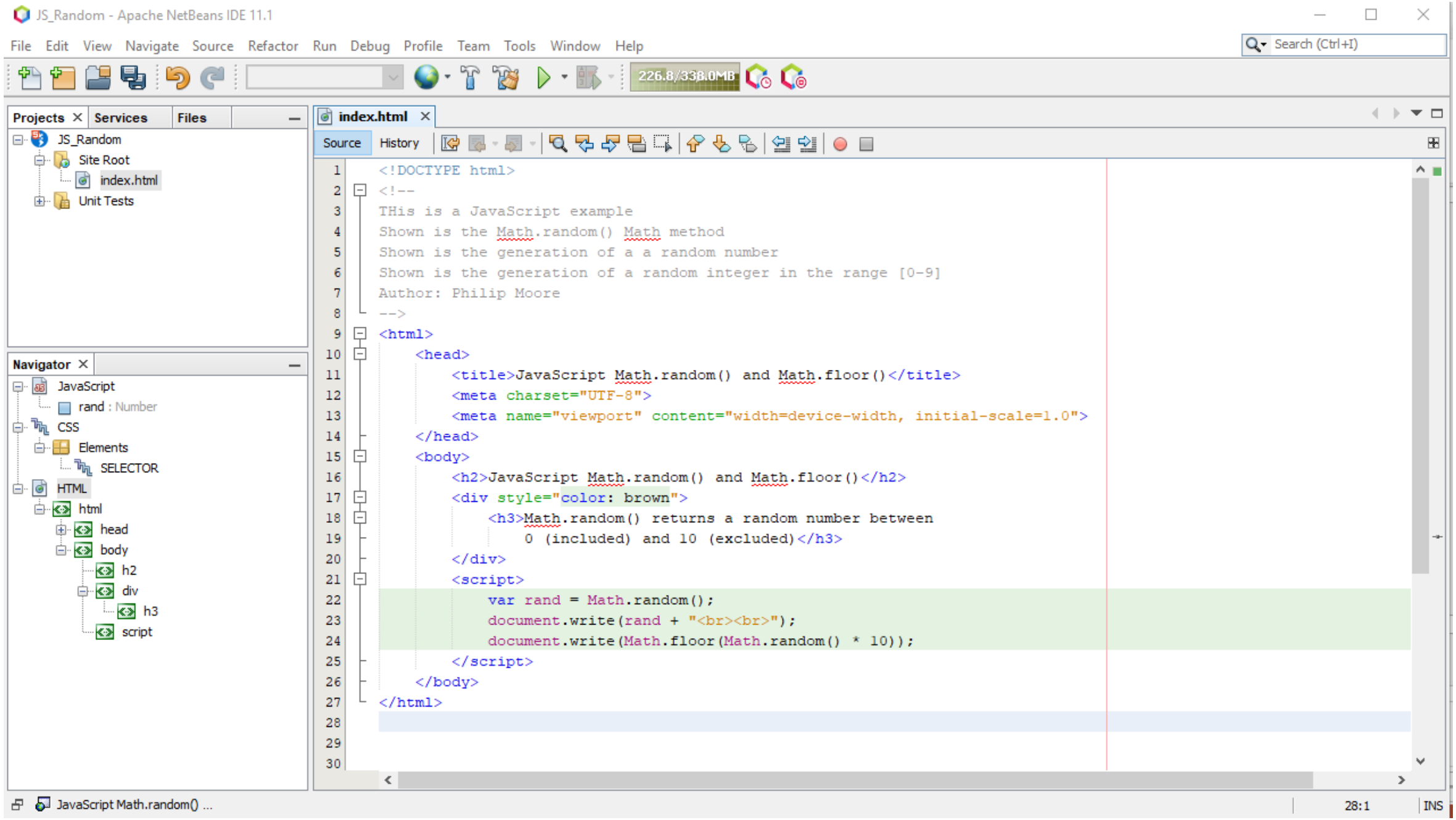

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
251.1/346.0MB
Projects
Services
Files
JS_random
JS_random_1-10
Site Root
index.html
Unit Tests
Important Files
Karma
Navigator
Browser DOM
JavaScript
myFunction() : undefined
document
getElementById
innerHTML : n
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
p
button
p id=demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript examples to create random integers between 1 and 10
4 -->
5 <html>
6 <head>
7 <title>Math.random() 1-10</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 Click the button to display a random number between 1 and 10.
13 <button onclick="myFunction()">Try it</button>
14
15 <script>
16 function myFunction() {
17 var n = Math.floor((Math.random() * 10) + 1);
18 document.getElementById("demo").innerHTML = n;
19 }
20 </script>
21 </body>
22 </html>
23
24
25
CSS Styles
Variables
Call Stack
Breakpoints
DN
18:62
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
189.6/322.5MB
Projects
Services
Files
JS_random
JS_random_1-10
Site Root
index.html
Unit Tests
Important Files
Karma
Navigator
Browser DOM
html
head
body
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript examples to create random integers between 1 and 10
4 -->
5 <html>
6 <head>
7 <title>Math.random() 1-10</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
CSS Styles
Math.random() 1-10
http://localhost:8383/JS_random_1-10/index.html
100%
Click the button to display a random number between 1 and 10.
Try it
Variables
Call Stack
Breakpoints
DN
18:62
INS

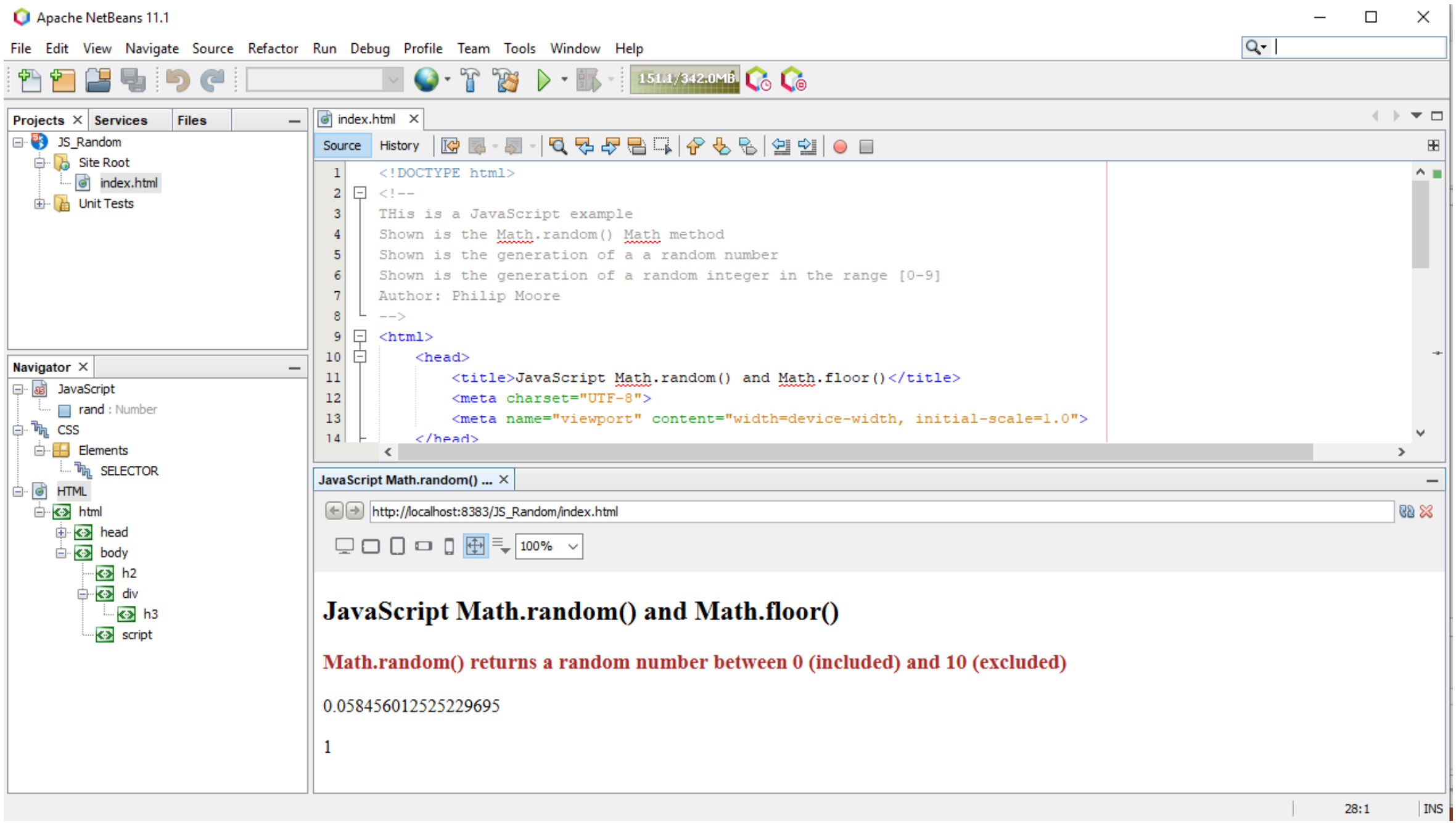

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
232.4/293.0MB
Projects
Services
Files
JS_random
JS_random_1-10
Site Root
index.html
Unit Tests
Important Files
Karma
Navigator
Browser DOM
html
head
body
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript examples to create random integers between 1 and 10
4 -->
5 <html>
6 <head>
7 <title>Math.random() 1-10</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
CSS Styles
Math.random() 1-10
http://localhost:8383/JS_random_1-10/index.html
100%
Click the button to display a random number between 1 and 10.
Try it
4
Variables
Call Stack
Breakpoints
DN
18:62
INS

Math.random()

- The following slides show
  - The generation of a 64 bit random number
  - The generation of a random integer in the range in a defined range
- The following example returns a 64 bit random number (a double)
  - `math.random()`
- The following JavaScript returns a random number between 0 (included) and 10 (excluded)
  - `Math.floor(Math.random() * 10);`
- The following JavaScript returns a random number between 1 and 10 (both included)
  - `Math.floor((Math.random() * 10 + 1));`


JS_Random - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
226.8/338.0MB
Projects
Services
Files
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
rand : Number
CSS
Elements
SELECTOR
HTML
html
head
body
h2
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 THis is a JavaScript example
4 Shown is the Math.random() Math method
5 Shown is the generation of a a random number
6 Shown is the generation of a random integer in the range [0-9]
7 Author: Philip Moore
8 -->
9 <html>
10 <head>
11 <title>JavaScript Math.random() and Math.floor()</title>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 </head>
15 <body>
16 <h2>JavaScript Math.random() and Math.floor()</h2>
17
18 <h3>Math.random() returns a random number between
19 0 (included) and 10 (excluded)</h3>
20
21 <script>
22 var rand = Math.random();
23 document.write(rand + "

");
24 document.write(Math.floor(Math.random() * 10));
25 </script>
26 </body>
27 </html>
28
29
30
JavaScript Math.random() ...
28:1
INS


Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
151.1/342.0MB
Projects Services Files
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
rand : Number
CSS
Elements
SELECTOR
HTML
html
head
body
h2
div
h3
script
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 THis is a JavaScript example
4 Shown is the Math.random() Math method
5 Shown is the generation of a a random number
6 Shown is the generation of a random integer in the range [0-9]
7 Author: Philip Moore
8 -->
9 <html>
10 <head>
11 <title>JavaScript Math.random() and Math.floor()</title>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 </head>
JavaScript Math.random() ...
http://localhost:8383/JS_Random/index.html
100%
JavaScript Math.random() and Math.floor()
Math.random() returns a random number between 0 (included) and 10 (excluded)
0.058456012525229695
1
28:1
INS

Generating Random Numbers

- The following slide shows:
  - A practical application of the JavaScript `math.random()` method
  - The program generates an array of random integers in the range [0…100] included and an array of floating point numbers
  - The JavaScript method `Math.floor((Math.random() * 100 + 1))` creates the random integers
- Note:
  - The for loop to iterate over the array
  - The assignment of the random number to the array element

bubble_sort.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Bubble Sort</title>
</head>
<body>

  <p>A basic bubble sort implementation</p>

  <script>
    var arr = [];
    var i;
    document.write("original array of random numbers <br>");
    for(i = 0; i <=20; i++) {
            //var x = (Math.random() * 100 + 1);
            var x = (Math.floor(Math.random() * 100 + 1));
            arr[i] = x;
            document.write(arr[i] + " *  ");
        }
    document.write("<br>array.length: " + arr.length + "<br>");
    document.write("unsorted array <br>");
    document.write(arr + "<br>");
    var n = arr.length;
    var temp = 0;
    for(var i=0; i < n; i++){
        for(var j=1; j < (n-i); j++){
            if(arr[j-1] > arr[j]){
                //swap elements
                temp = arr[j-1];
                arr[j-1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    document.write("sorted array");
    document.write("<br>" + arr);
  </script>

</body>
</html>
```

```
U:---  bubble_sort.html   All L15    (HTML+JS)
Wrote /Users/philip/Desktop/bubble_sort.html
```

file:///Users/philip/Desktop/bubble_s

A basic bubble sort implementation

original array of random numbers
78 * 41 * 1 * 83 * 31 * 60 * 58 * 98 * 58 * 33 * 26 * 75 * 45 * 24 * 64 * 97 * 87 * 59 * 60 * 77 * 88 *
array.length: 21
unsorted array
78,41,1,83,31,60,58,98,58,33,26,75,45,24,64,97,87,59,60,77,88
sorted array
1,24,26,31,33,41,45,58,58,59,60,60,64,75,77,78,83,87,88,97,98

bubble_sort.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Bubble Sort</title>
</head>
<body>

  <p>A basic bubble sort implementation</p>

  <script>
    var arr = [];
    var i;
    document.write("original array of random numbers <br>");
    for(i = 0; i <= 3;i++) {
            var x = (Math.random() * 100 + 1);
            //var x = (Math.floor(Math.random() * 100 + 1));
            arr[i] = x;
            document.write(arr[i] + " *  ");
        }
    document.write("<br>array.length: " + arr.length + "<br>");
    document.write("unsorted array <br>");
    document.write(arr + "<br>");
    var n = arr.length;
    var temp = 0;
    for(var i=0; i < n; i++){
        for(var j=1; j < (n-i); j++){
            if(arr[j-1] > arr[j]){
                //swap elements
                temp = arr[j-1];
                arr[j-1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    document.write("sorted array");
    document.write("<br>" + arr);
  </script>

</body>
</html>
```

U:--- **bubble_sort.html** All L15 (HTML+JS)
Wrote /Users/philip/Desktop/bubble_sort.html

file:///Users/philip/Desktop/bubble_s

A basic bubble sort implementation

original array of random numbers
32.25461985733381 * 1.1602934928296995 * 23.978567087294046 * 16.802437258240243 *
array.length: 4
unsorted array
32.25461985733381,1.1602934928296995,23.978567087294046,16.802437258240243
sorted array
1.1602934928296995,16.802437258240243,23.978567087294046,32.25461985733381

bubble_sort.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Bubble Sort</title>
</head>
<body>

  <p>A basic bubble sort implementation</p>

  <script>
    var arr = [];
    var i;
    document.write("original array of random numbers <br>");
    for(i = 0; i <= 500; i++) {
            //var x = (Math.random() * 100 + 1);
            var x = (Math.floor(Math.random() * 100 + 1));
            arr[i] = x;
            document.write(arr[i] + " *  ");
        }
    document.write("<br>array.length: " + arr.length + "<br>");
    document.write("unsorted array <br>");
    document.write(arr + "<br>");
    var n = arr.length;
    var temp = 0;
    for(var i=0; i < n; i++){
        for(var j=1; j < (n-i); j++){
            if(arr[j-1] > arr[j]){
                //swap elements
                temp = arr[j-1];
                arr[j-1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    document.write("sorted array");
    document.write("<br>" + arr);
  </script>

