INFO 151
Web Systems and Services

Week 6 (T2)

Dr Philip Moore
Dr Zhili Zhao

Overview

- In this tutorial we extend our introduction to arrays and will introduce:
  - Reversing arrays
  - Review expressions and operators
  - Comparison and conditional operators
  - Logical operators
  - Comparing different types
  - Boolean values
  - Ternary conditional operator
  - Events and event handling
  - Catching runtime errors

Reversing Arrays

Reversing Array Elements

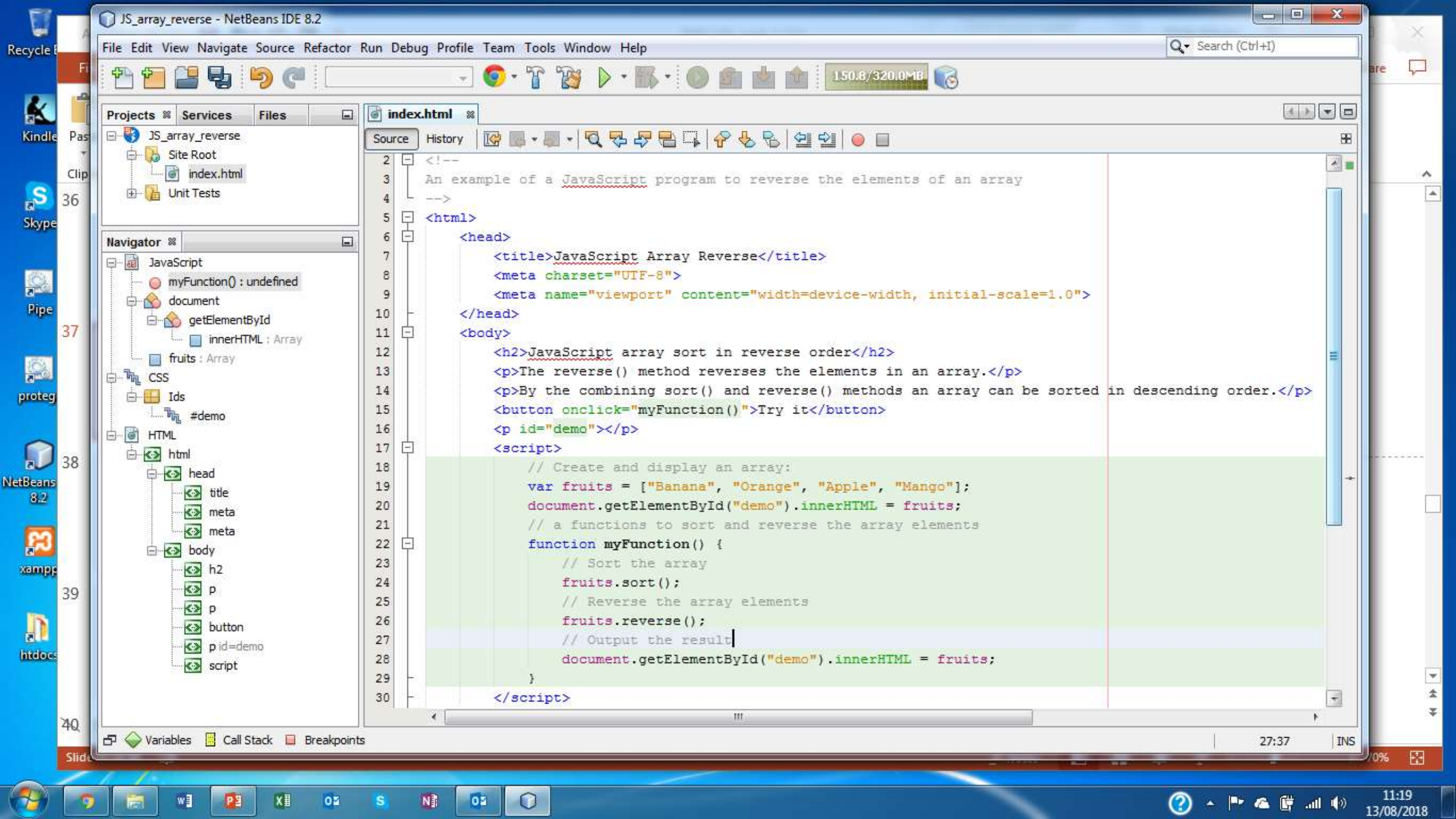
- The **`reverse()`** method reverses the elements in an array and can be used to sort an array in descending order
- For example:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