</body>
</html>
```

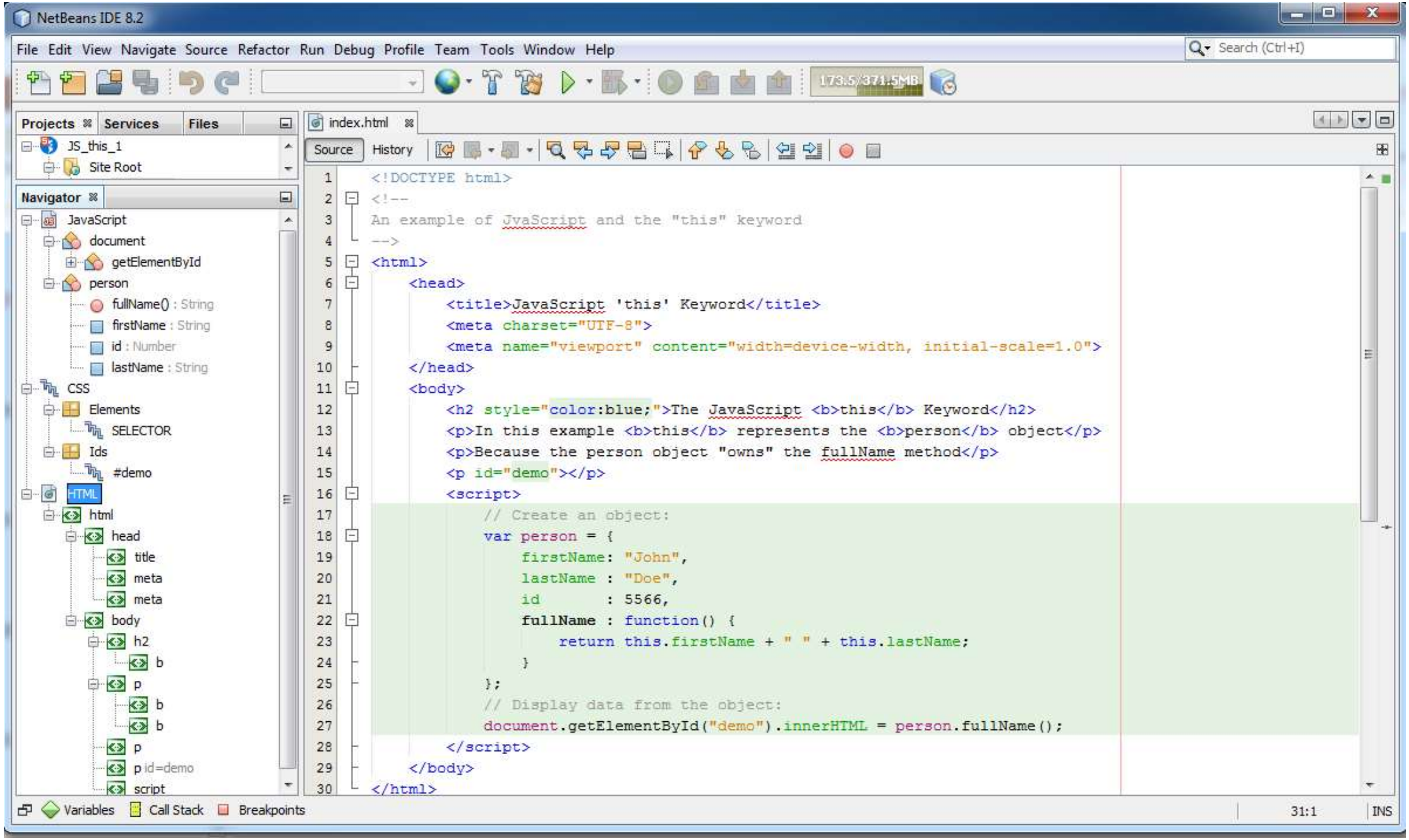
U:--- **bubble_sort.html** All L42 (HTML+)

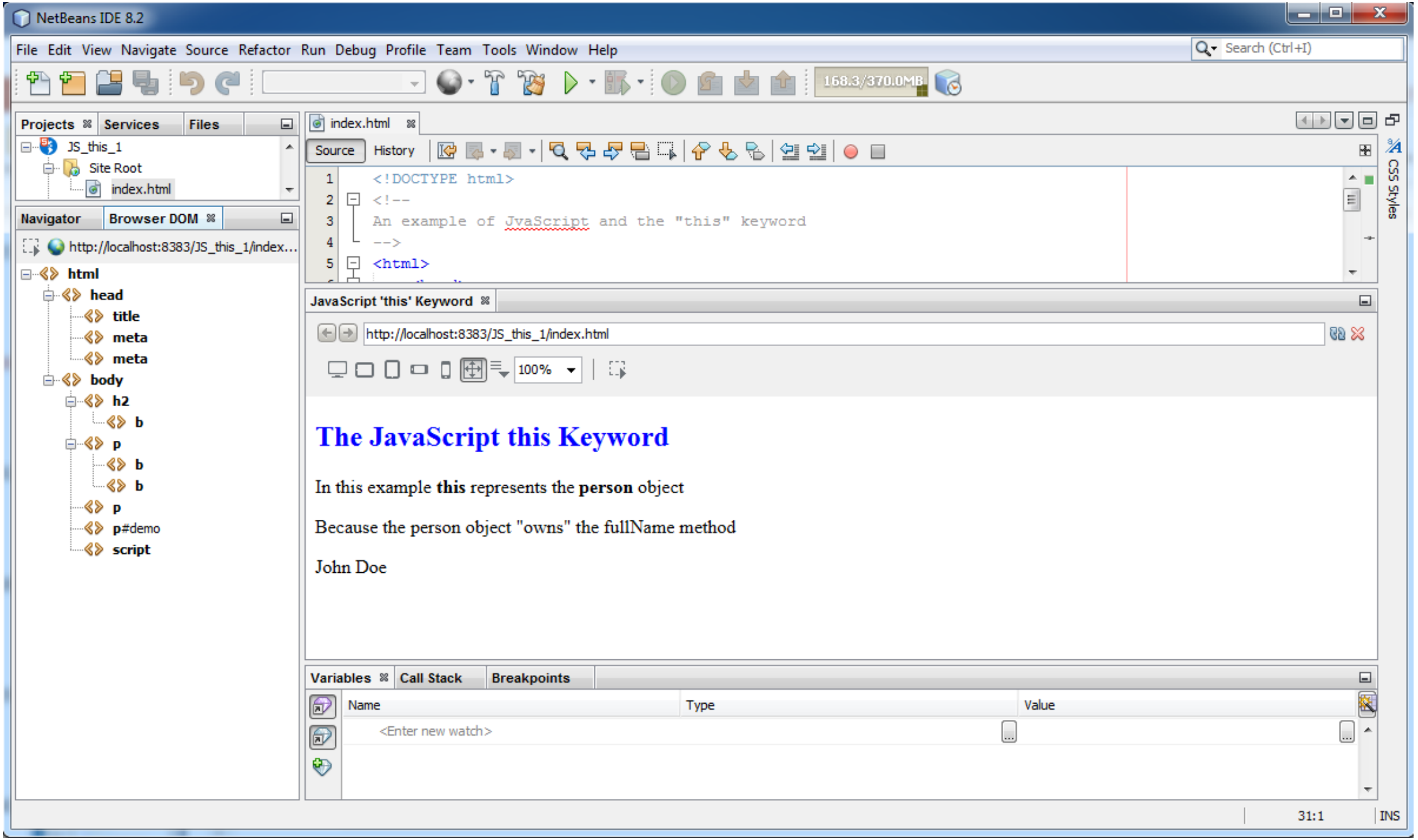
file:///Users/philip/Desktop

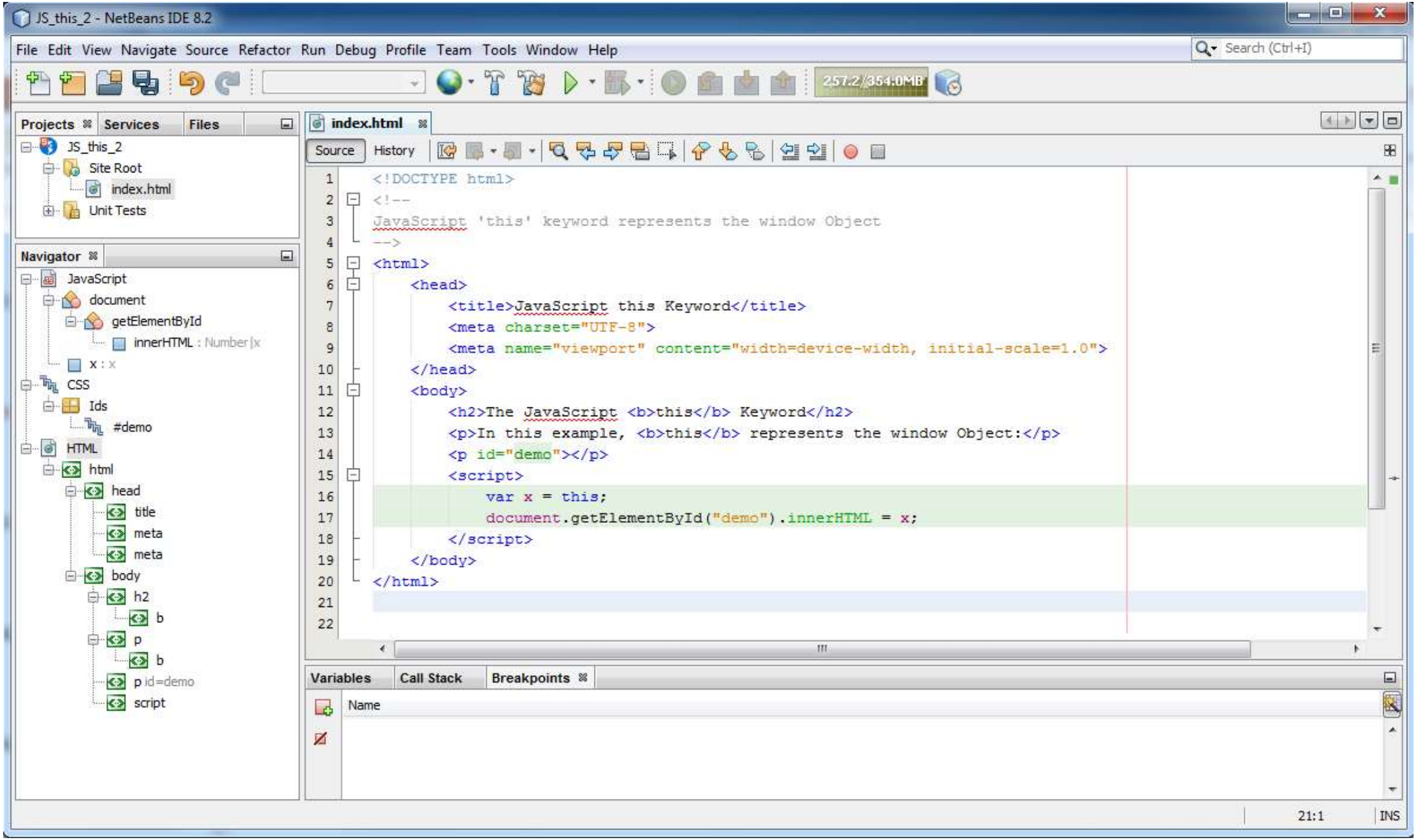
A basic bubble sort implementation

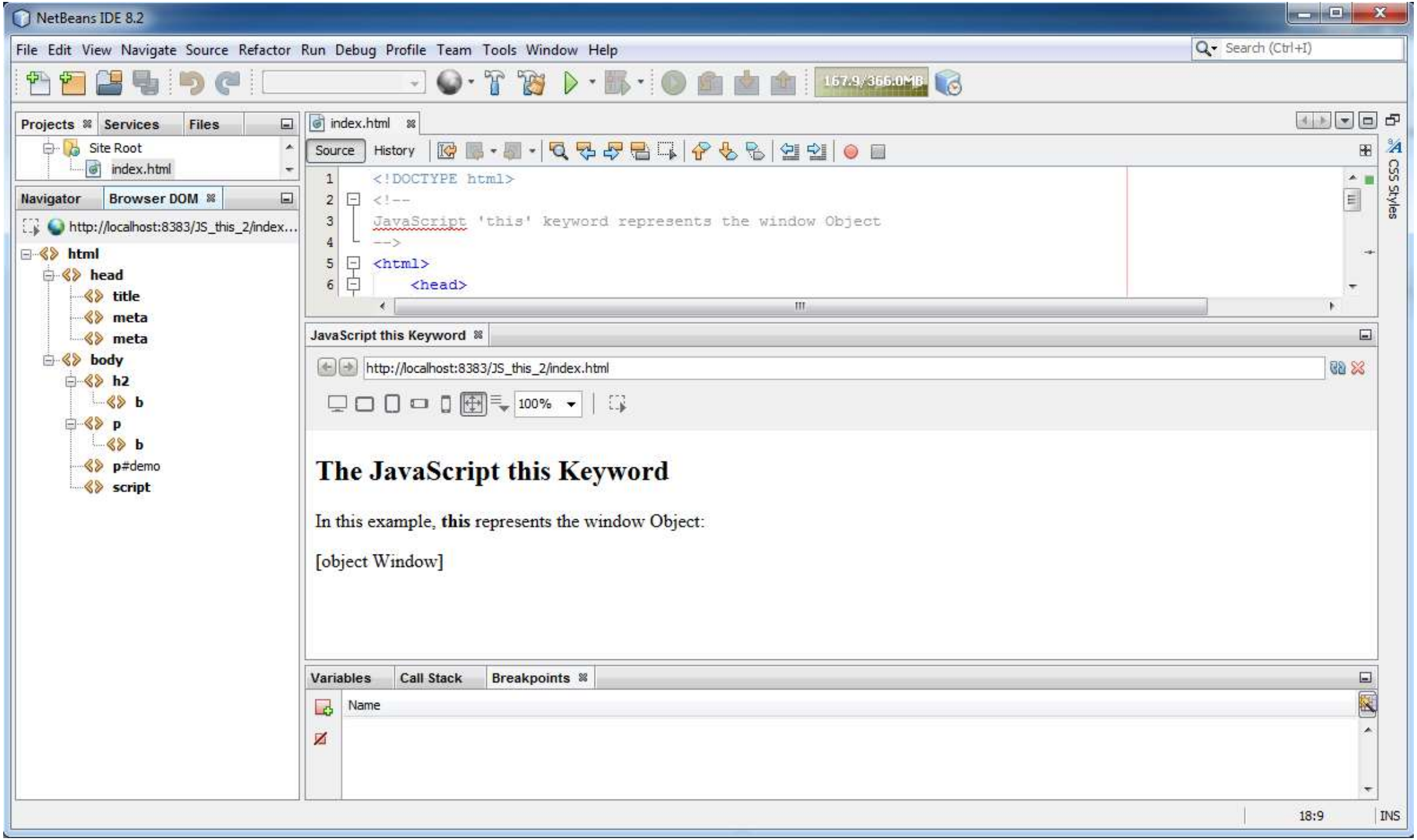
original array of random numbers
75 * 99 * 31 * 69 * 35 * 13 * 35 * 6 * 33 * 56 * 57 * 73 * 66 * 100 * 60 * 7 * 7 * 14 * 56 * 5 * 10 *
79 * 52 * 55 * 62 * 91 * 40 * 42 * 50 * 28 * 100 * 33 * 67 * 23 * 36 * 92 * 11 * 64 * 30 * 27 * 25 *
46 * 59 * 93 * 57 * 42 * 21 * 56 * 34 * 2 * 97 * 60 * 35 * 7 * 65 * 12 * 89 * 79 * 28 * 44 * 89 * 22 *
5 * 59 * 3 * 4 * 66 * 52 * 83 * 74 * 78 * 48 * 9 * 96 * 89 * 4 * 9 * 38 * 54 * 14 * 60 * 47 * 7 * 23 *
47 * 92 * 47 * 86 * 33 * 73 * 43 * 95 * 85 * 36 * 73 * 85 * 72 * 16 * 7 * 17 * 64 * 73 * 12 * 26 * 57 *
63 * 13 * 4 * 80 * 19 * 57 * 33 * 67 * 55 * 21 * 94 * 77 * 2 * 77 * 17 * 51 * 51 * 6 * 31 * 53 * 15 *
70 * 39 * 62 * 53 * 7 * 9 * 32 * 4 * 27 * 75 * 23 * 43 * 13 * 48 * 9 * 90 * 76 * 28 * 13 * 44 * 53 * 44
* 29 * 92 * 25 * 24 * 7 * 23 * 25 * 71 * 34 * 87 * 60 * 67 * 32 * 98 * 100 * 91 * 59 * 10 * 55 * 17 *
17 * 81 * 24 * 11 * 95 * 98 * 100 * 13 * 43 * 73 * 50 * 6 * 76 * 31 * 86 * 70 * 7 * 7 * 13 * 7 * 12 *
26 * 37 * 64 * 97 * 34 * 84 * 87 * 65 * 19 * 27 * 93 * 58 * 29 * 53 * 52 * 26 * 79 * 60 * 49 * 65 * 12
* 96 * 37 * 36 * 54 * 84 * 38 * 36 * 58 * 49 * 25 * 19 * 91 * 88 * 61 * 38 * 99 * 10 * 55 * 46 * 43 *
89 * 15 * 89 * 66 * 11 * 66 * 26 * 33 * 87 * 7 * 72 * 45 * 57 * 33 * 15 * 47 * 9 * 73 * 70 * 8 * 18 *
34 * 60 * 83 * 42 * 40 * 66 * 34 * 4 * 34 * 18 * 54 * 55 * 39 * 76 * 71 * 49 * 16 * 49 * 36 * 25 * 82 *
76 * 68 * 50 * 79 * 62 * 27 * 82 * 58 * 53 * 96 * 59 * 51 * 39 * 99 * 25 * 50 * 12 * 64 * 58 * 16 * 13
* 49 * 38 * 27 * 93 * 77 * 71 * 30 * 86 * 3 * 49 * 52 * 5 * 13 * 83 * 94 * 36 * 85 * 98 * 47 * 11 * 98
* 57 * 25 * 73 * 66 * 73 * 40 * 6 * 56 * 36 * 44 * 6 * 40 * 13 * 19 * 24 * 57 * 82 * 96 * 90 * 8 * 70 *
18 * 38 * 97 * 73 * 70 * 92 * 78 * 38 * 57 * 21 * 69 * 44 * 61 * 96 * 6 * 91 * 42 * 84 * 21 * 52 * 17 *
96 * 95 * 1 * 91 * 15 * 13 * 79 * 91 * 90 * 83 * 2 * 36 * 62 * 55 * 82 * 38 * 45 * 65 * 12 * 37 * 99 *
25 * 70 * 94 * 33 * 26 * 5 * 97 * 74 * 12 * 98 * 30 * 88 * 96 * 71 * 26 * 93 * 63 * 31 * 80 * 17 * 89 *
77 * 43 * 11 * 57 * 63 * 6 * 49 * 22 * 72 * 84 * 56 * 95 * 2 * 44 * 44 * 36 * 3 * 9 * 56 * 43 * 77 * 50
* 80 * 31 * 18 * 49 * 97 * 61 * 75 * 77 * 76 * 25 * 23 * 75 * 52 * 30 * 30 * 77 * 69 * 38 * 77 * 58 *
57 * 78 * 97 * 43 * 22 * 22 * 6 * 68 * 81 * 65 * 42 * 84 * 38 * 66 * 87 * 15 * 18 * 43 * 39 * 43 * 66 *
67 * 63 * 34 * 19 * 13 * 70 * 83 * 69 * 46 * 7 * 50 * 90 * 88 * 1 * 85 * 22 * 76 * 19 * 23 * 26 * 84 *
17 * 80 * 53 * 45 * 24 * 26 * 44 * 56 * 25 * 84 * 7 * 60 * 49 * 79 * 6 * 76 * 54 * 72 * 98 *
array.length: 501
unsorted array
75,99,31,69,35,13,35,6,33,56,57,73,66,100,60,7,7,14,56,5,10,79,52,55,62,91,40,42,50,28,100,33,67,23,3
sorted array
1,1,2,2,2,2,3,3,3,4,4,4,4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,8,8,9,9,9,9,9,9,10,10,10,11,11

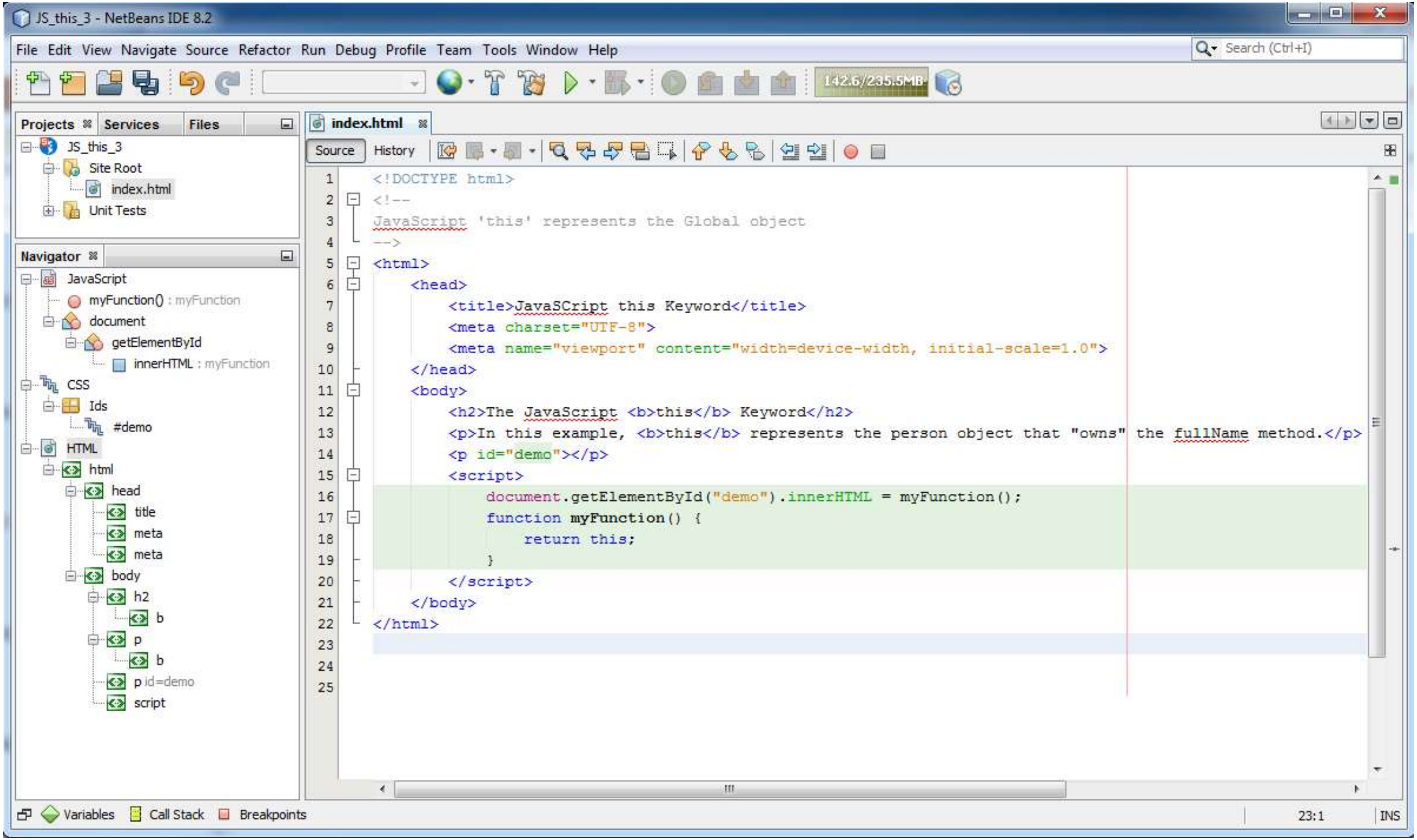
The `this` Keyword

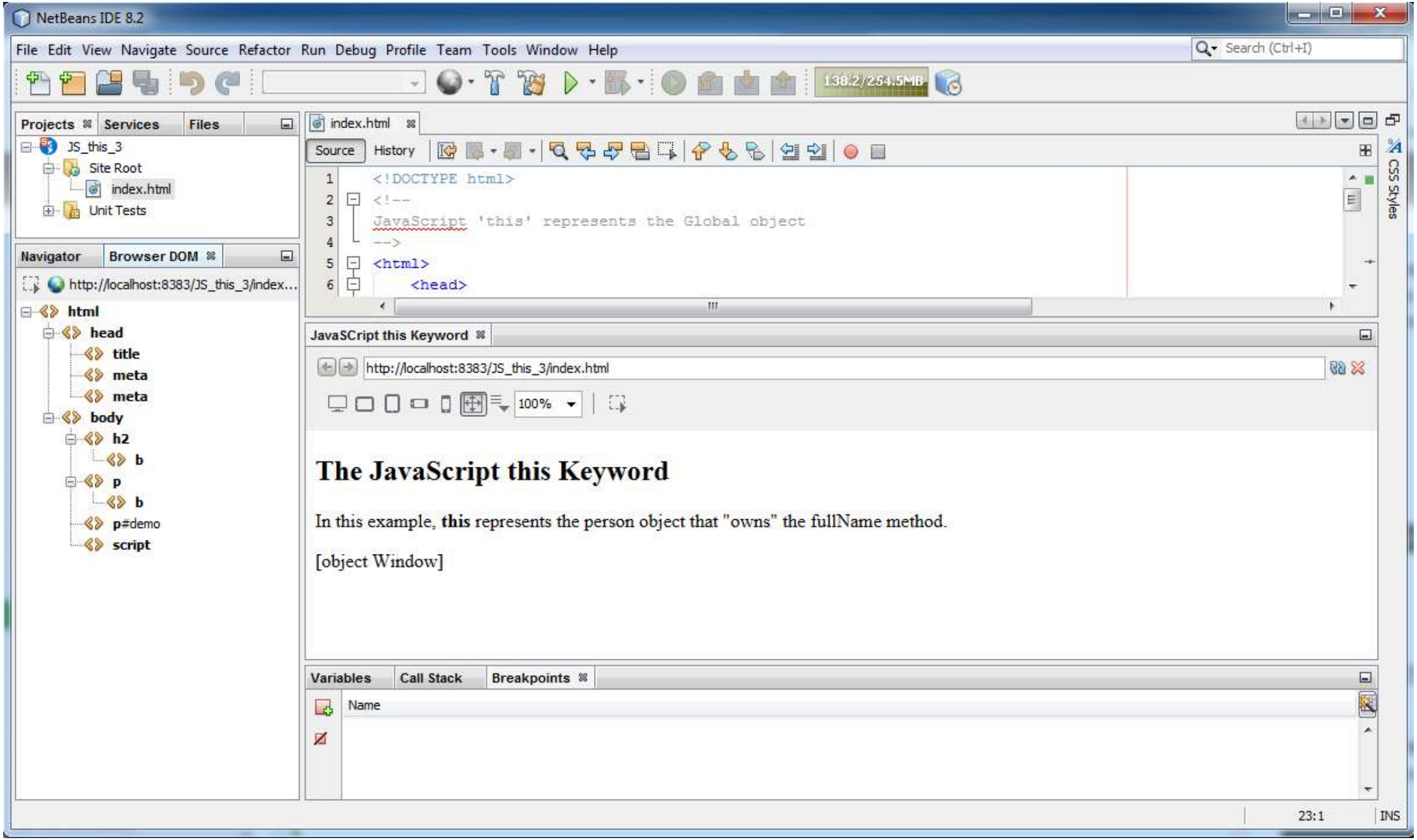

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Services
Files
JS_this_1
Site Root
Navigator
JavaScript
document
getElementById
person
fullName() : String
firstName : String
id : Number
lastName : String
CSS
Elements
SELECTOR
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
b
p
b
b
p
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 An example of JvaScript and the "this" keyword
4 -->
5 <html>
6 <head>
7 <title>JavaScript 'this' Keyword</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <h2 style="color:blue;">The JavaScript this Keyword</h2>
13 In this example this represents the person object
14 Because the person object "owns" the fullName method
15
16 <script>
17 // Create an object:
18 var person = {
19 firstName: "John",
20 lastName : "Doe",
21 id : 5566,
22 fullName : function() {
23 return this.firstName + " " + this.lastName;
24 }
25 };
26 // Display data from the object:
27 document.getElementById("demo").innerHTML = person.fullName();
28 </script>
29 </body>
30 </html>
Variables
Call Stack
Breakpoints
31:1
INS

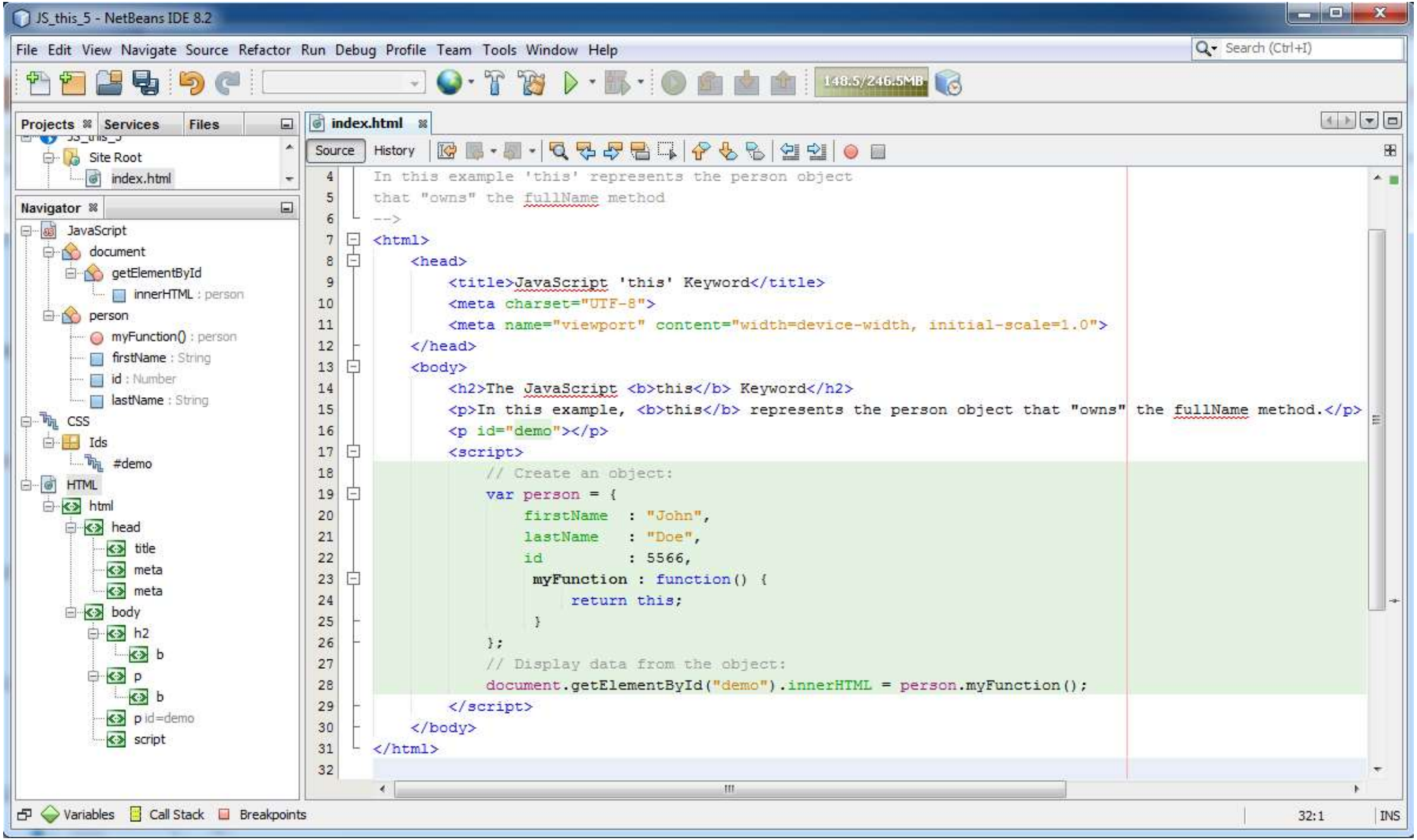

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
168.3/370.0MB
Projects
Services
Files
JS_this_1
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_this_1/index...
html
head
title
meta
meta
body
h2
b
p
b
b
p
p#demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 An example of JvaScript and the "this" keyword
4 -->
5 <html>
CSS Styles
JavaScript 'this' Keyword
http://localhost:8383/JS_this_1/index.html
100%
The JavaScript this Keyword
In this example this represents the person object
Because the person object "owns" the fullName method
John Doe
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
31:1
INS

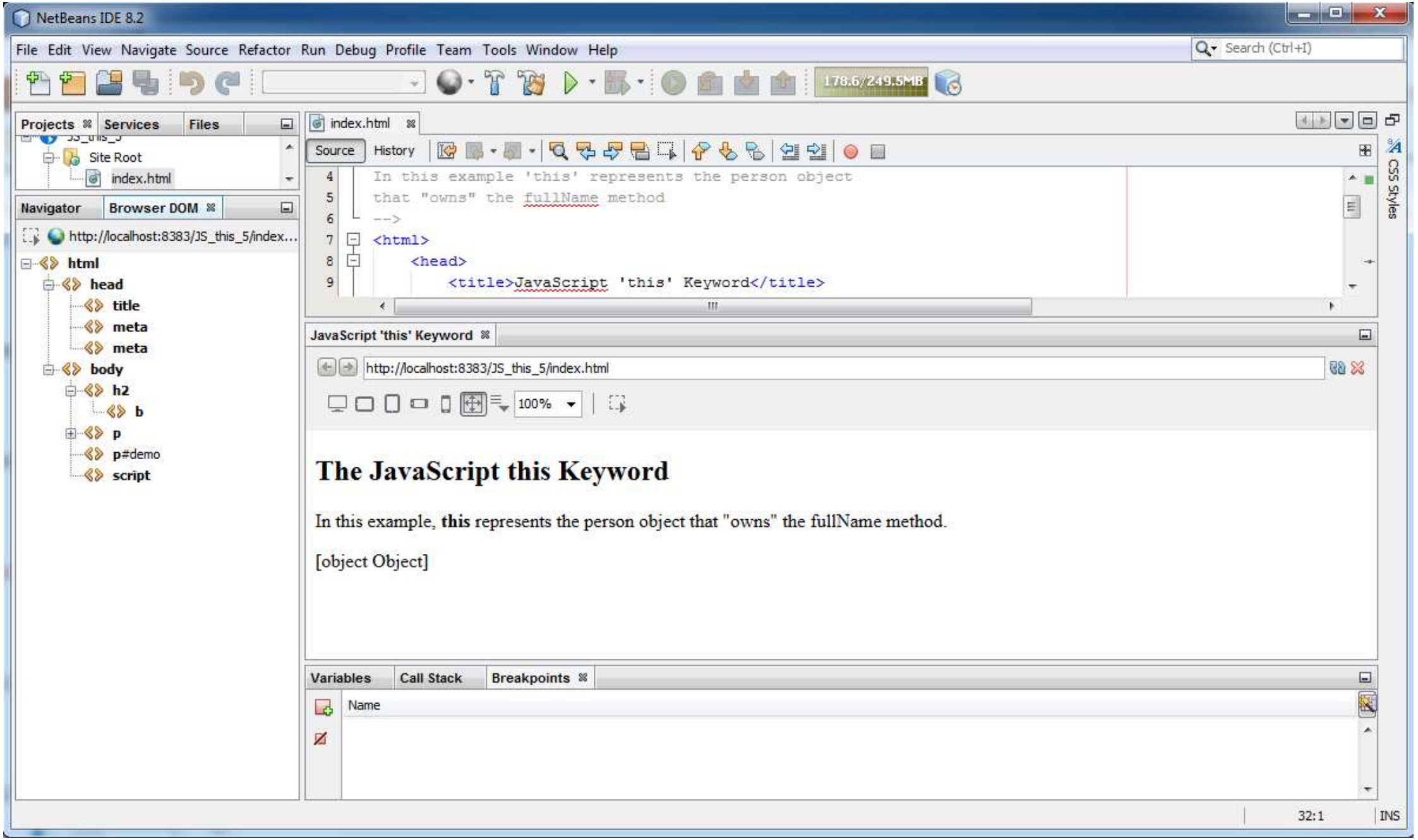

JS_this_2 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
257.2/354.0MB
Projects
Services
Files
JS_this_2
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Number|x
x : x
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
b
p
b
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript 'this' keyword represents the window Object
4 -->
5 <html>
6 <head>
7 <title>JavaScript this Keyword</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <h2>The JavaScript this Keyword</h2>
13 In this example, this represents the window Object:
14
15 <script>
16 var x = this;
17 document.getElementById("demo").innerHTML = x;
18 </script>
19 </body>
20 </html>
21
22
Variables
Call Stack
Breakpoints
Name
21:1
INS

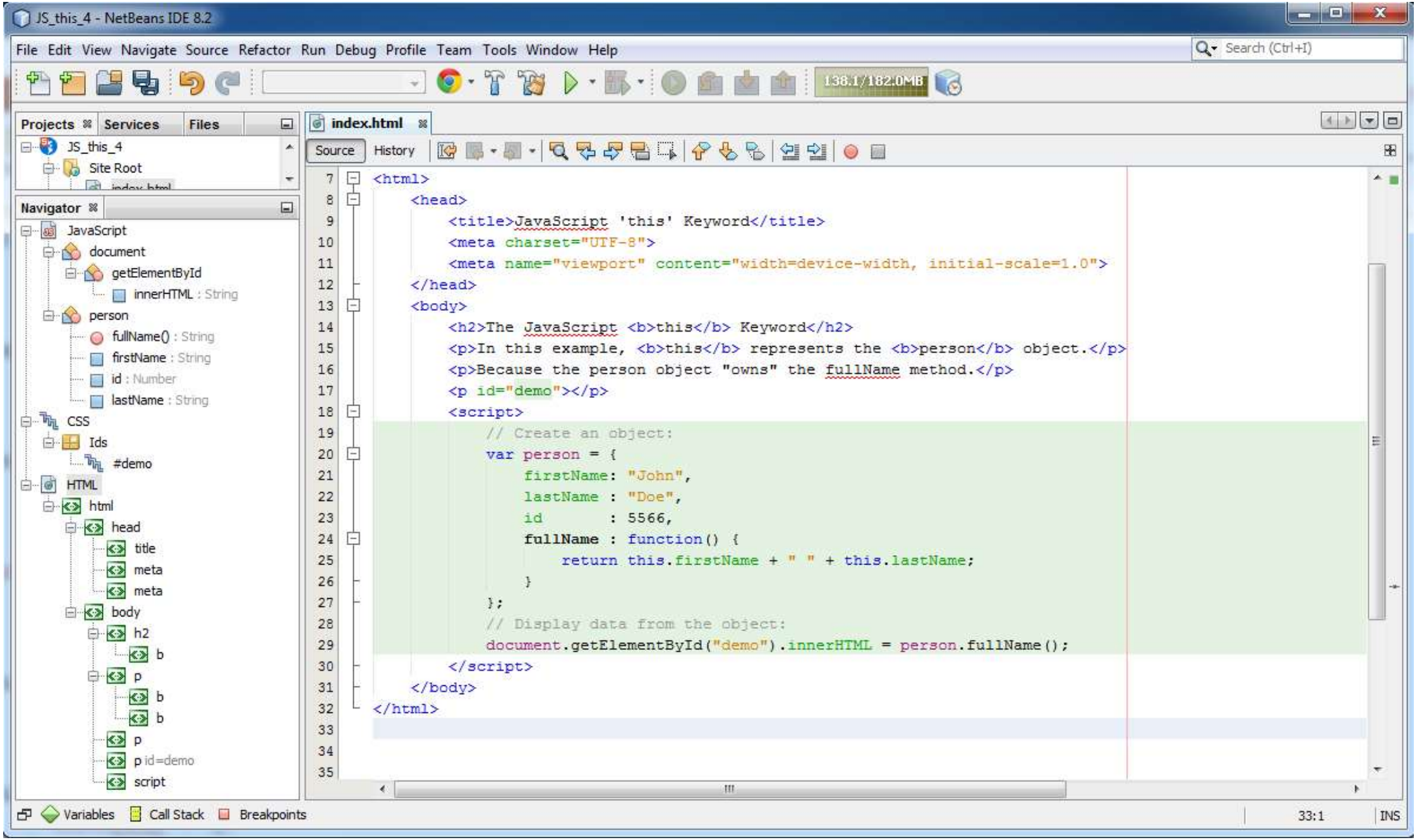

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
167.9/366.0MB
Projects
Services
Files
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_this_2/index...
html
head
title
meta
meta
body
h2
b
p
b
p#demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript 'this' keyword represents the window Object
4 -->
5 <html>
6 <head>
CSS Styles
JavaScript this Keyword
http://localhost:8383/JS_this_2/index.html
100%
The JavaScript this Keyword
In this example, this represents the window Object:
[object Window]
Variables
Call Stack
Breakpoints
Name
18:9
INS

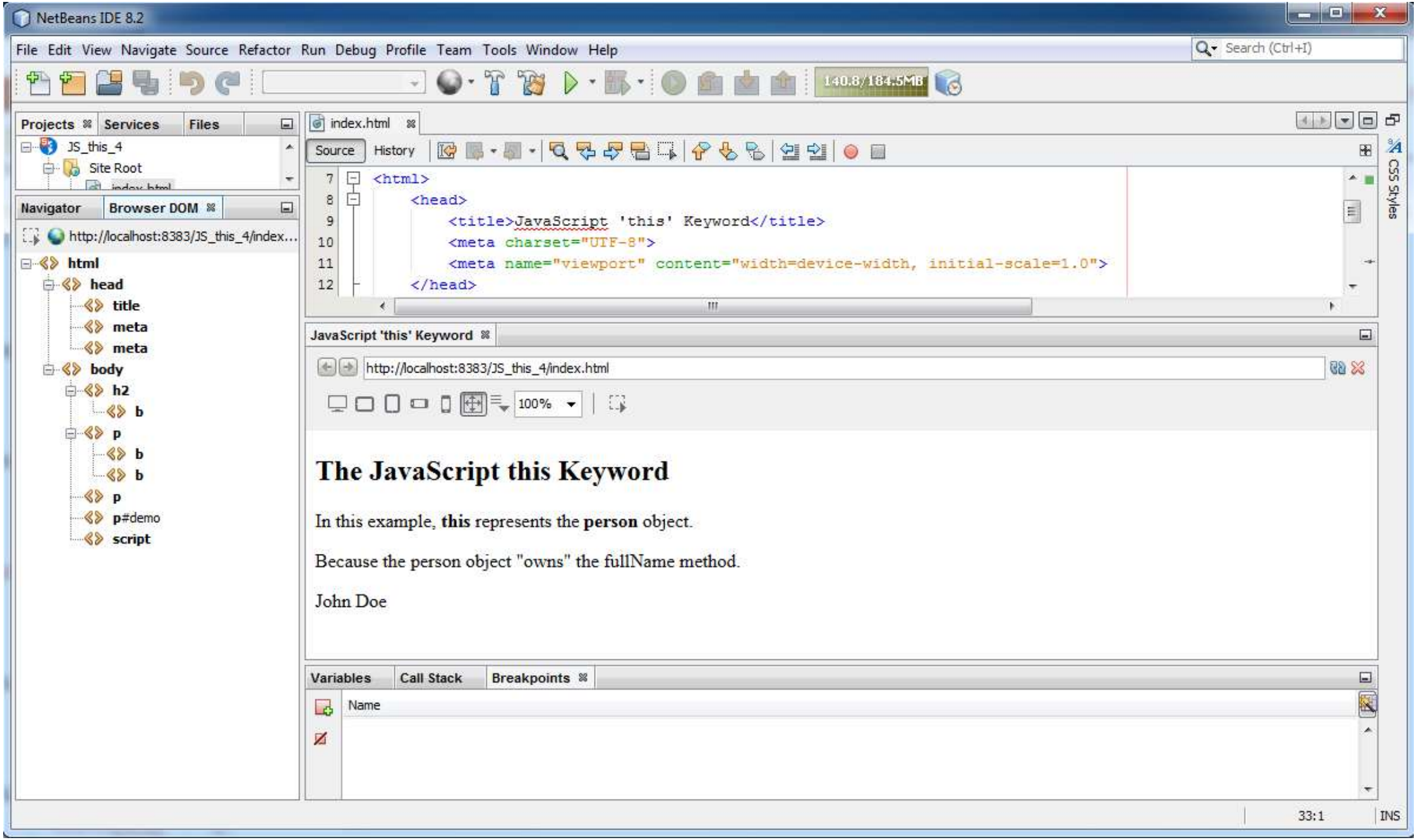

JS_this_3 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
142.6/235.5MB
Projects
Services
Files
JS_this_3
Site Root
index.html
Unit Tests
Navigator
JavaScript
myFunction() : myFunction
document
getElementById
innerHTML : myFunction
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
b
p
b
p id=demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript 'this' represents the Global object
4 -->
5 <html>
6 <head>
7 <title>JavaScript this Keyword</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <h2>The JavaScript this Keyword</h2>
13 In this example, this represents the person object that "owns" the fullName method.
14
15 <script>
16 document.getElementById("demo").innerHTML = myFunction();
17 function myFunction() {
18 return this;
19 }
20 </script>
21 </body>
22 </html>
23
24
25
Variables
Call Stack
Breakpoints
23:1
INS

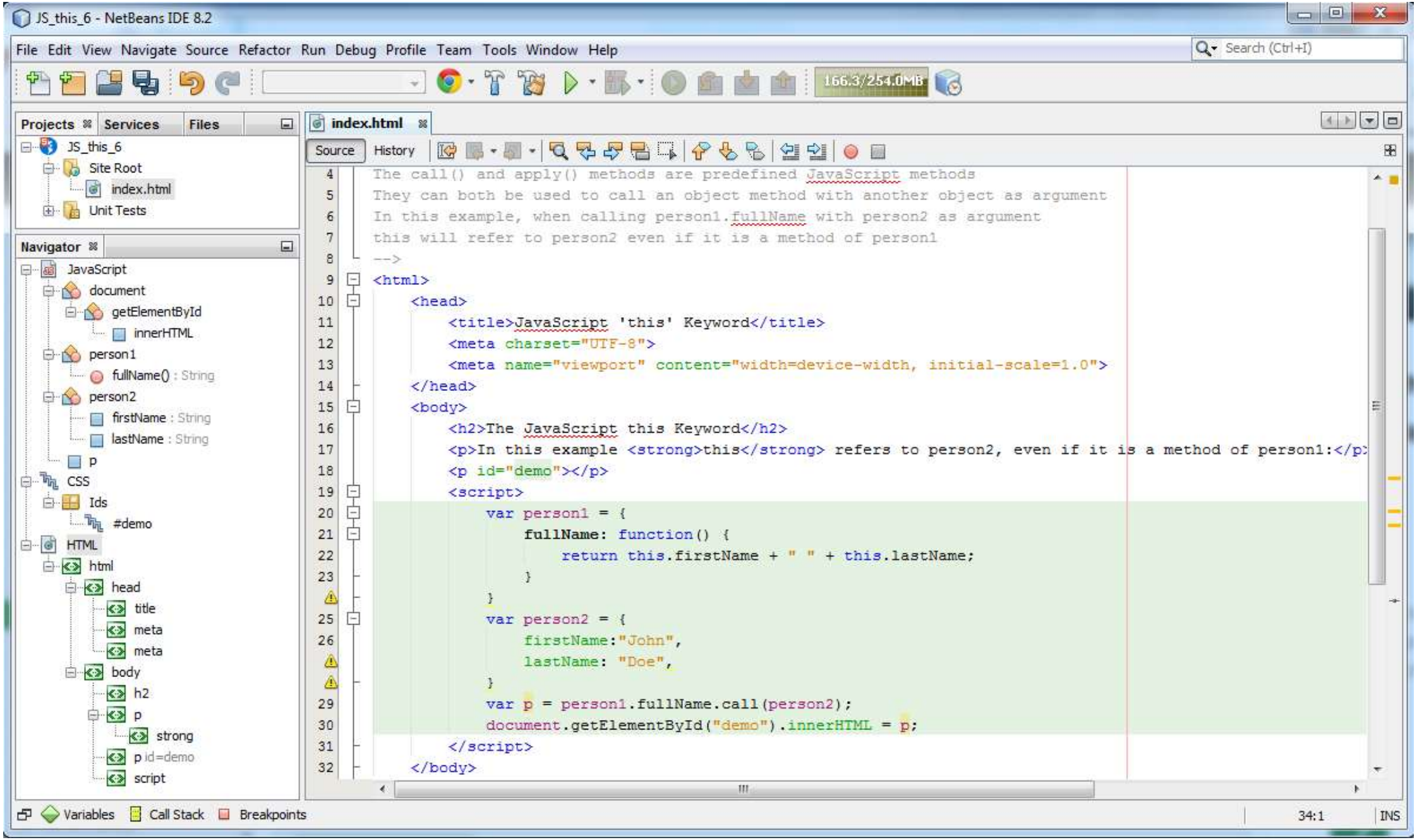

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
138.2/254.5MB
Projects
Services
Files
JS_this_3
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_this_3/index...
html
head
title
meta
meta
body
h2
b
p
b
p#demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript 'this' represents the Global object
4 -->
5 <html>
6 <head>
CSS Styles
JavaSCript this Keyword
http://localhost:8383/JS_this_3/index.html
100%
The JavaScript this Keyword
In this example, this represents the person object that "owns" the fullName method.
[object Window]
Variables
Call Stack
Breakpoints
Name
23:1
INS

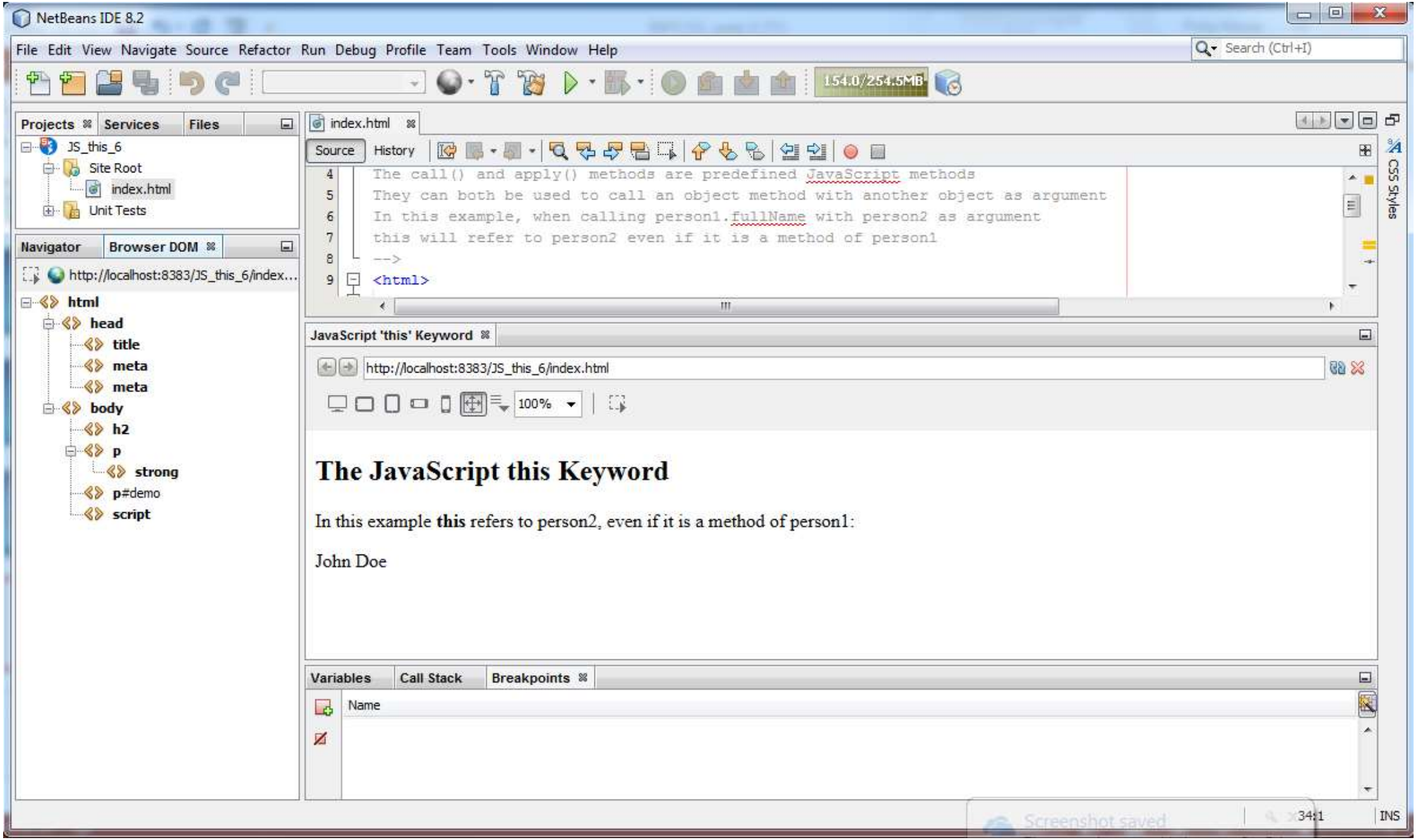

JS_this_5 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
148.5/246.5MB
Projects
Services
Files
Site Root
index.html
Navigator
JavaScript
document
getElementById
innerHTML : person
person
myFunction() : person
firstName : String
id : Number
lastName : String
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
b
p
b
p id=demo
script
index.html
Source
History
4 In this example 'this' represents the person object
5 that "owns" the fullName method
6 -->
7 <html>
8 <head>
9 <title>JavaScript 'this' Keyword</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>The JavaScript this Keyword</h2>
15 In this example, this represents the person object that "owns" the fullName method.
16
17 <script>
18 // Create an object:
19 var person = {
20 firstName : "John",
21 lastName : "Doe",
22 id : 5566,
23 myFunction : function() {
24 return this;
25 }
26 };
27 // Display data from the object:
28 document.getElementById("demo").innerHTML = person.myFunction();
29 </script>
30 </body>
31 </html>
32
Variables
Call Stack
Breakpoints
32:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
178.6/249.5MB
Projects
Services
Files
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_this_5/index...
html
head
title
meta
meta
body
h2
b
p
p#demo
script
index.html
Source
History
4 In this example 'this' represents the person object
5 that "owns" the fullName method
6 -->
7 <html>
8 <head>
9 <title>JavaScript 'this' Keyword</title>