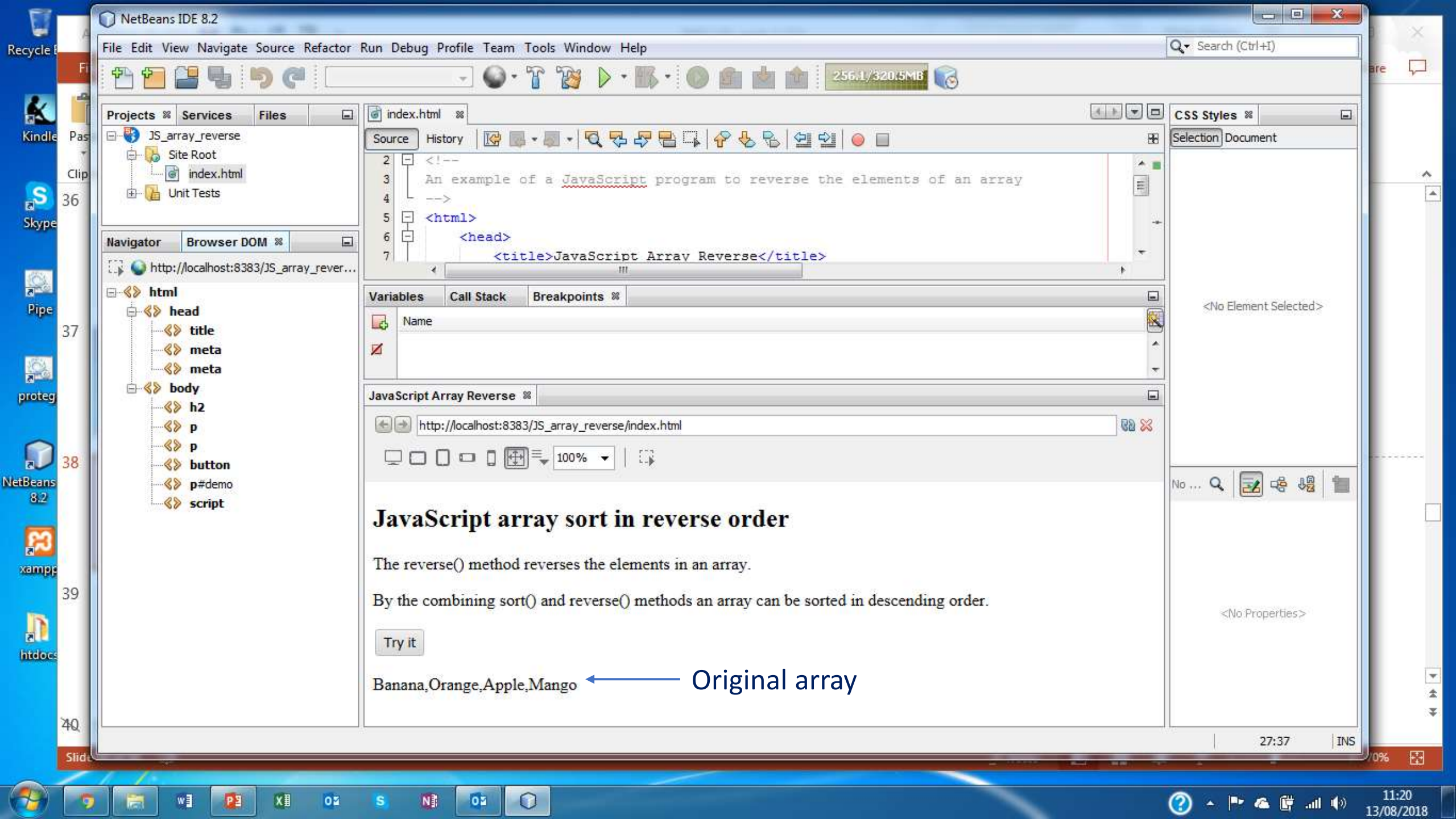
`fruits.sort()` sorts the elements of fruits array (ascending order)