CSS Styles
JavaScript 'this' Keyword
http://localhost:8383/JS_this_5/index.html
100%
The JavaScript this Keyword
In this example, this represents the person object that "owns" the fullName method.
[object Object]
Variables
Call Stack
Breakpoints
Name
32:1
INS


JS_this_4 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
138.1/182.0MB
Projects
Services
Files
JS_this_4
Site Root
Navigator
JavaScript
document
getElementById
innerHTML : String
person
fullName() : String
firstName : String
id : Number
lastName : String
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
b
p
b
b
p
p id=demo
script
index.html
Source
History
7 <html>
8 <head>
9 <title>JavaScript 'this' Keyword</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>The JavaScript this Keyword</h2>
15 In this example, this represents the person object.
16 Because the person object "owns" the fullName method.
17
18 <script>
19 // Create an object:
20 var person = {
21 firstName: "John",
22 lastName : "Doe",
23 id : 5566,
24 fullName : function() {
25 return this.firstName + " " + this.lastName;
26 }
27 };
28 // Display data from the object:
29 document.getElementById("demo").innerHTML = person.fullName();
30 </script>
31 </body>
32 </html>
33
34
35
Variables
Call Stack
Breakpoints
33:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
140.8/184.5MB
Projects
Services
Files
JS_this_4
Site Root
Navigator
Browser DOM
http://localhost:8383/JS_this_4/index...
html
head
title
meta
meta
body
h2
b
p
b
b
p
p#demo
script
index.html
Source
History
7 <html>
8 <head>
9 <title>JavaScript 'this' Keyword</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
CSS Styles
JavaScript 'this' Keyword
http://localhost:8383/JS_this_4/index.html
100%
The JavaScript this Keyword
In this example, this represents the person object.
Because the person object "owns" the fullName method.
John Doe
Variables
Call Stack
Breakpoints
Name
33:1
INS


JS_this_6 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
166.3/254.0MB
Projects
Services
Files
JS_this_6
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML
person1
fullName() : String
person2
firstName : String
lastName : String
p
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
strong
p id=demo
script
index.html
Source
History
4 The call() and apply() methods are predefined JavaScript methods
5 They can both be used to call an object method with another object as argument
6 In this example, when calling person1.fullName with person2 as argument
7 this will refer to person2 even if it is a method of person1
8 -->
9 <html>
10 <head>
11 <title>JavaScript 'this' Keyword</title>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 </head>
15 <body>
16 <h2>The JavaScript this Keyword</h2>
17 In this example this refers to person2, even if it is a method of person1:
18
19 <script>
20 var person1 = {
21 fullName: function() {
22 return this.firstName + " " + this.lastName;
23 }
}
25 var person2 = {
26 firstName:"John",
lastName: "Doe",
}
29 var p = person1.fullName.call(person2);
30 document.getElementById("demo").innerHTML = p;
31 </script>
32 </body>
Variables
Call Stack
Breakpoints
34:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
154.0/254.5MB
Projects
Services
Files
JS_this_6
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_this_6/index...
html
head
title
meta
meta
body
h2
p
strong
p#demo
script
index.html
Source
History
4 The call() and apply() methods are predefined JavaScript methods
5 They can both be used to call an object method with another object as argument
6 In this example, when calling person1.fullName with person2 as argument
7 this will refer to person2 even if it is a method of person1
8 -->
9 <html>
CSS Styles
JavaScript 'this' Keyword
http://localhost:8383/JS_this_6/index.html
100%
The JavaScript this Keyword
In this example this refers to person2, even if it is a method of person1:
John Doe
Variables
Call Stack
Breakpoints
Name
Screenshot saved
34:1
INS

Math Methods

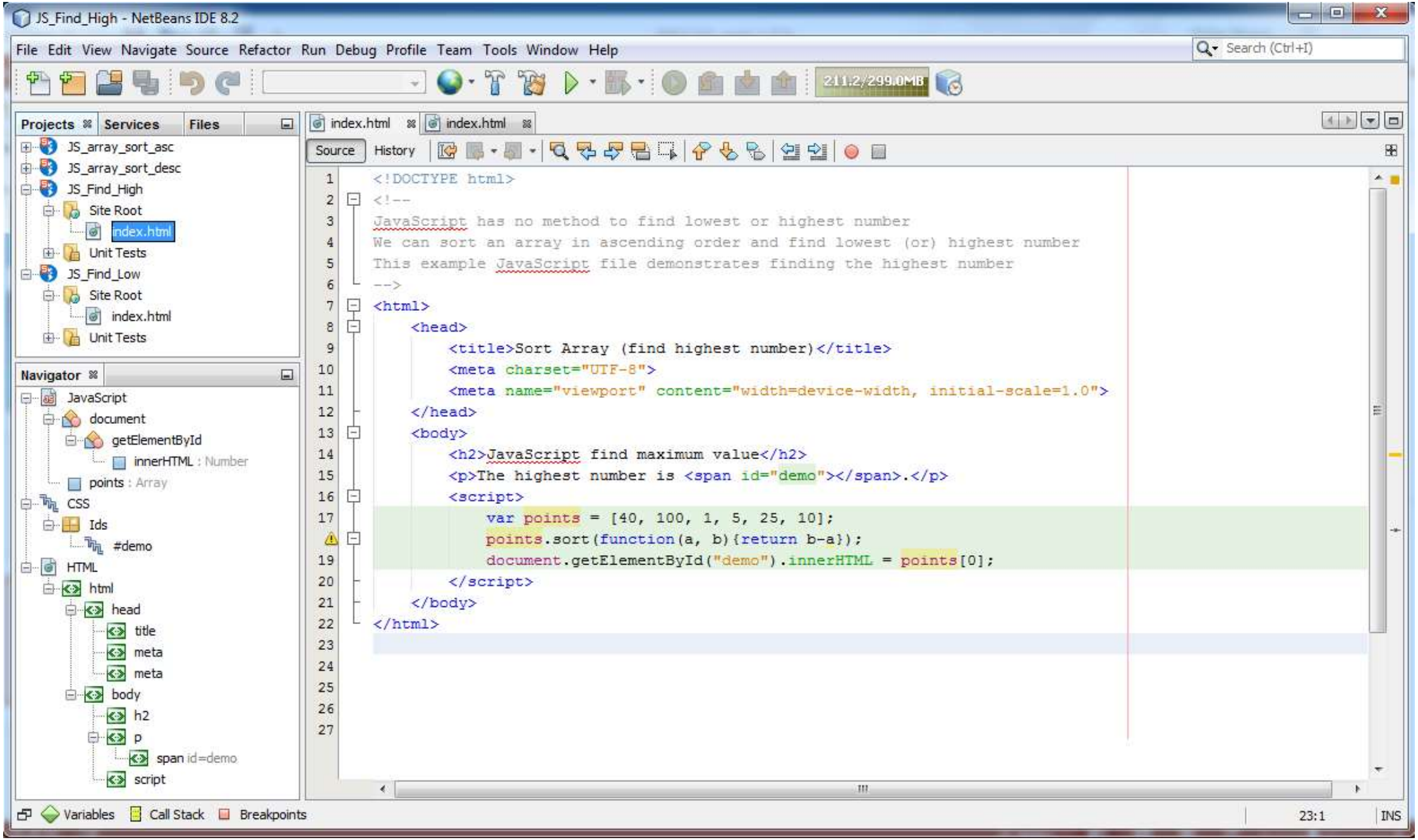
Math Methods

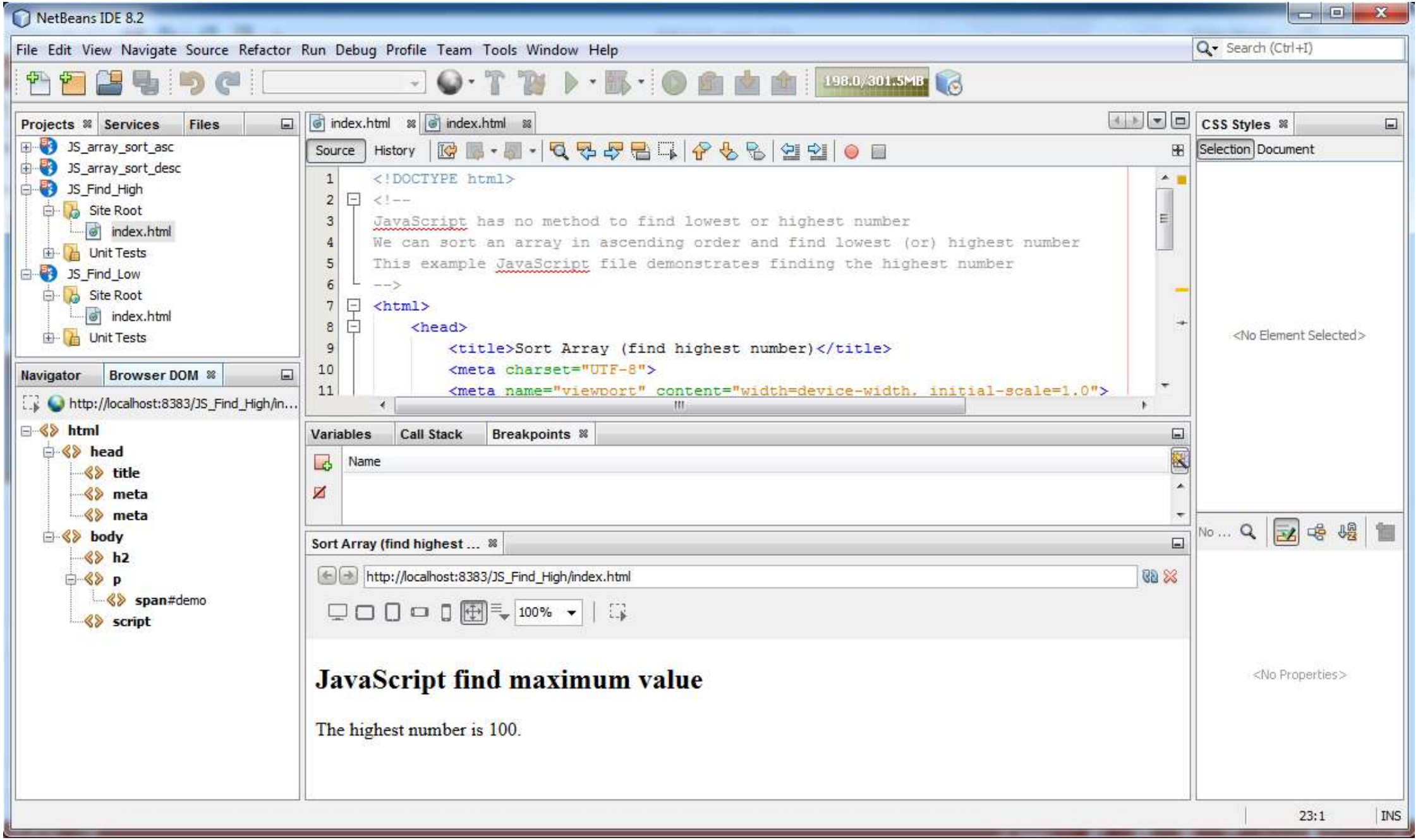
- There are many math methods for implementing generally used mathematic calculations
- The following slides demonstrate some frequently used math methods
  - Math.max() and Math.min() and Math.max.apply
  - Math.round()
  - Math.PI (shows the use of a GLOBAL variable as a constant)
- Comprehensive details of the math methods may be found at
  - https://www.w3schools.com/js/js_arrays.asp
  - https://www.w3schools.com/js/js_math.asp

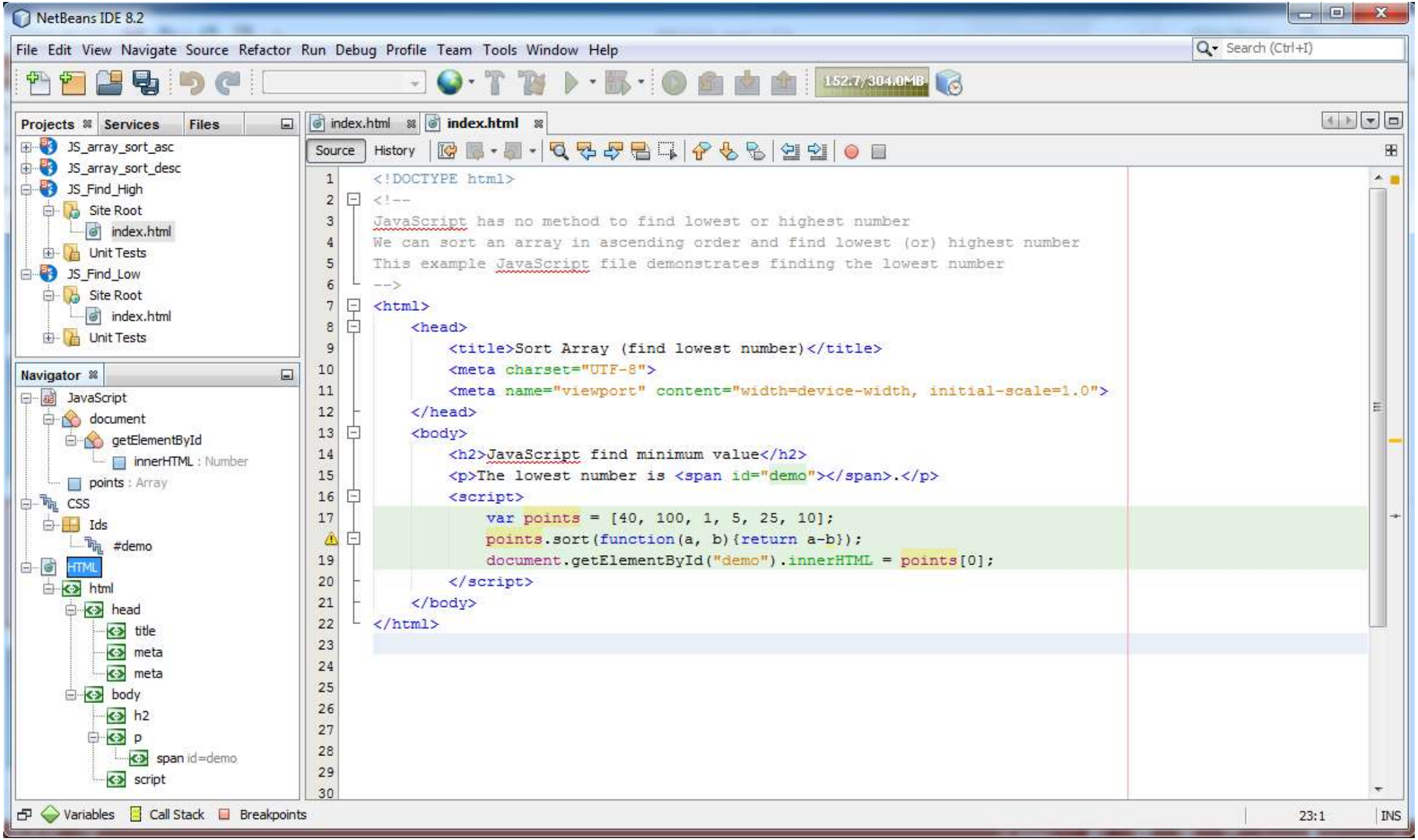
Max and Min Methods

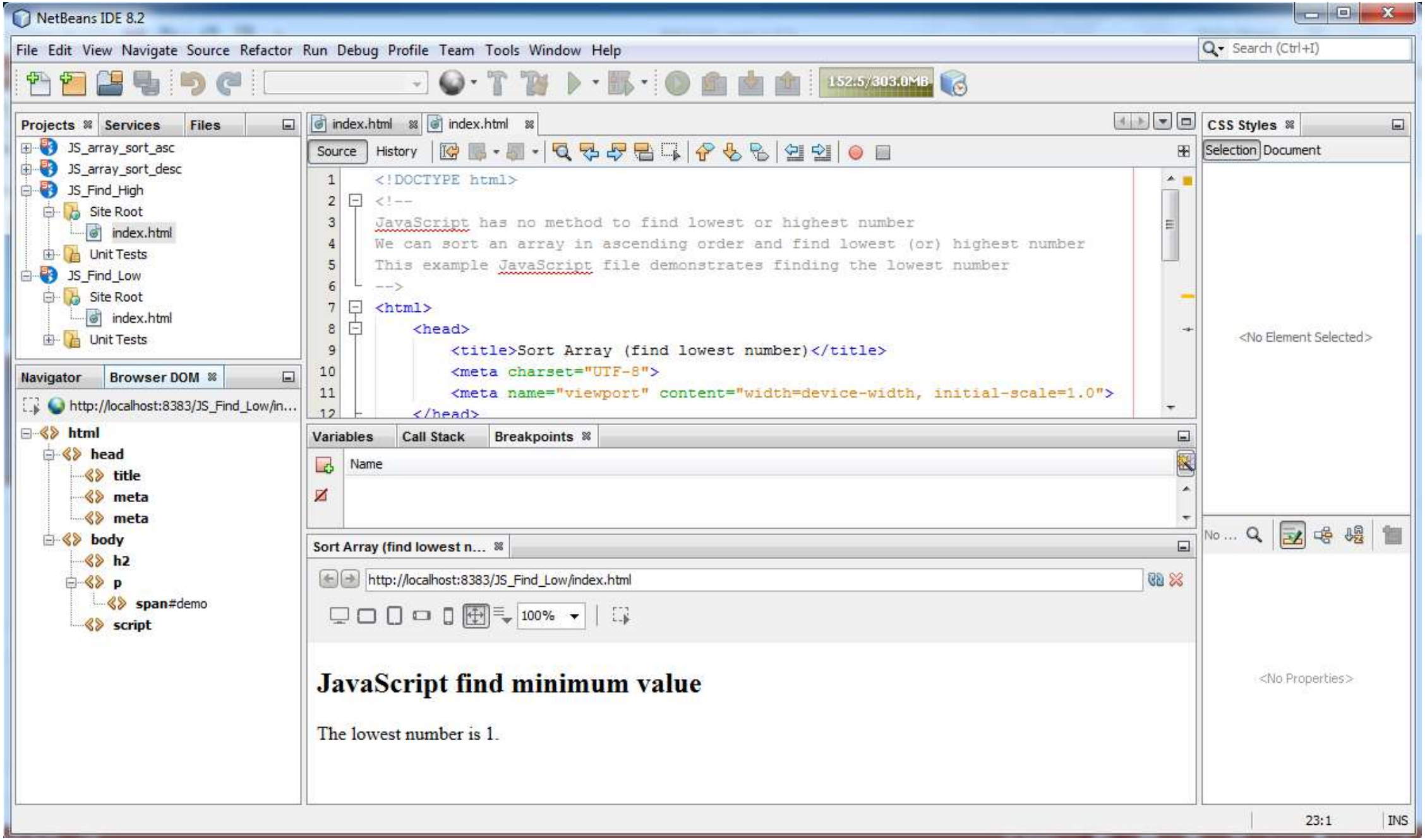
Find the Max or Min value in an array

- In JavaScript following an array sort we can use the index to obtain the highest and lowest values
  - The key line of code can be seen in line 18 of both ascending and descending order JavaScript files
  - Find max value: `points.sort(function(a, b){return a-b});`
  - Find min value: `points.sort(function(a, b){return b-a});`
  - The JavaScript code is the same except for the return
- The following slides show worked examples of the process


JS_Find_High - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
211.2/299.0MB
Projects
Services
Files
JS_array_sort_asc
JS_array_sort_desc
JS_Find_High
Site Root
index.html
Unit Tests
JS_Find_Low
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Number
points : Array
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
span id=demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript has no method to find lowest or highest number
4 We can sort an array in ascending order and find lowest (or) highest number
5 This example JavaScript file demonstrates finding the highest number
6 -->
7 <html>
8 <head>
9 <title>Sort Array (find highest number)</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>JavaScript find maximum value</h2>
15 The highest number is .
16 <script>
17 var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b-a});
19 document.getElementById("demo").innerHTML = points[0];
20 </script>
21 </body>
22 </html>
23
24
25
26
27
Variables
Call Stack
Breakpoints
23:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
198.0/301.5MB
Projects
Services
Files
JS_array_sort_asc
JS_array_sort_desc
JS_Find_High
Site Root
index.html
Unit Tests
JS_Find_Low
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Find_High/in...
html
head
title
meta
meta
body
h2
p
span#demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript has no method to find lowest or highest number
4 We can sort an array in ascending order and find lowest (or) highest number
5 This example JavaScript file demonstrates finding the highest number
6 -->
7 <html>
8 <head>
9 <title>Sort Array (find highest number)</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
Variables
Call Stack
Breakpoints
Name
Sort Array (find highest ...
http://localhost:8383/JS_Find_High/index.html
100%
JavaScript find maximum value
The highest number is 100.
CSS Styles
Selection
Document
<No Element Selected>
<No Properties>
23:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
152.7/304.0MB
Projects
Services
Files
JS_array_sort_asc
JS_array_sort_desc
JS_Find_High
Site Root
index.html
Unit Tests
JS_Find_Low
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Number
points : Array
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
span id=demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript has no method to find lowest or highest number
4 We can sort an array in ascending order and find lowest (or) highest number
5 This example JavaScript file demonstrates finding the lowest number
6 -->
7 <html>
8 <head>
9 <title>Sort Array (find lowest number)</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>JavaScript find minimum value</h2>
15 The lowest number is .
16 <script>
17 var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return a-b});
19 document.getElementById("demo").innerHTML = points[0];
20 </script>
21 </body>
22 </html>
23
24
25
26
27
28
29
30
Variables
Call Stack
Breakpoints
23:1
INS

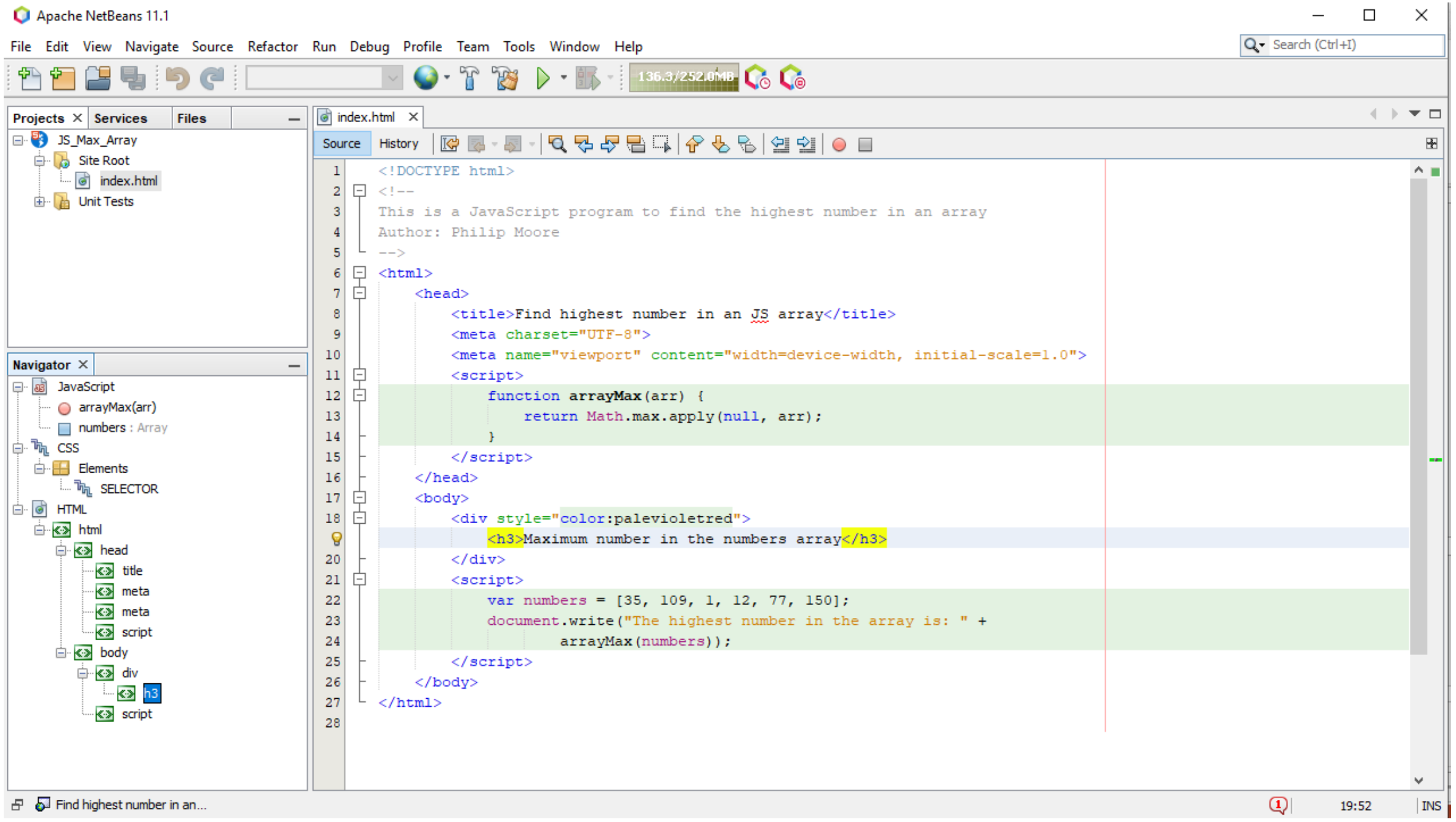

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
152.5/303.0MB
Projects
Services
Files
JS_array_sort_asc
JS_array_sort_desc
JS_Find_High
Site Root
index.html
Unit Tests
JS_Find_Low
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Find_Low/in...
html
head
title
meta
meta
body
h2
p
span#demo
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 JavaScript has no method to find lowest or highest number
4 We can sort an array in ascending order and find lowest (or) highest number
5 This example JavaScript file demonstrates finding the lowest number
6 -->
7 <html>
8 <head>
9 <title>Sort Array (find lowest number)</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
CSS Styles
Selection
Document
<No Element Selected>
Variables
Call Stack
Breakpoints
Name
Sort Array (find lowest n...
http://localhost:8383/JS_Find_Low/index.html
100%
JavaScript find minimum value
The lowest number is 1.
<No Properties>
23:1
INS

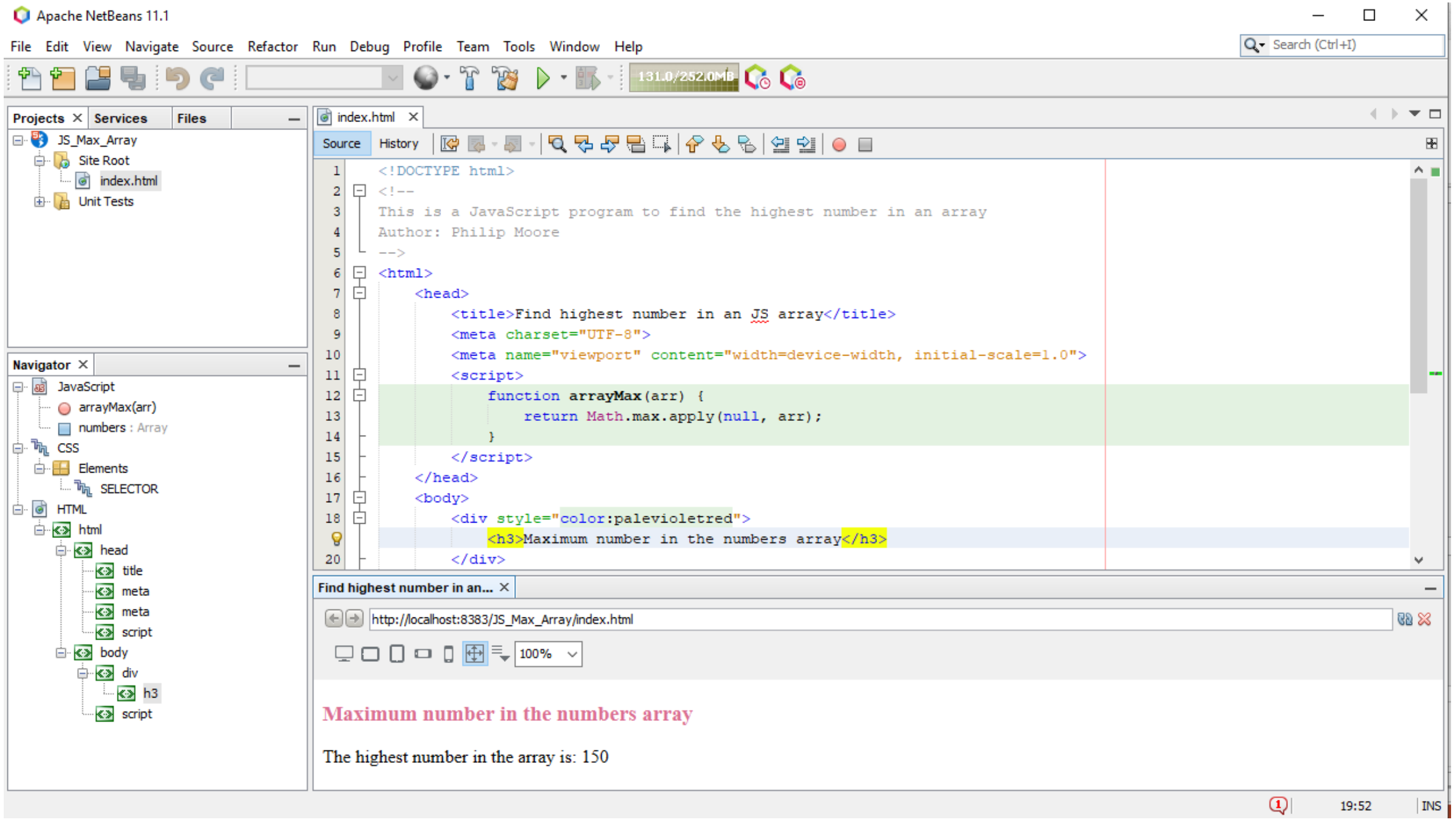
A Max Function for Arrays

- JavaScript math methods are often used on JavaScript arrays
- For example
  - **`Math.max.apply`** may be used to find the highest number in an array:

```
function myArrayMax(arr) {
    return Math.max.apply(null, arr);
}
```

- Math.max.apply([1, 2, 3]) is equivalent to Math.max(1, 2, 3)
- The following worked example shows the method and the result

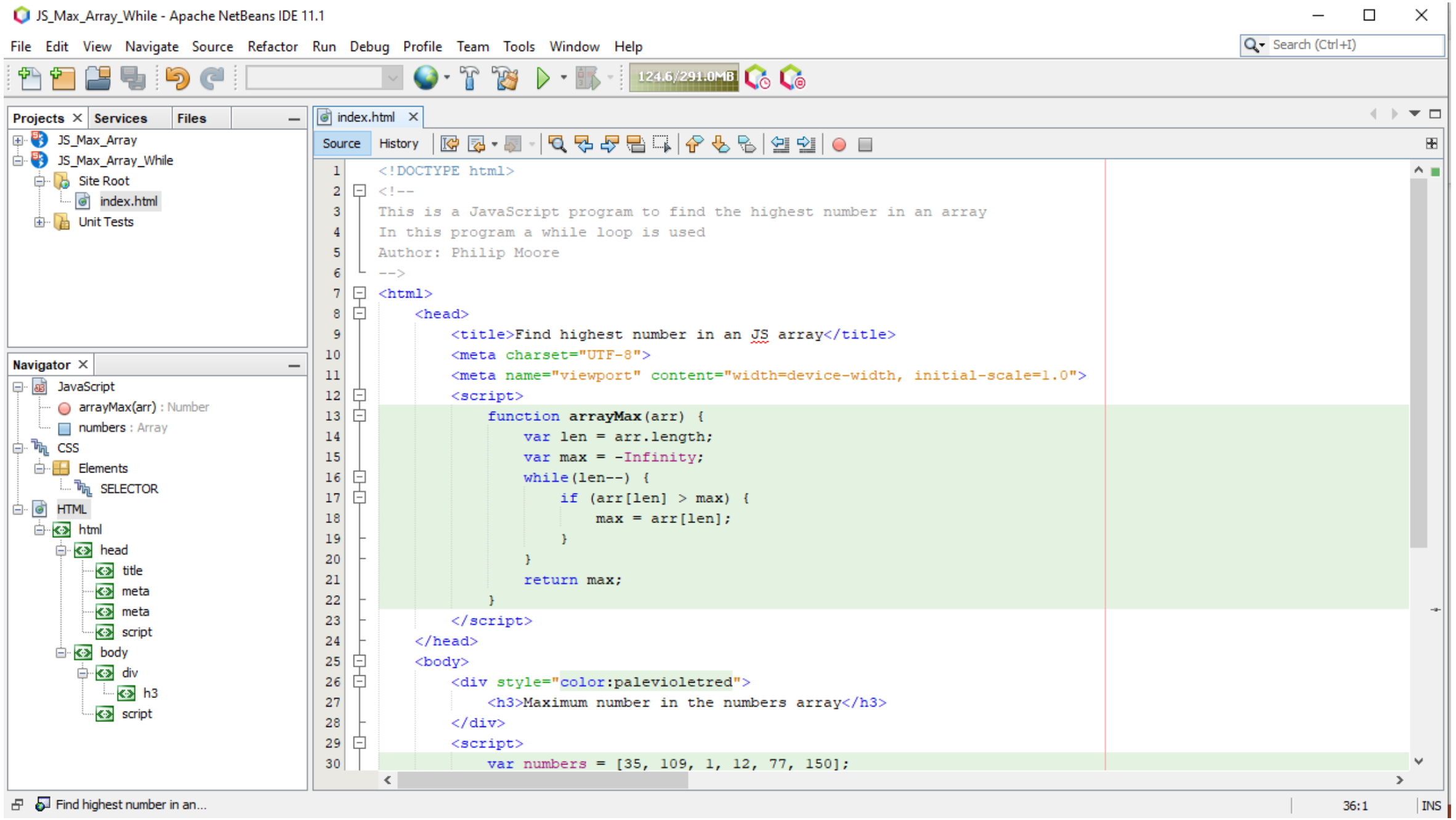

Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
136.3/252.0MB
Projects
Services
Files
JS_Max_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMax(arr)
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the highest number in an array
4 Author: Philip Moore
5 -->
6 <html>
7 <head>
8 <title>Find highest number in an JS array</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <script>
12 function arrayMax(arr) {
13 return Math.max.apply(null, arr);
14 }
15 </script>
16 </head>
17 <body>
18
<h3>Maximum number in the numbers array</h3>
20
21 <script>
22 var numbers = [35, 109, 1, 12, 77, 150];
23 document.write("The highest number in the array is: " +
24 arrayMax(numbers));
25 </script>
26 </body>
27 </html>
28
Find highest number in an...
19:52
INS

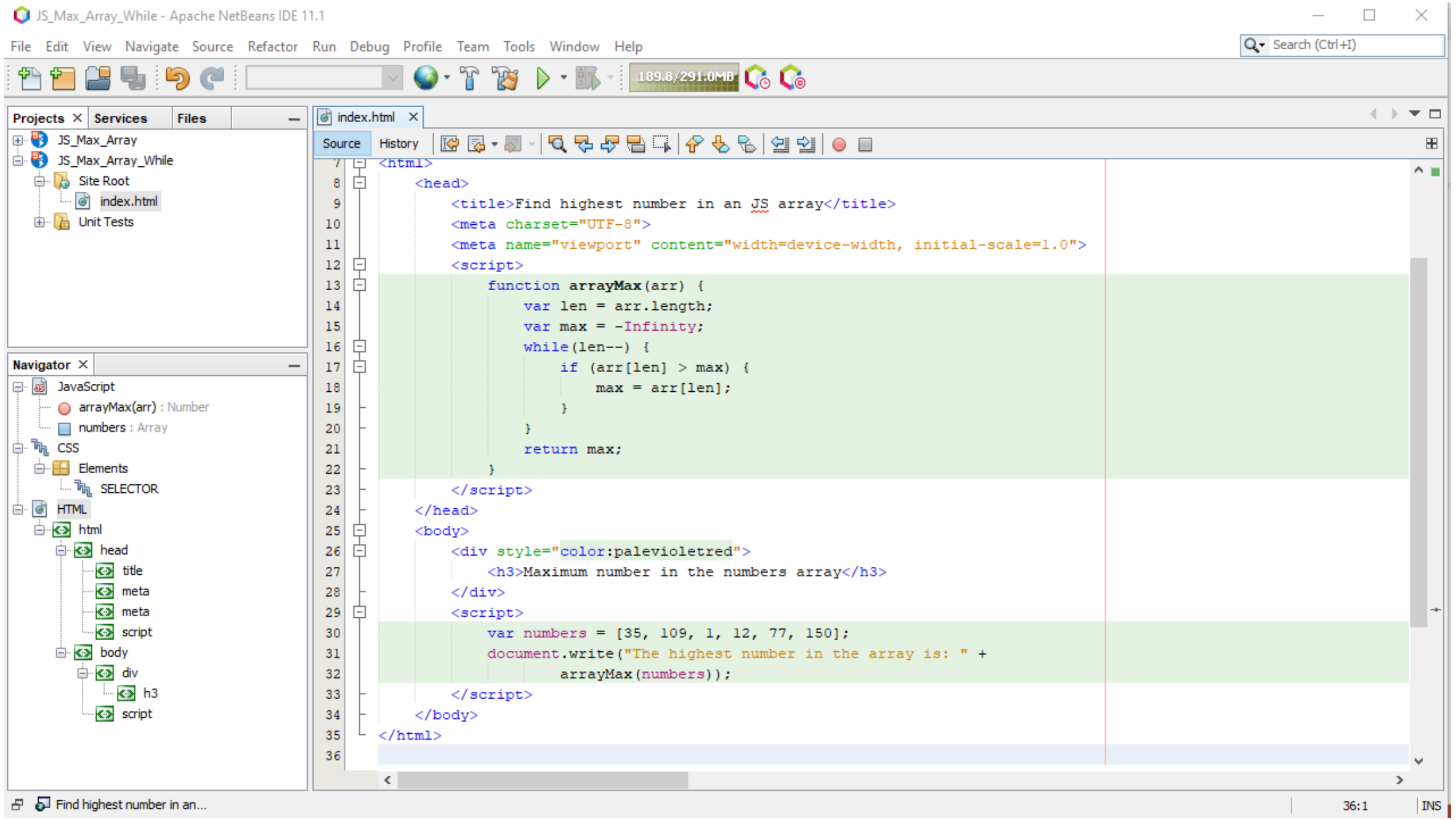

Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
131.0/252.0MB
Projects
Services
Files
JS_Max_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMax(arr)
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the highest number in an array
4 Author: Philip Moore
5 -->
6 <html>
7 <head>
8 <title>Find highest number in an JS array</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <script>
12 function arrayMax(arr) {
13 return Math.max.apply(null, arr);
14 }
15 </script>
16 </head>
17 <body>
18
<h3>Maximum number in the numbers array</h3>
20
Find highest number in an...
http://localhost:8383/JS_Max_Array/index.html
100%
Maximum number in the numbers array
The highest number in the array is: 150
19:52
INS

A Max Function for Arrays

- This function uses a while loop to access an array and compare each value to find the highest number

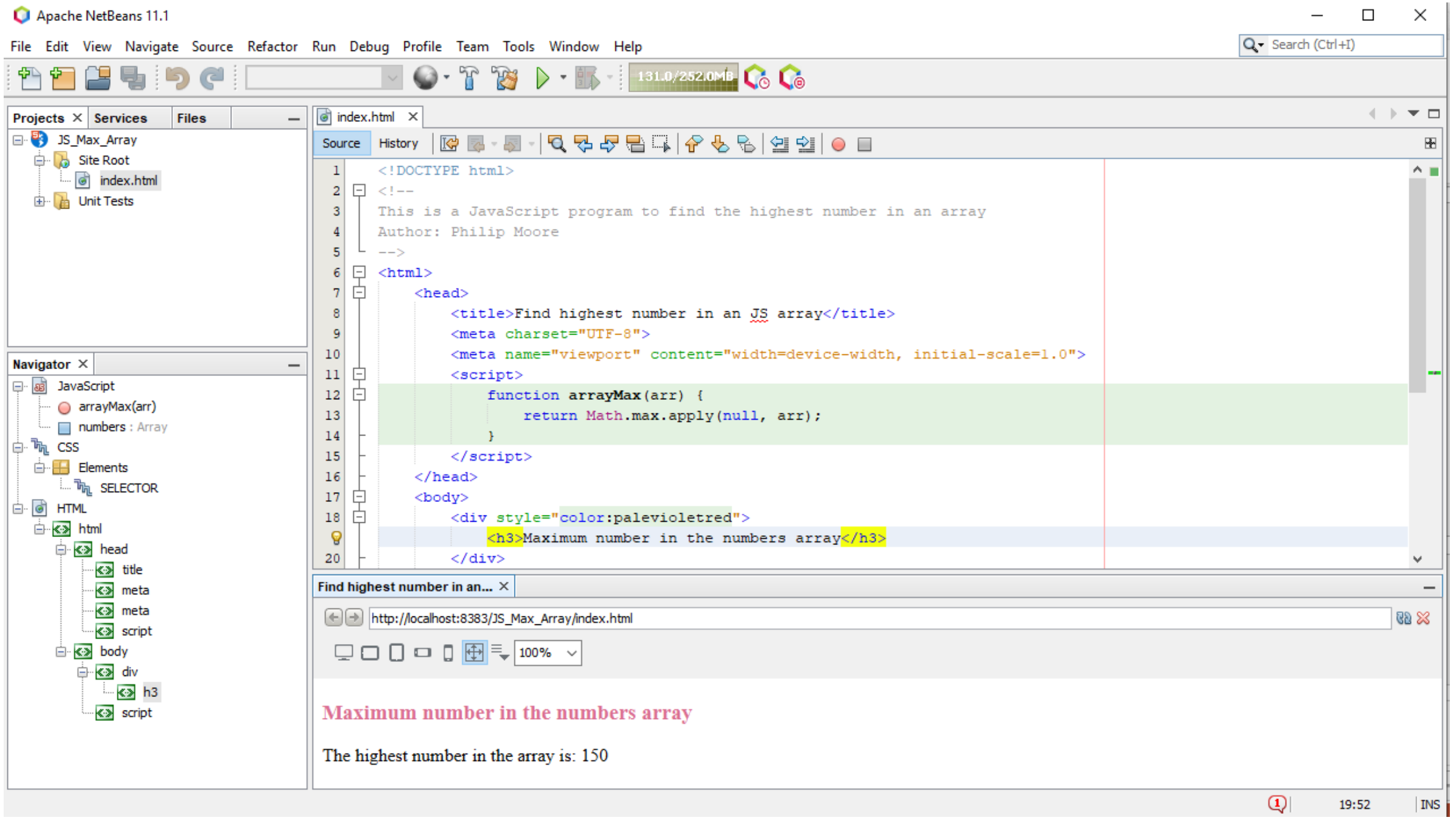
```
function myArrayMax(arr) {
      var len = arr.length
      var max = -Infinity;
      while(len--) { // note: the decrementing (len--)
            if (arr[len] > max) { //note the if statement
                  max = arr[len];
            }
      }
return max; //the return statement
}
```


JS_Max_Array_While - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
124.6/291.0MB
Projects
Services
Files
JS_Max_Array
JS_Max_Array_While
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMax(arr) : Number
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the highest number in an array
4 In this program a while loop is used
5 Author: Philip Moore
6 -->
7 <html>
8 <head>
9 <title>Find highest number in an JS array</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <script>
13 function arrayMax(arr) {
14 var len = arr.length;
15 var max = -Infinity;
16 while(len--) {
17 if (arr[len] > max) {
18 max = arr[len];
19 }
20 }
21 return max;
22 }
23 </script>
24 </head>
25 <body>
26
27 <h3>Maximum number in the numbers array</h3>
28
29 <script>
30 var numbers = [35, 109, 1, 12, 77, 150];
Find highest number in an...
36:1
INS


JS_Max_Array_While - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
189.8/291.0MB
Projects
Services
Files
JS_Max_Array
JS_Max_Array_While
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMax(arr) : Number
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
Source
History
<html>
<head>
<title>Find highest number in an JS array</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<script>
function arrayMax(arr) {
var len = arr.length;
var max = -Infinity;
while(len--) {
if (arr[len] > max) {
max = arr[len];
}
}
return max;
}
</script>
</head>
<body>

<h3>Maximum number in the numbers array</h3>

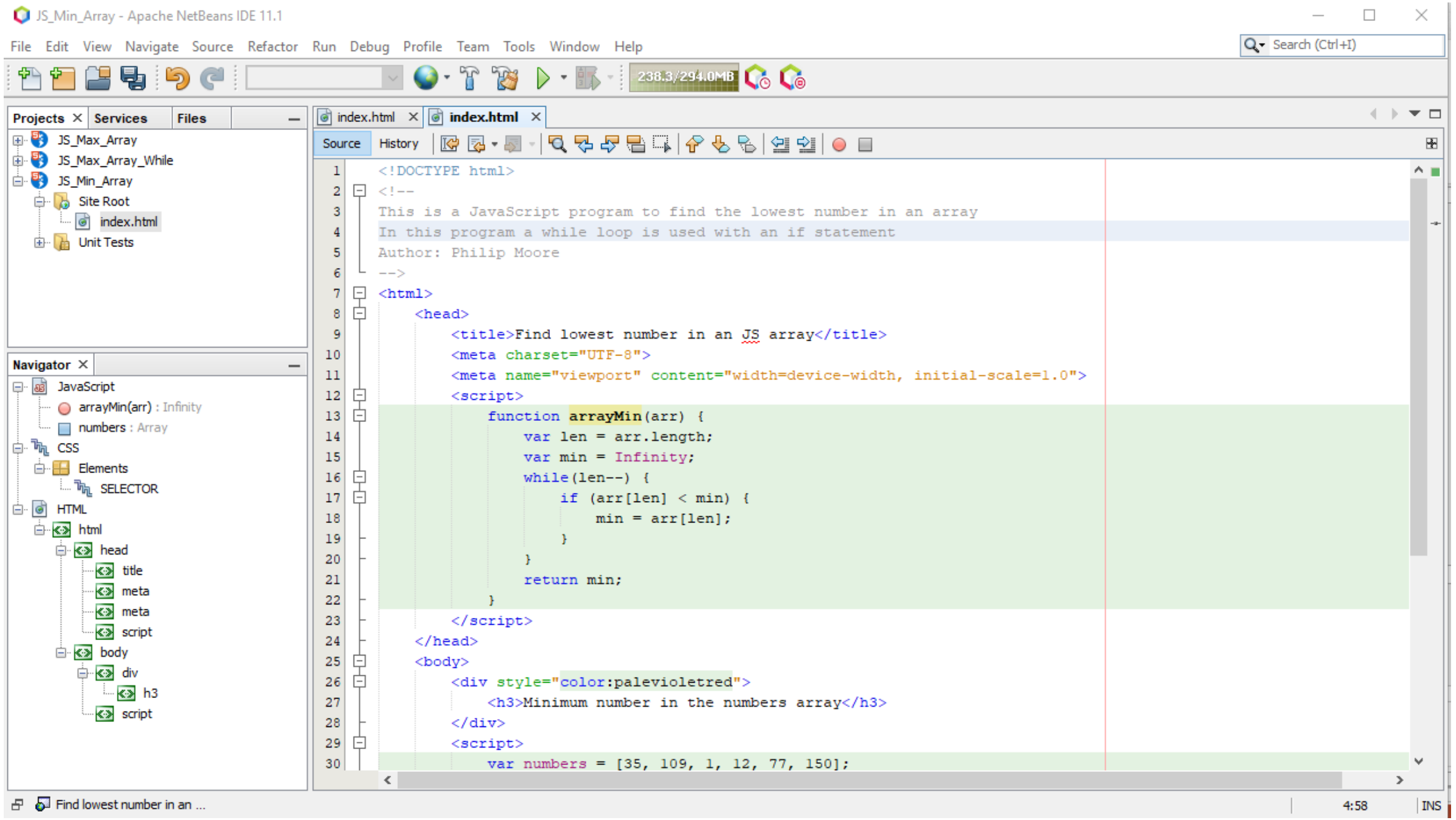
<script>
var numbers = [35, 109, 1, 12, 77, 150];
document.write("The highest number in the array is: " +
arrayMax(numbers));
</script>
</body>
</html>
Find highest number in an...
36:1
INS

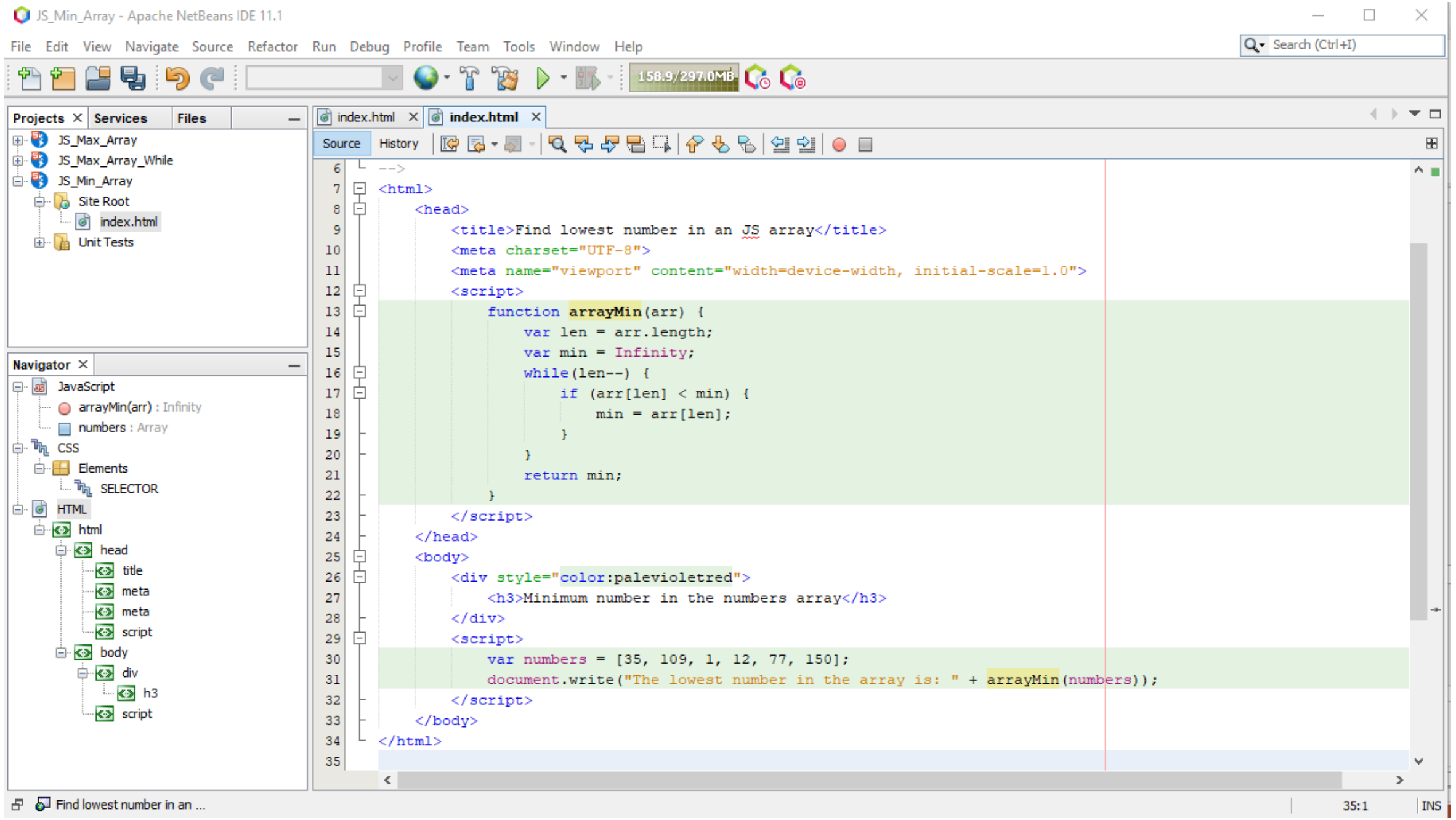

Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
131.0/252.0MB
Projects
Services
Files
JS_Max_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMax(arr)
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the highest number in an array
4 Author: Philip Moore
5 -->
6 <html>
7 <head>
8 <title>Find highest number in an JS array</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <script>
12 function arrayMax(arr) {
13 return Math.max.apply(null, arr);
14 }
15 </script>
16 </head>
17 <body>
18
<h3>Maximum number in the numbers array</h3>
20
Find highest number in an...
http://localhost:8383/JS_Max_Array/index.html
100%
Maximum number in the numbers array
The highest number in the array is: 150
19:52
INS

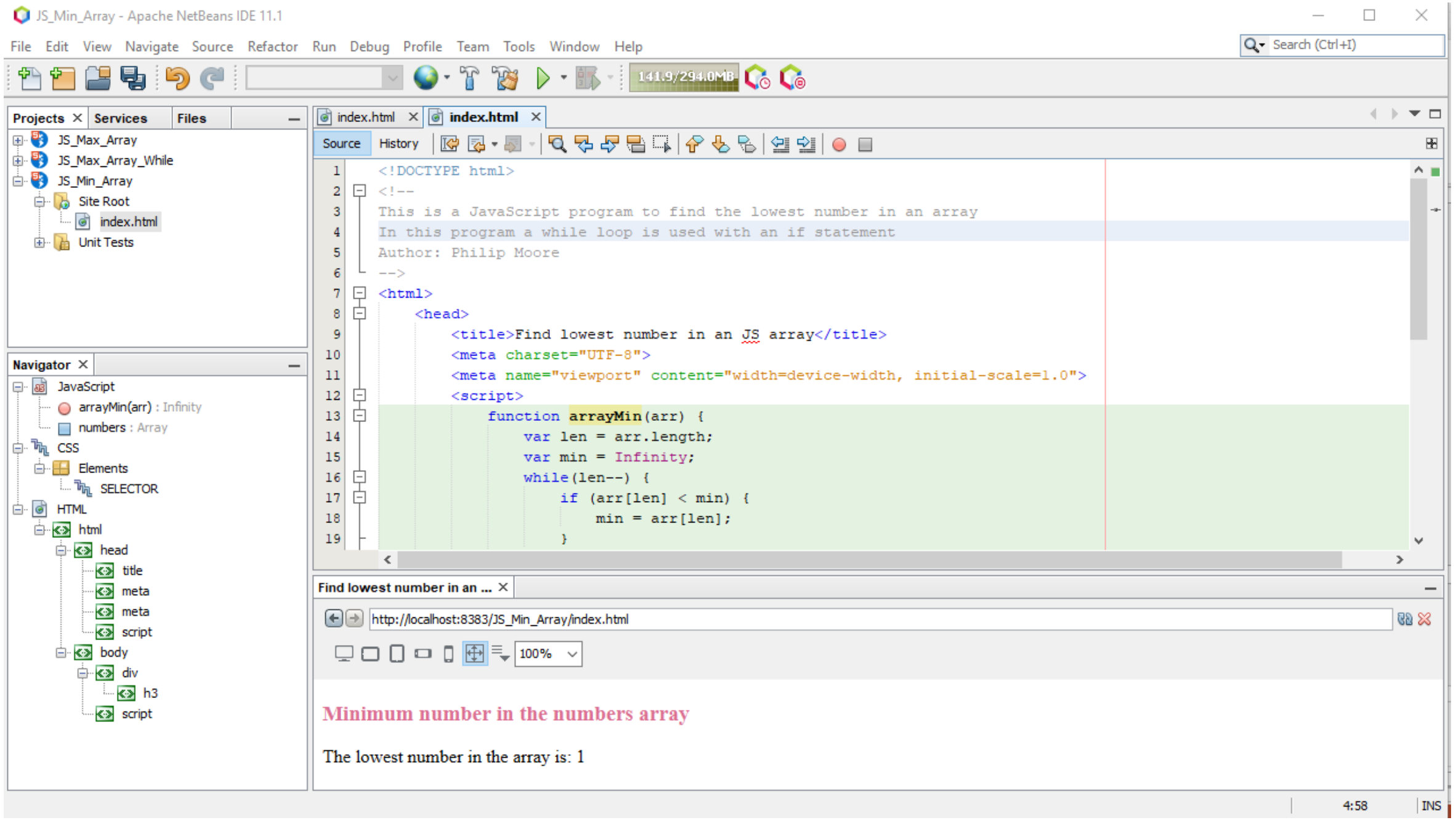
A Min Function for Arrays

- This function uses a while loop to access an array and compare each value to find the lowest number