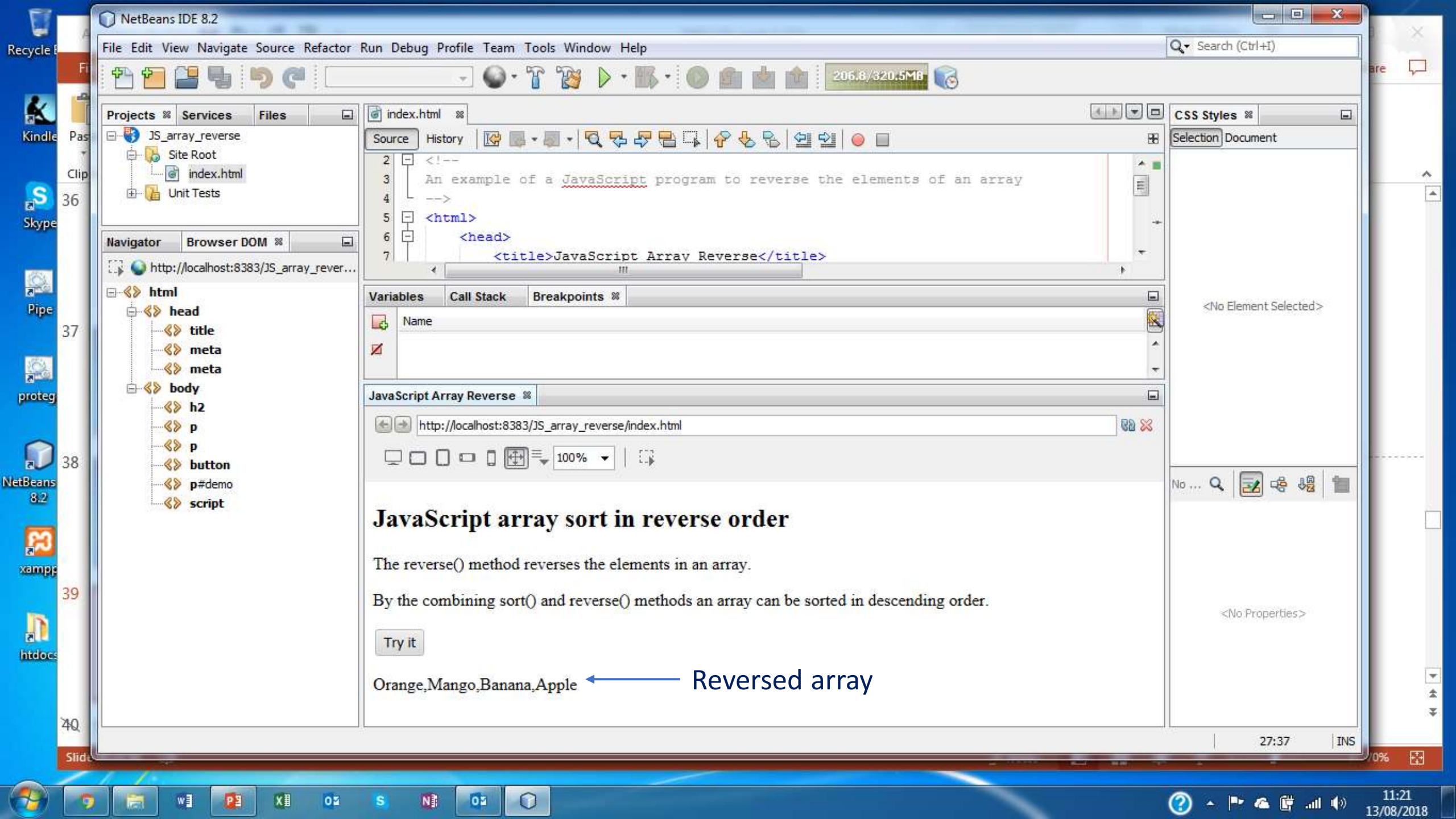
`fruits.reverse()` reverses the order of the elements (descending order)

- The following slides show worked examples of the process


JS_array_reverse - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
150.8/320.0MB
Projects
Services
Files
JS_array_reverse
Site Root
index.html
Unit Tests
Navigator
JavaScript
myFunction() : undefined
document
getElementById
innerHTML : Array
fruits : Array
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
p
button
p id=demo
script
index.html
Source
History
<!--
An example of a JavaScript program to reverse the elements of an array
-->
<html>
<head>
<title>JavaScript Array Reverse</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h2>JavaScript array sort in reverse order</h2>
The reverse() method reverses the elements in an array.
By the combining sort() and reverse() methods an array can be sorted in descending order.
<button onclick="myFunction()">Try it</button>

<script>
// Create and display an array:
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;
// a functions to sort and reverse the array elements
function myFunction() {
// Sort the array
fruits.sort();
// Reverse the array elements
fruits.reverse();
// Output the result
document.getElementById("demo").innerHTML = fruits;
}
</script>
Variables
Call Stack
Breakpoints
27:37
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
256.1/320.5MB
Projects Services Files
JS_array_reverse
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_array_rever...
html
head
title
meta
meta
body
h2
p
p
button
p#demo
script
index.html
Source History
2 <!--
3 An example of a JavaScript program to reverse the elements of an array
4 -->
5 <html>
6 <head>
7 <title>JavaScript Array Reverse</title>
Variables Call Stack Breakpoints
Name
JavaScript Array Reverse
http://localhost:8383/JS_array_reverse/index.html
100%
JavaScript array sort in reverse order
The reverse() method reverses the elements in an array.
By the combining sort() and reverse() methods an array can be sorted in descending order.
Try it
Banana,Orange,Apple,Mango
Original array
CSS Styles
Selection Document
<No Element Selected>
<No Properties>
27:37
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_array_reverse
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_array_rever...
html
head
title
meta
meta
body
h2
p
p
button
p#demo
script
index.html
Source History
2 <!--
3 An example of a JavaScript program to reverse the elements of an array
4 -->
5 <html>
6 <head>
7 <title>JavaScript Array Reverse</title>
Variables Call Stack Breakpoints
Name
JavaScript Array Reverse
http://localhost:8383/JS_array_reverse/index.html
100%
JavaScript array sort in reverse order
The reverse() method reverses the elements in an array.
By the combining sort() and reverse() methods an array can be sorted in descending order.
Try it
Orange,Mango,Banana,Apple
Reversed array
CSS Styles
Selection Document
<No Element Selected>
<No Properties>
27:37
INS

Review
Expressions and Operators

Operators

Operator	Syntax	Example	Definition
addition	+	x + y	Sum of x and y
subtraction	-	x - y	Difference of x and y
multiplication	*	x * y	Product of x and y
division	/	x / y	Quotient of x and y
modulo	%	x % y	Remainder of x / y
exponent	**	x ** y	x to the y power
increment	++	x++	x plus one
decrement	--	x--	x minus one