```
function myArrayMax(arr) {
      var len = arr.length
      var max = Infinity;
      while (len--) {// note: the decrementing (len--)
            if (arr[len] < min) {//note the if statement
                  max = arr[len];
            }
      }
return min; // the return statement
}
```

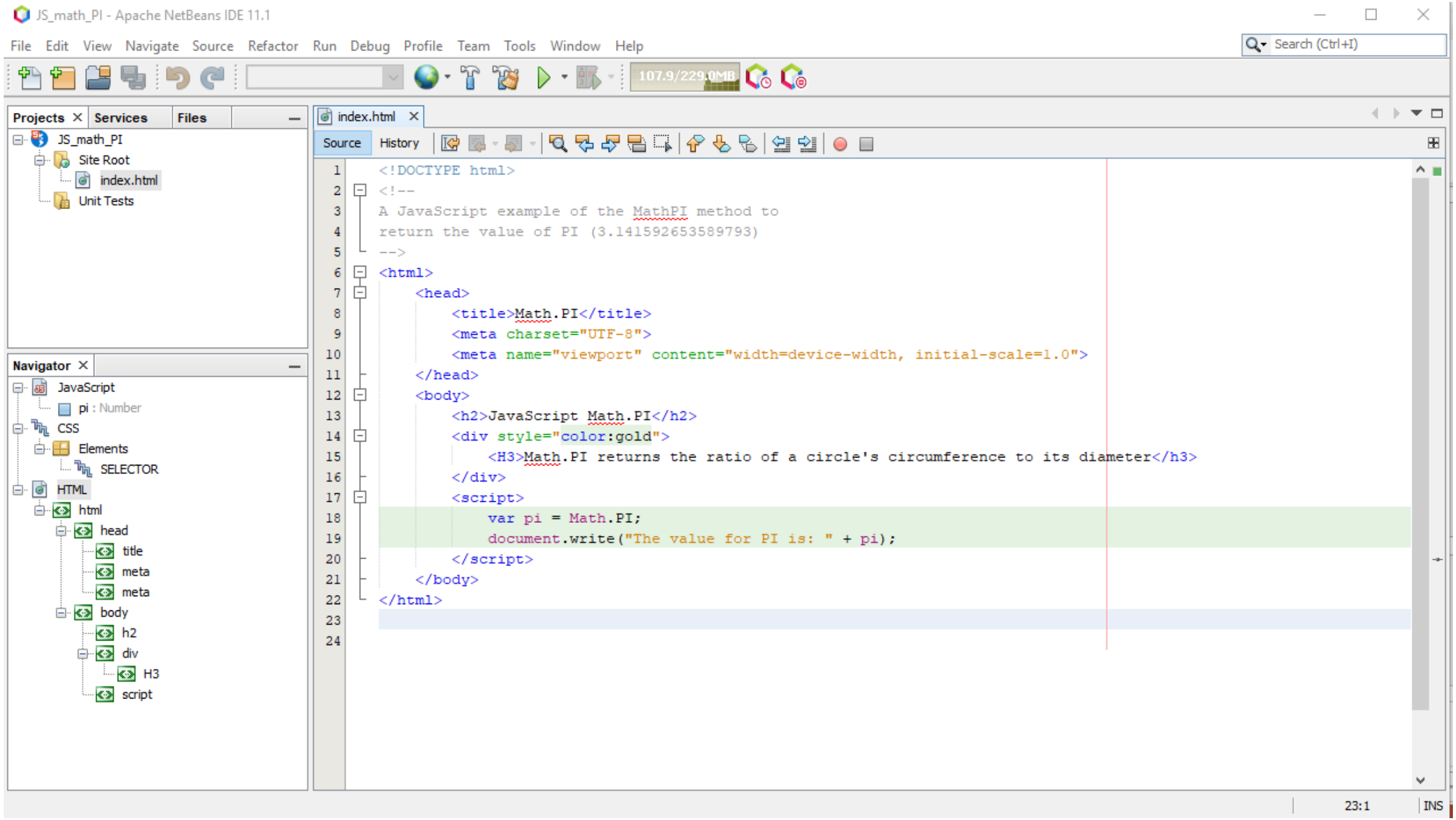

JS_Min_Array - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
238.3/294.0MB
Projects
Services
Files
JS_Max_Array
JS_Max_Array_While
JS_Min_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMin(arr) : Infinity
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the lowest number in an array
4 In this program a while loop is used with an if statement
5 Author: Philip Moore
6 -->
7 <html>
8 <head>
9 <title>Find lowest number in an JS array</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <script>
13 function arrayMin(arr) {
14 var len = arr.length;
15 var min = Infinity;
16 while(len--) {
17 if (arr[len] < min) {
18 min = arr[len];
19 }
20 }
21 return min;
22 }
23 </script>
24 </head>
25 <body>
26
27 <h3>Minimum number in the numbers array</h3>
28
29 <script>
30 var numbers = [35, 109, 1, 12, 77, 150];
Find lowest number in an ...
4:58
INS

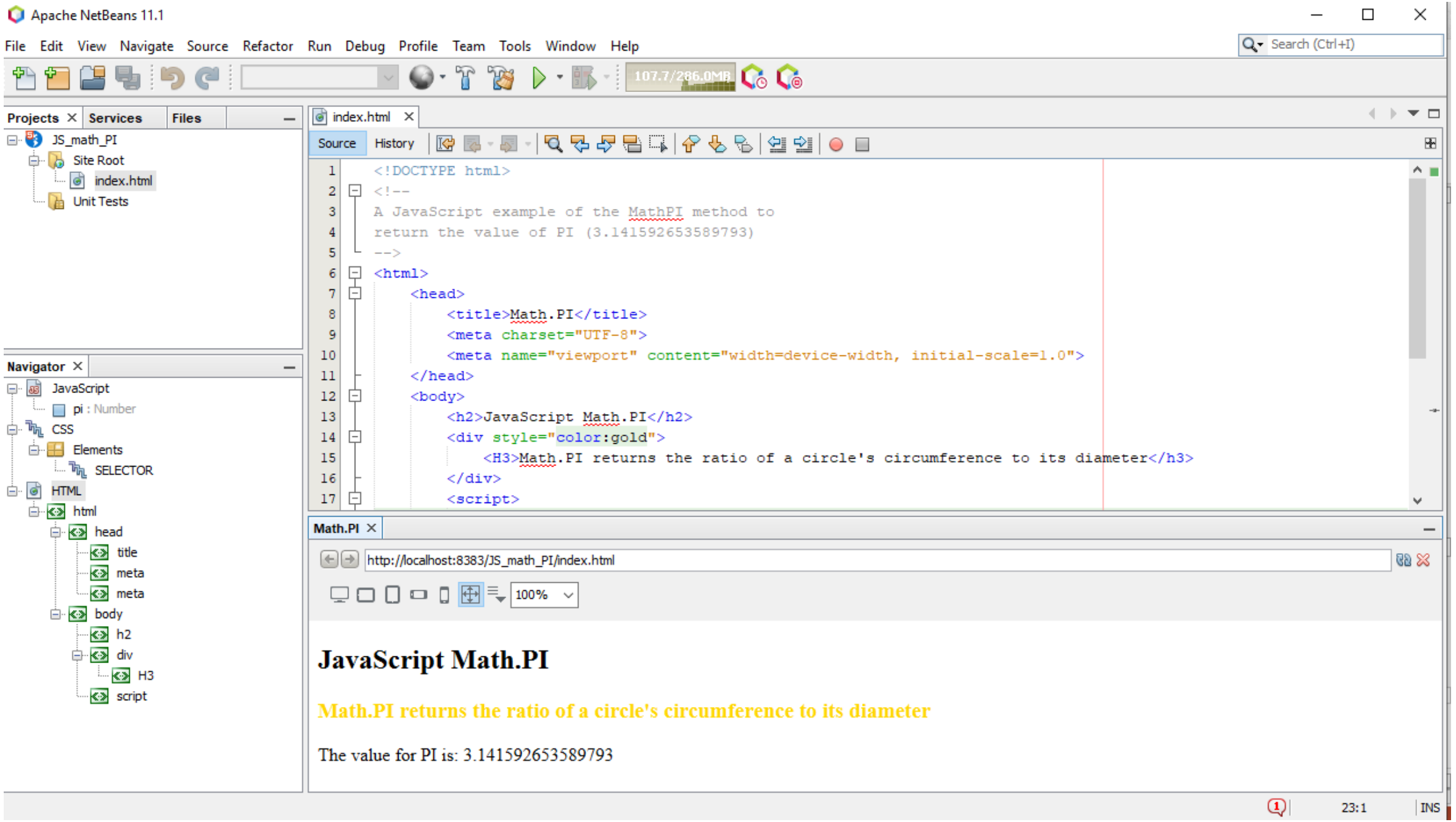

JS_Min_Array - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
158.9/297.0MB
Projects
Services
Files
JS_Max_Array
JS_Max_Array_While
JS_Min_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMin(arr) : Infinity
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
index.html
Source
History
6 -->
7 <html>
8 <head>
9 <title>Find lowest number in an JS array</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <script>
13 function arrayMin(arr) {
14 var len = arr.length;
15 var min = Infinity;
16 while(len--) {
17 if (arr[len] < min) {
18 min = arr[len];
19 }
20 }
21 return min;
22 }
23 </script>
24 </head>
25 <body>
26
27 <h3>Minimum number in the numbers array</h3>
28
29 <script>
30 var numbers = [35, 109, 1, 12, 77, 150];
31 document.write("The lowest number in the array is: " + arrayMin(numbers));
32 </script>
33 </body>
34 </html>
35
Find lowest number in an ...
35:1
INS

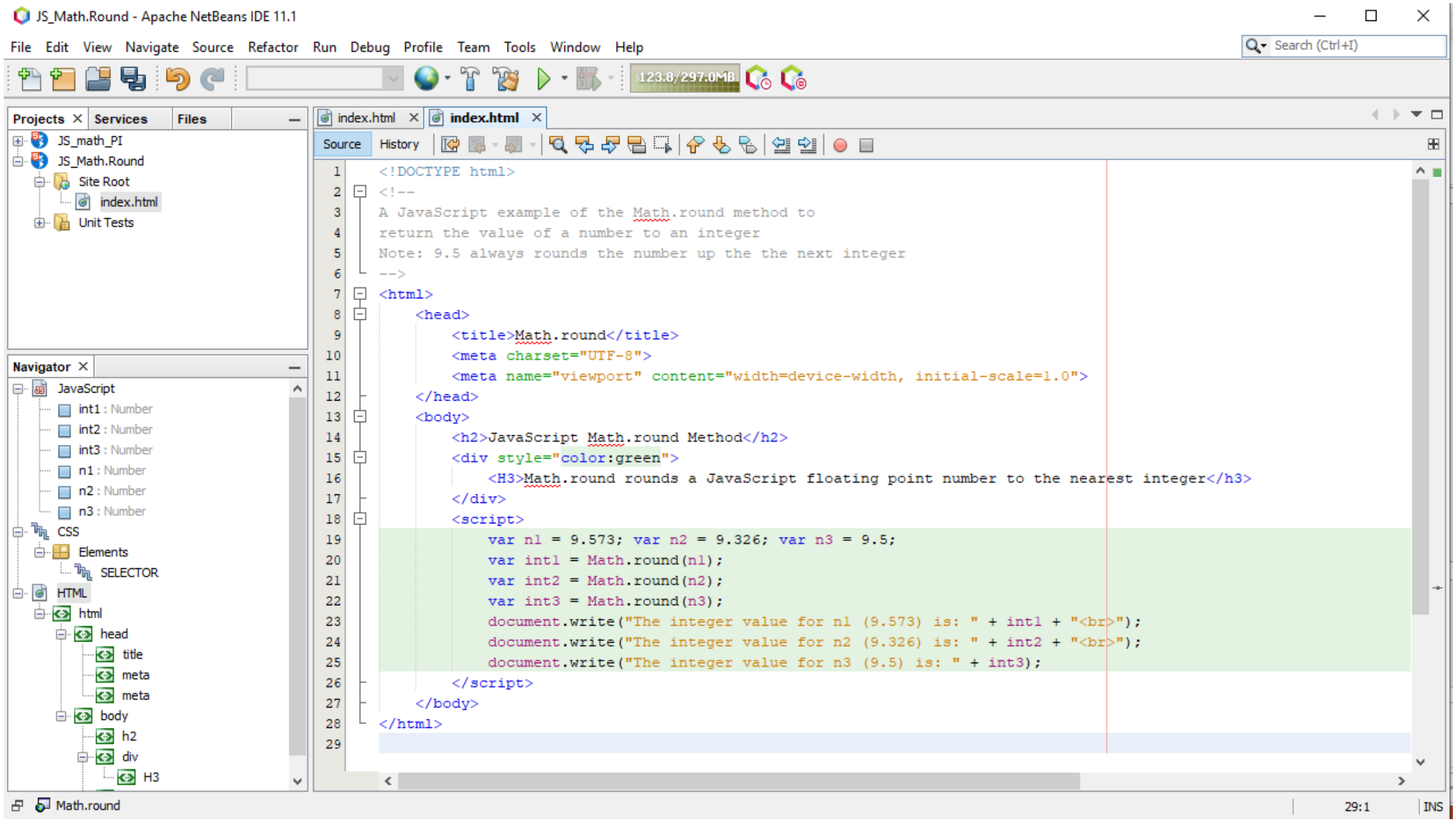

JS_Min_Array - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
141.9/294.0MB
Projects
Services
Files
JS_Max_Array
JS_Max_Array_While
JS_Min_Array
Site Root
index.html
Unit Tests
Navigator
JavaScript
arrayMin(arr) : Infinity
numbers : Array
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h3
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 This is a JavaScript program to find the lowest number in an array
4 In this program a while loop is used with an if statement
5 Author: Philip Moore
6 -->
7 <html>
8 <head>
9 <title>Find lowest number in an JS array</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <script>
13 function arrayMin(arr) {
14 var len = arr.length;
15 var min = Infinity;
16 while(len--) {
17 if (arr[len] < min) {
18 min = arr[len];
19 }
Find lowest number in an ...
http://localhost:8383/JS_Min_Array/index.html
100%
Minimum number in the numbers array
The lowest number in the array is: 1
4:58
INS

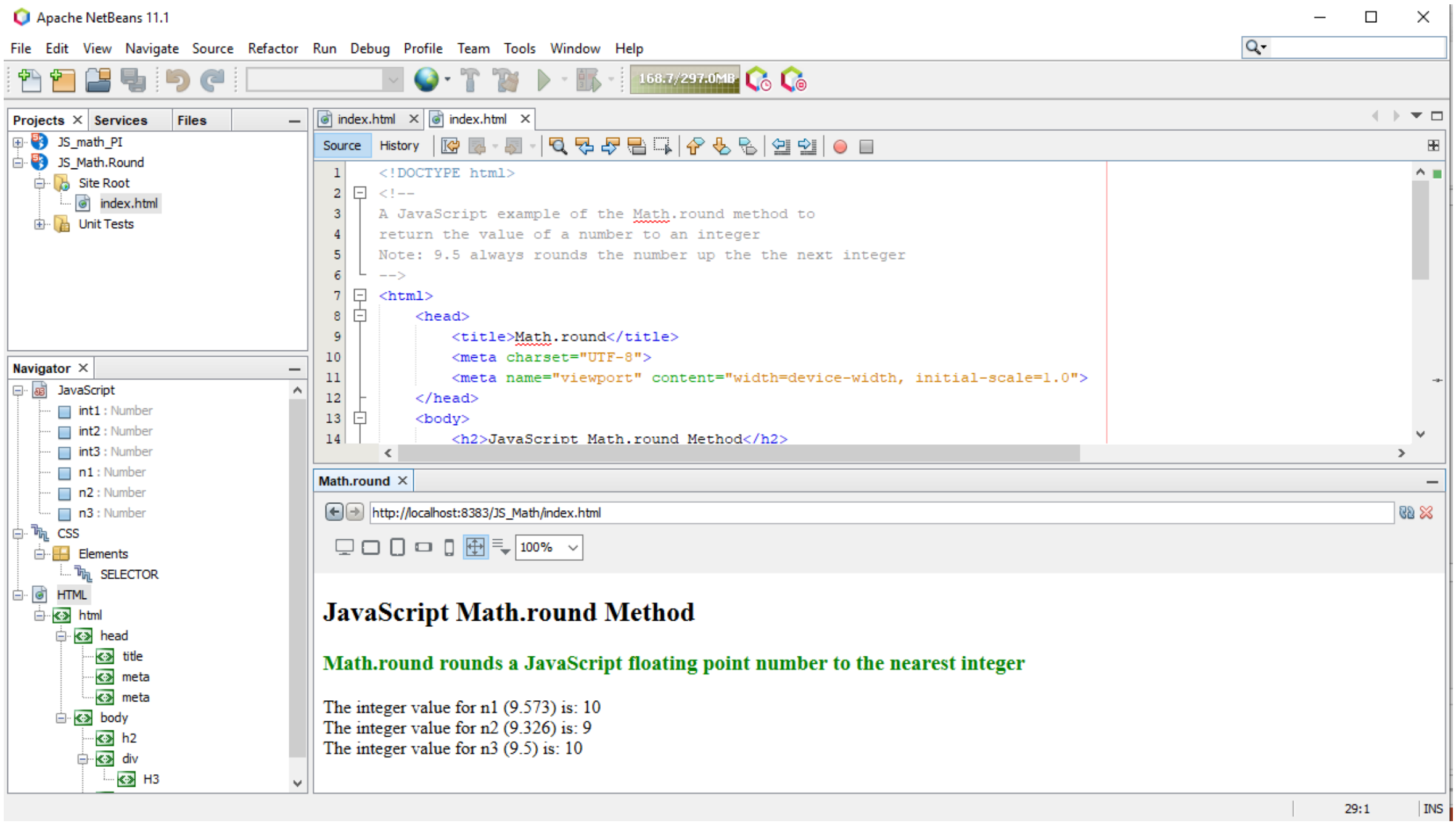
Math Methods

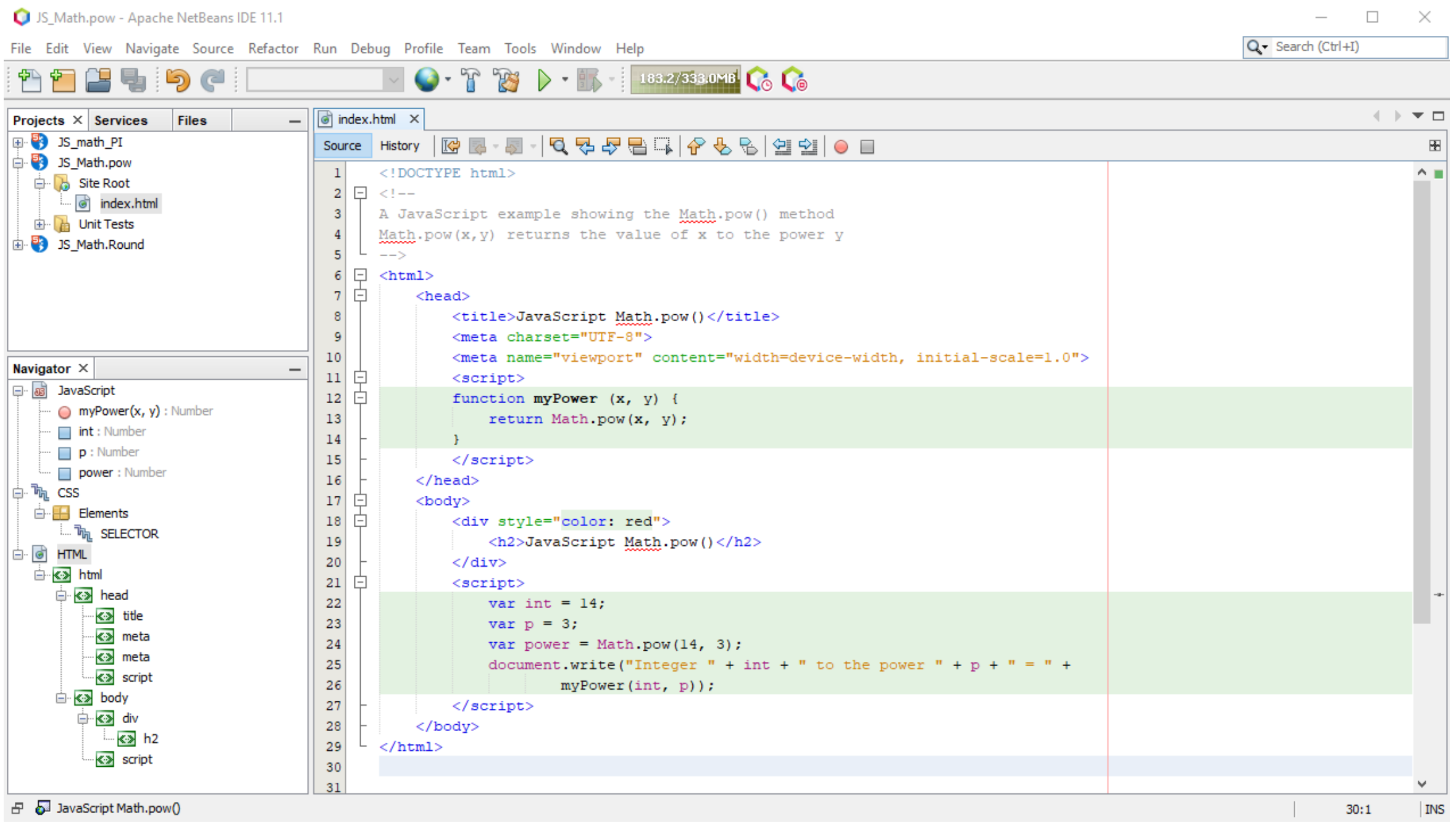
- There are many math methods for implementing generally used mathematic calculations
- A frequent use of Math methods is the generation of random numbers
- The following slides demonstrate some frequently used math methods
  - **`Math.max()` / `Math.min()` / `Math.max.apply` / `Math.round()` / `Math.sqrt()` / `Math.random()` / `Math.pow()` / `...`**
  - **`Math.PI`** shows the use of a GLOBAL variable as a constant
- Comprehensive details of the Math methods may be found in the course resources

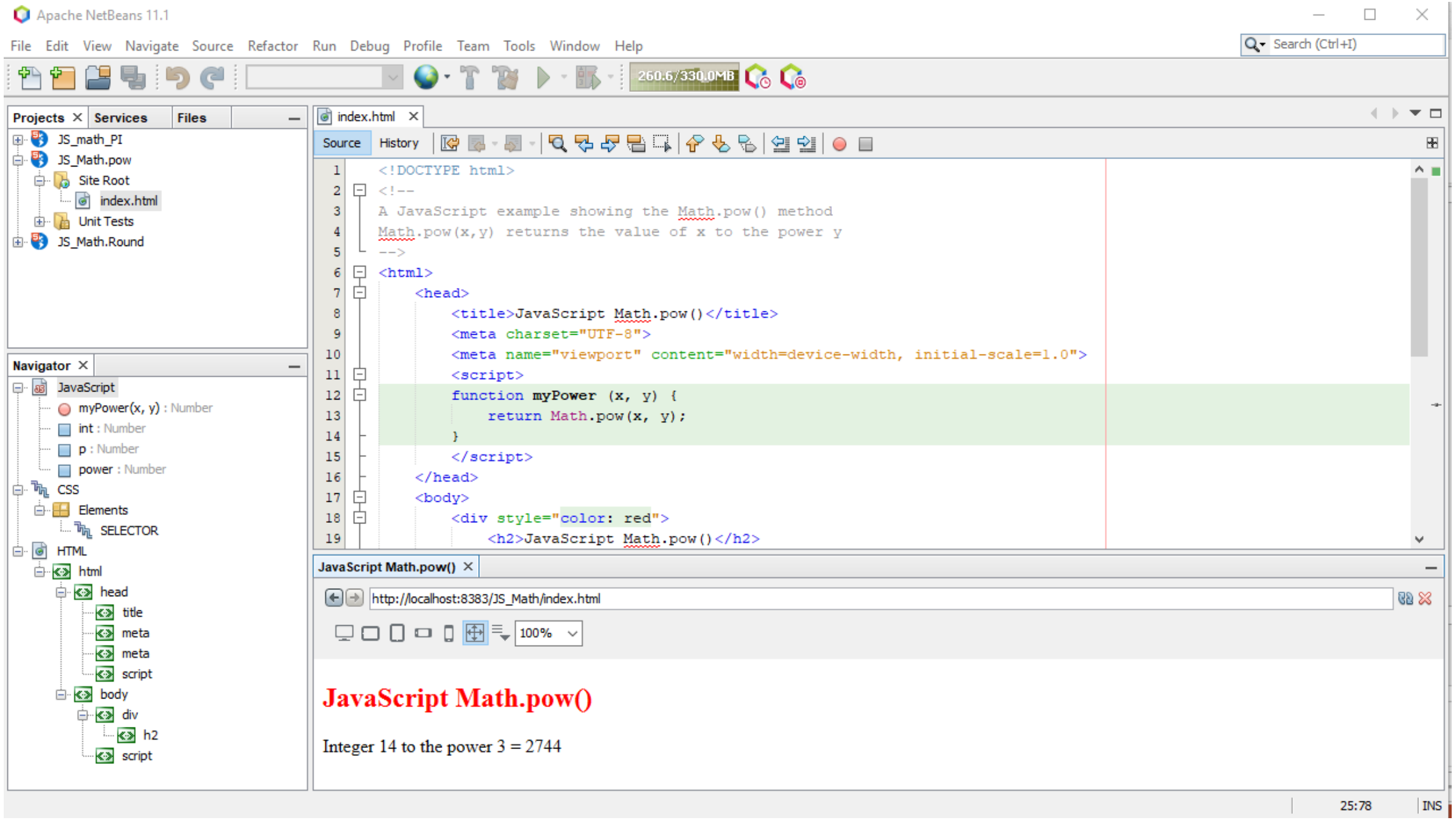

JS_math_PI - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
107.9/229.0MB
Projects
Services
Files
JS_math_PI
Site Root
index.html
Unit Tests
Navigator
JavaScript
pi : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
body
h2
div
H3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of the MathPI method to
4 return the value of PI (3.141592653589793)
5 -->
6 <html>
7 <head>
8 <title>Math.PI</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 </head>
12 <body>
13 <h2>JavaScript Math.PI</h2>
14
15 <H3>Math.PI returns the ratio of a circle's circumference to its diameter</h3>
16
17 <script>
18 var pi = Math.PI;
19 document.write("The value for PI is: " + pi);
20 </script>
21 </body>
22 </html>
23
24
23:1
INS

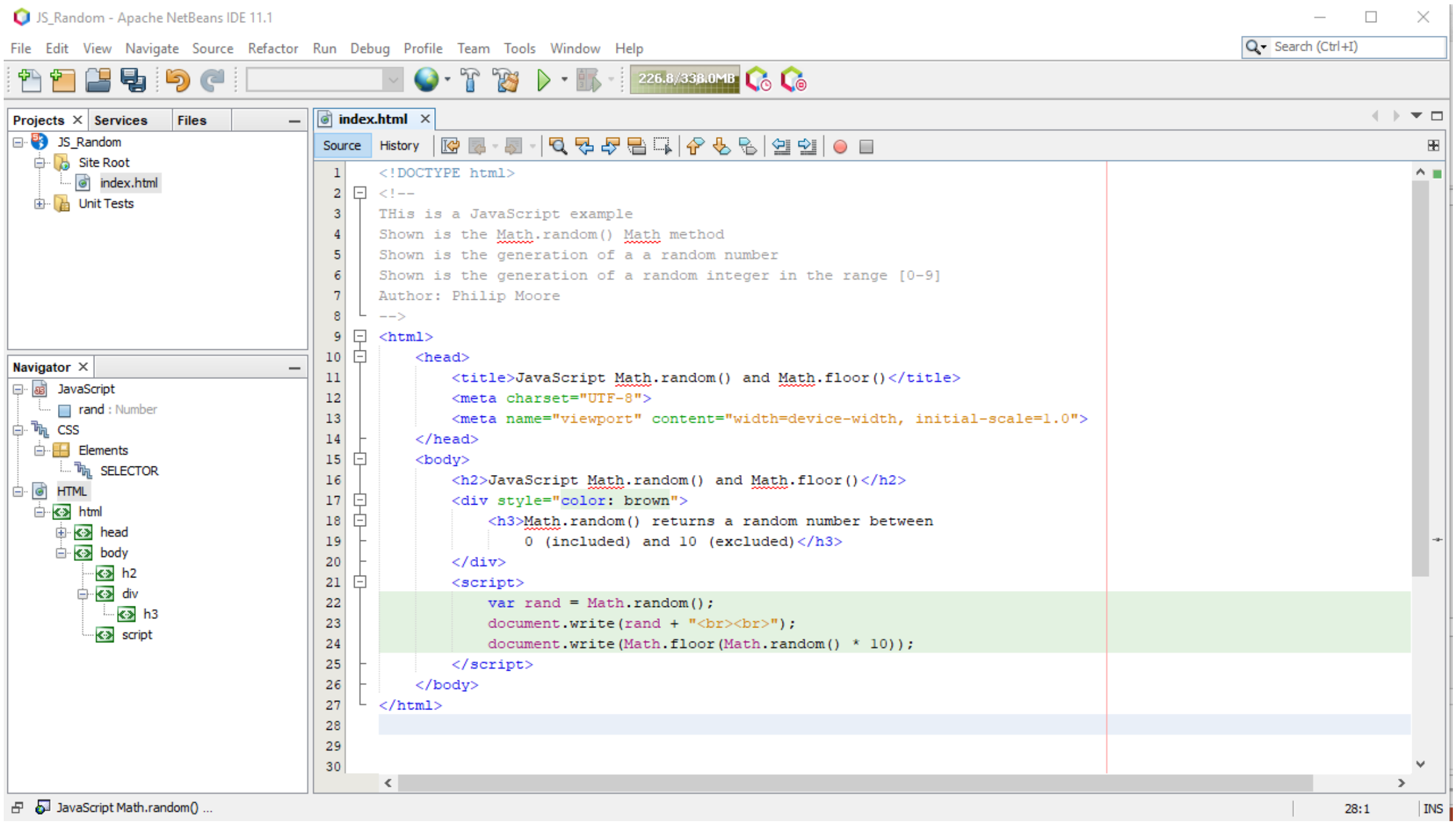

Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
107.7/286.0MB
Projects
Services
Files
JS_math_PI
Site Root
index.html
Unit Tests
Navigator
JavaScript
pi : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
body
h2
div
H3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of the MathPI method to
4 return the value of PI (3.141592653589793)
5 -->
6 <html>
7 <head>
8 <title>Math.PI</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 </head>
12 <body>
13 <h2>JavaScript Math.PI</h2>
14
15 <H3>Math.PI returns the ratio of a circle's circumference to its diameter</h3>
16
17 <script>
Math.PI
http://localhost:8383/JS_math_PI/index.html
100%
JavaScript Math.PI
Math.PI returns the ratio of a circle's circumference to its diameter
The value for PI is: 3.141592653589793
23:1
INS


JS_Math.Round - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
123.8/297.0MB
Projects
Services
Files
JS_math_PI
JS_Math.Round
Site Root
index.html
Unit Tests
Navigator
JavaScript
int1 : Number
int2 : Number
int3 : Number
n1 : Number
n2 : Number
n3 : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
body
h2
div
H3
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of the Math.round method to
4 return the value of a number to an integer
5 Note: 9.5 always rounds the number up the the next integer
6 -->
7 <html>
8 <head>
9 <title>Math.round</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>JavaScript Math.round Method</h2>
15
16 <H3>Math.round rounds a JavaScript floating point number to the nearest integer</h3>
17
18 <script>
19 var n1 = 9.573; var n2 = 9.326; var n3 = 9.5;
20 var int1 = Math.round(n1);
21 var int2 = Math.round(n2);
22 var int3 = Math.round(n3);
23 document.write("The integer value for n1 (9.573) is: " + int1 + "
");
24 document.write("The integer value for n2 (9.326) is: " + int2 + "
");
25 document.write("The integer value for n3 (9.5) is: " + int3);
26 </script>
27 </body>
28 </html>
29
Math.round
29:1
INS

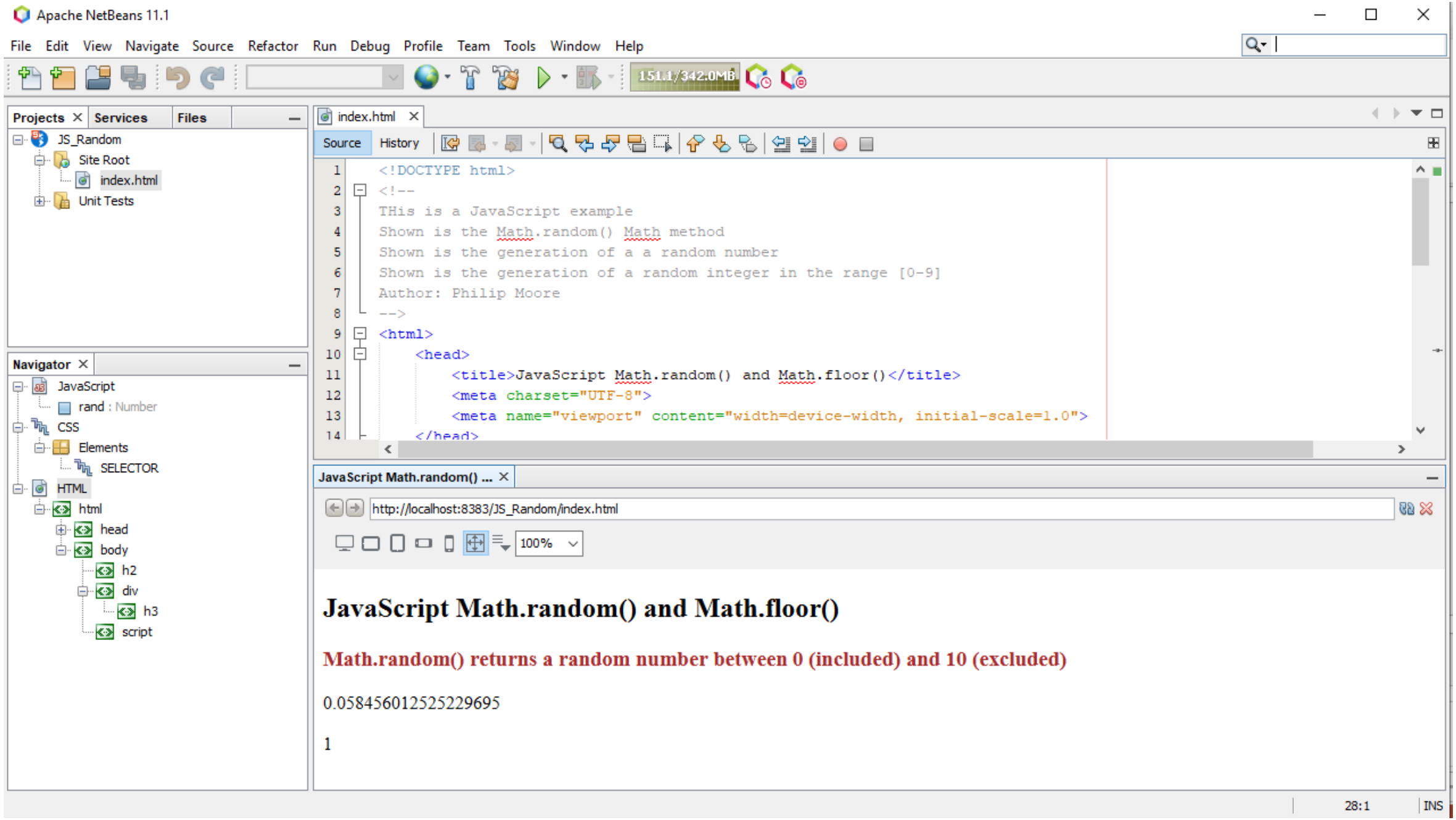

Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
168.7/297.0MB
Projects Services Files
JS_math_PI
JS_Math.Round
Site Root
index.html
Unit Tests
Navigator
JavaScript
int1 : Number
int2 : Number
int3 : Number
n1 : Number
n2 : Number
n3 : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
body
h2
div
H3
index.html
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of the Math.round method to
4 return the value of a number to an integer
5 Note: 9.5 always rounds the number up the the next integer
6 -->
7 <html>
8 <head>
9 <title>Math.round</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <h2>JavaScript Math.round Method</h2>
Math.round
http://localhost:8383/JS_Math/index.html
100%
JavaScript Math.round Method
Math.round rounds a JavaScript floating point number to the nearest integer
The integer value for n1 (9.573) is: 10
The integer value for n2 (9.326) is: 9
The integer value for n3 (9.5) is: 10
29:1
INS

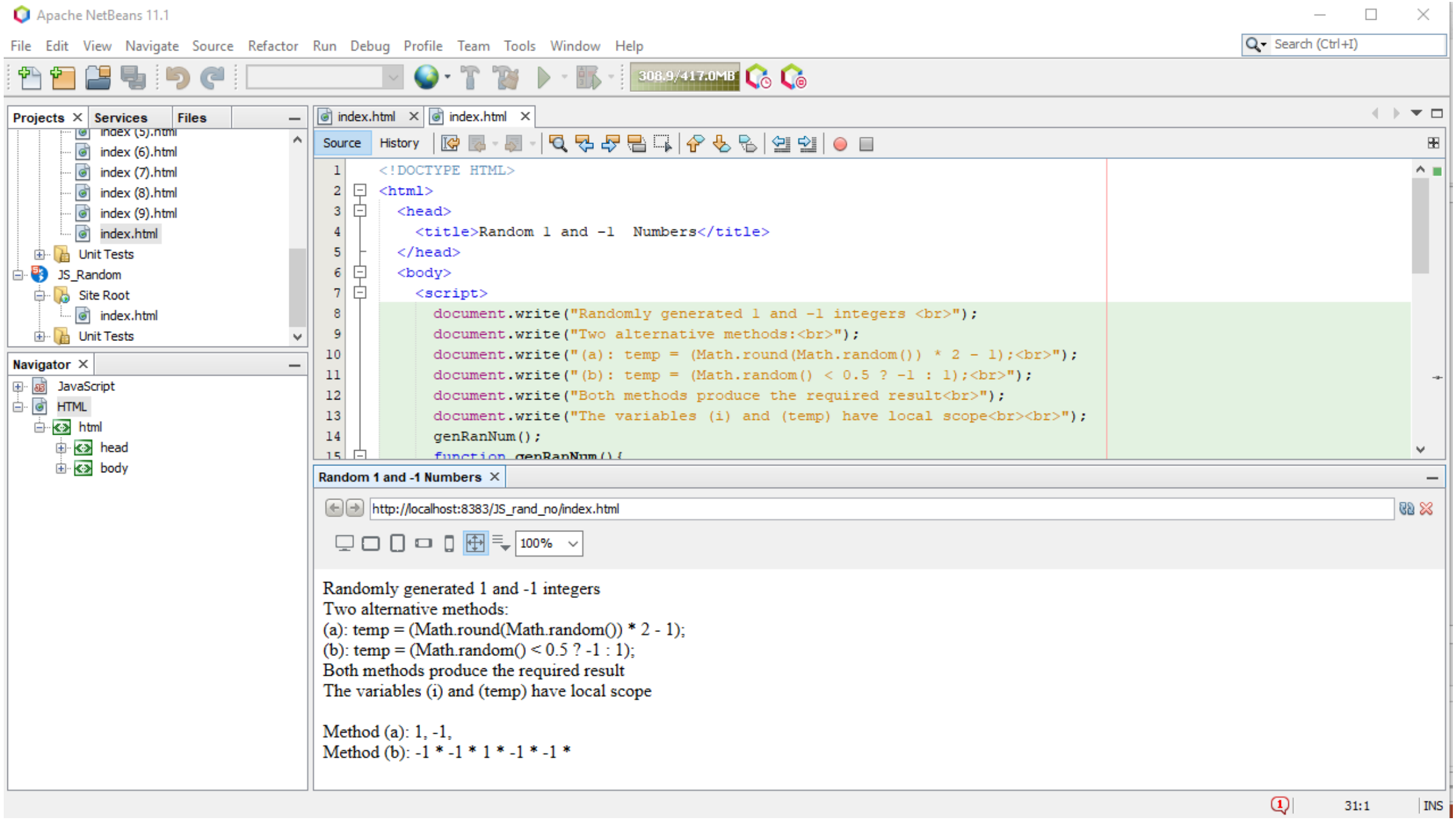

JS_Math.pow - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
183.2/333.0MB
Projects
Services
Files
JS_math_PI
JS_Math.pow
Site Root
index.html
Unit Tests
JS_Math.Round
Navigator
JavaScript
myPower(x, y) : Number
int : Number
p : Number
power : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h2
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example showing the Math.pow() method
4 Math.pow(x,y) returns the value of x to the power y
5 -->
6 <html>
7 <head>
8 <title>JavaScript Math.pow()</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <script>
12 function myPower (x, y) {
13 return Math.pow(x, y);
14 }
15 </script>
16 </head>
17 <body>
18
19 <h2>JavaScript Math.pow()</h2>
20
21 <script>
22 var int = 14;
23 var p = 3;
24 var power = Math.pow(14, 3);
25 document.write("Integer " + int + " to the power " + p + " = " +
26 myPower(int, p));
27 </script>
28 </body>
29 </html>
30
31
JavaScript Math.pow()
30:1
INS


Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
260.6/330.0MB
Projects
Services
Files
JS_math_PI
JS_Math.pow
Site Root
index.html
Unit Tests
JS_Math.Round
Navigator
JavaScript
myPower(x, y) : Number
int : Number
p : Number
power : Number
CSS
Elements
SELECTOR
HTML
html
head
title
meta
meta
script
body
div
h2
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example showing the Math.pow() method
4 Math.pow(x,y) returns the value of x to the power y
5 -->
6 <html>
7 <head>
8 <title>JavaScript Math.pow()</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 <script>
12 function myPower (x, y) {
13 return Math.pow(x, y);
14 }
15 </script>
16 </head>
17 <body>
18
19 <h2>JavaScript Math.pow()</h2>
JavaScript Math.pow()
http://localhost:8383/JS_Math/index.html
100%
JavaScript Math.pow()
Integer 14 to the power 3 = 2744
25:78
INS

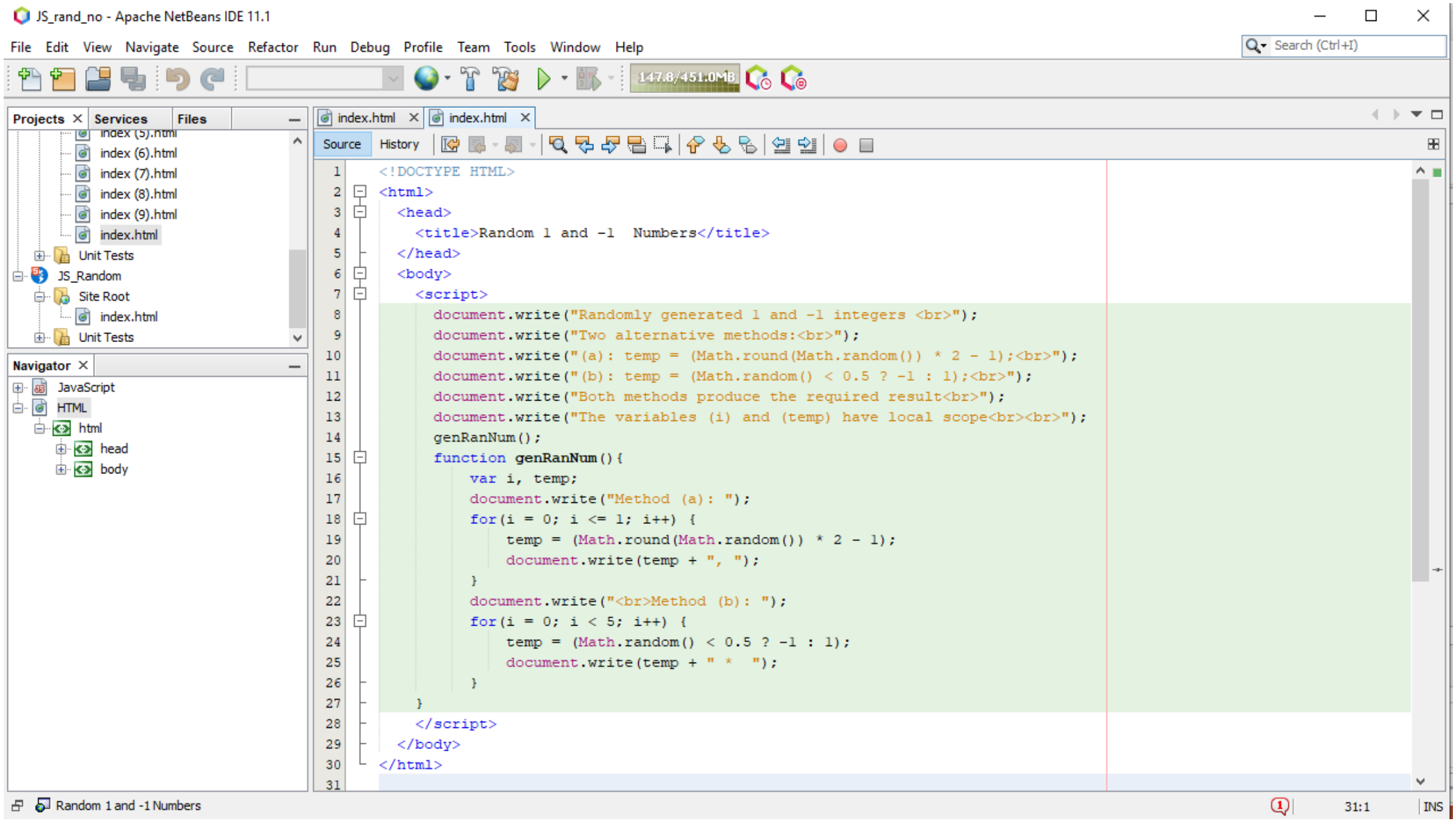

JS_Random - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
226.8/338.0MB
Projects
Services
Files
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
rand : Number
CSS
Elements
SELECTOR
HTML
html
head
body
h2
div
h3
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 THis is a JavaScript example
4 Shown is the Math.random() Math method
5 Shown is the generation of a a random number
6 Shown is the generation of a random integer in the range [0-9]
7 Author: Philip Moore
8 -->
9 <html>
10 <head>
11 <title>JavaScript Math.random() and Math.floor()</title>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 </head>
15 <body>
16 <h2>JavaScript Math.random() and Math.floor()</h2>
17
18 <h3>Math.random() returns a random number between
19 0 (included) and 10 (excluded)</h3>
20
21 <script>
22 var rand = Math.random();
23 document.write(rand + "

");
24 document.write(Math.floor(Math.random() * 10));
25 </script>
26 </body>
27 </html>
28
29
30
JavaScript Math.random() ...
28:1
INS


Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
151.1/342.0MB
Projects Services Files
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
rand : Number
CSS
Elements
SELECTOR
HTML
html
head
body
h2
div
h3
script
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 THis is a JavaScript example
4 Shown is the Math.random() Math method
5 Shown is the generation of a a random number
6 Shown is the generation of a random integer in the range [0-9]
7 Author: Philip Moore
8 -->
9 <html>
10 <head>
11 <title>JavaScript Math.random() and Math.floor()</title>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 </head>
JavaScript Math.random() ...
http://localhost:8383/JS_Random/index.html
100%
JavaScript Math.random() and Math.floor()
Math.random() returns a random number between 0 (included) and 10 (excluded)
0.058456012525229695
1
28:1
INS


Apache NetBeans 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
308.9/417.0MB
Projects
Services
Files
index (6).html
index (7).html
index (8).html
index (9).html
index.html
Unit Tests
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
HTML
html
head
body
index.html
index.html
Source
History
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>Random 1 and -1 Numbers</title>
5 </head>
6 <body>
7 <script>
8 document.write("Randomly generated 1 and -1 integers
");
9 document.write("Two alternative methods:
");
10 document.write("(a): temp = (Math.round(Math.random()) * 2 - 1);
");
11 document.write("(b): temp = (Math.random() < 0.5 ? -1 : 1);
");
12 document.write("Both methods produce the required result
");
13 document.write("The variables (i) and (temp) have local scope

");
14 genRanNum();
15 function genRanNum(){
Random 1 and -1 Numbers
http://localhost:8383/JS_rand_no/index.html
100%
Randomly generated 1 and -1 integers
Two alternative methods:
(a): temp = (Math.round(Math.random()) * 2 - 1);
(b): temp = (Math.random() < 0.5 ? -1 : 1);
Both methods produce the required result
The variables (i) and (temp) have local scope
Method (a): 1, -1,
Method (b): -1 * -1 * 1 * -1 * -1 *
31:1
INS


JS_rand_no - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
147.8/451.0MB
Projects
Services
Files
index (6).html
index (7).html
index (8).html
index (9).html
index.html
Unit Tests
JS_Random
Site Root
index.html
Unit Tests
Navigator
JavaScript
HTML
html
head
body
index.html
index.html
Source
History
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>Random 1 and -1 Numbers</title>
5 </head>
6 <body>
7 <script>
8 document.write("Randomly generated 1 and -1 integers
");
9 document.write("Two alternative methods:
");
10 document.write("(a): temp = (Math.round(Math.random()) * 2 - 1);
");
11 document.write("(b): temp = (Math.random() < 0.5 ? -1 : 1);
");
12 document.write("Both methods produce the required result
");
13 document.write("The variables (i) and (temp) have local scope

");
14 genRanNum();
15 function genRanNum(){
16 var i, temp;
17 document.write("Method (a): ");
18 for(i = 0; i <= 1; i++) {
19 temp = (Math.round(Math.random()) * 2 - 1);
20 document.write(temp + ", ");
21 }
22 document.write("
Method (b): ");
23 for(i = 0; i < 5; i++) {
24 temp = (Math.random() < 0.5 ? -1 : 1);
25 document.write(temp + " * ");
26 }
27 }
28 </script>
29 </body>
30 </html>
31
Random 1 and -1 Numbers
31:1
INS

Multi-Dimensional Arrays

Multi-Dimensional Arrays

- Multi-dimensional arrays (MDA) are a very useful method of displaying information
- In high-level programming languages there is support for MDA
- In JavaScript has no inbuilt support for MDA
  - However: in JavaScript MDA can be created by emulating this behaviour
  - The method requires the population of arrays with arrays (recall that JavaScript arrays are untyped) to create a multi-level structure
- I have provides an additional supplementary resource detailing MD arrays

Review

- In this session we have introduced
  - Working with dates and times
  - The generation of random numbers including the nature of random numbers in JavaScript
  - Comparison, conditional, and logical operators with Boolean types
  - The `this` keyword (important place in JavaScript web programming as it is frequently used in functions and objects)
  - Commonly used JavaScript Math methods
  - An introduction to multi-dimensional arrays
- Worked examples have shown topics covered in this tutorial

Bring it all Together!

- In this course we have introduced:
- The basics of computer programming
- We have shown the basic functions of HTML and JavaScript
- In the next tutorials we will introduce the basics of
  - PHP with database technologies and SQL with a focus on MySQL
- However:
  - Knowing the functional elements and methods of these technologies is not enough
  - We must identify the requirements specification and combine the functional elements and methods of the technologies using good design principles to meet the requirements