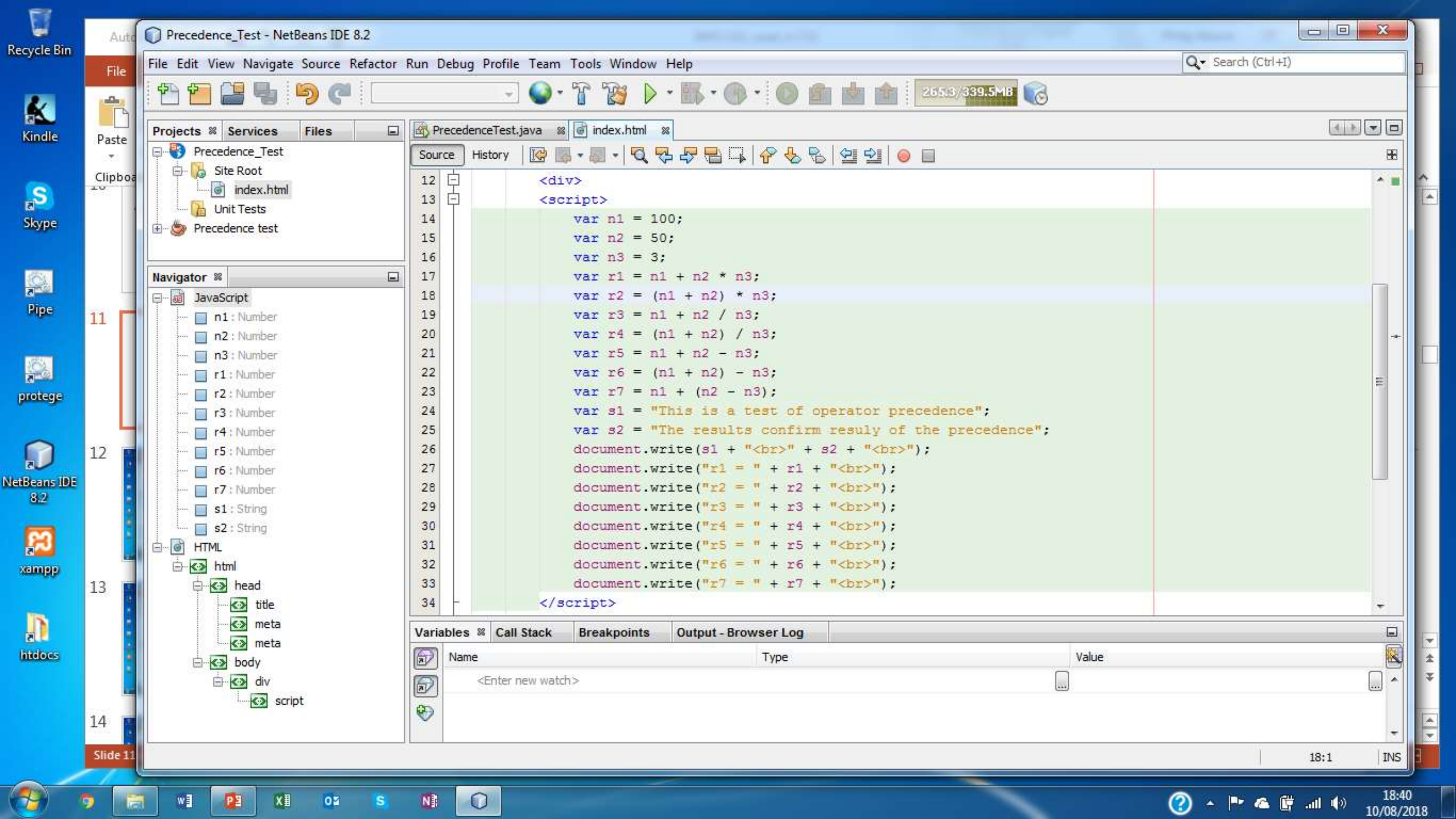
Expressions and Operators

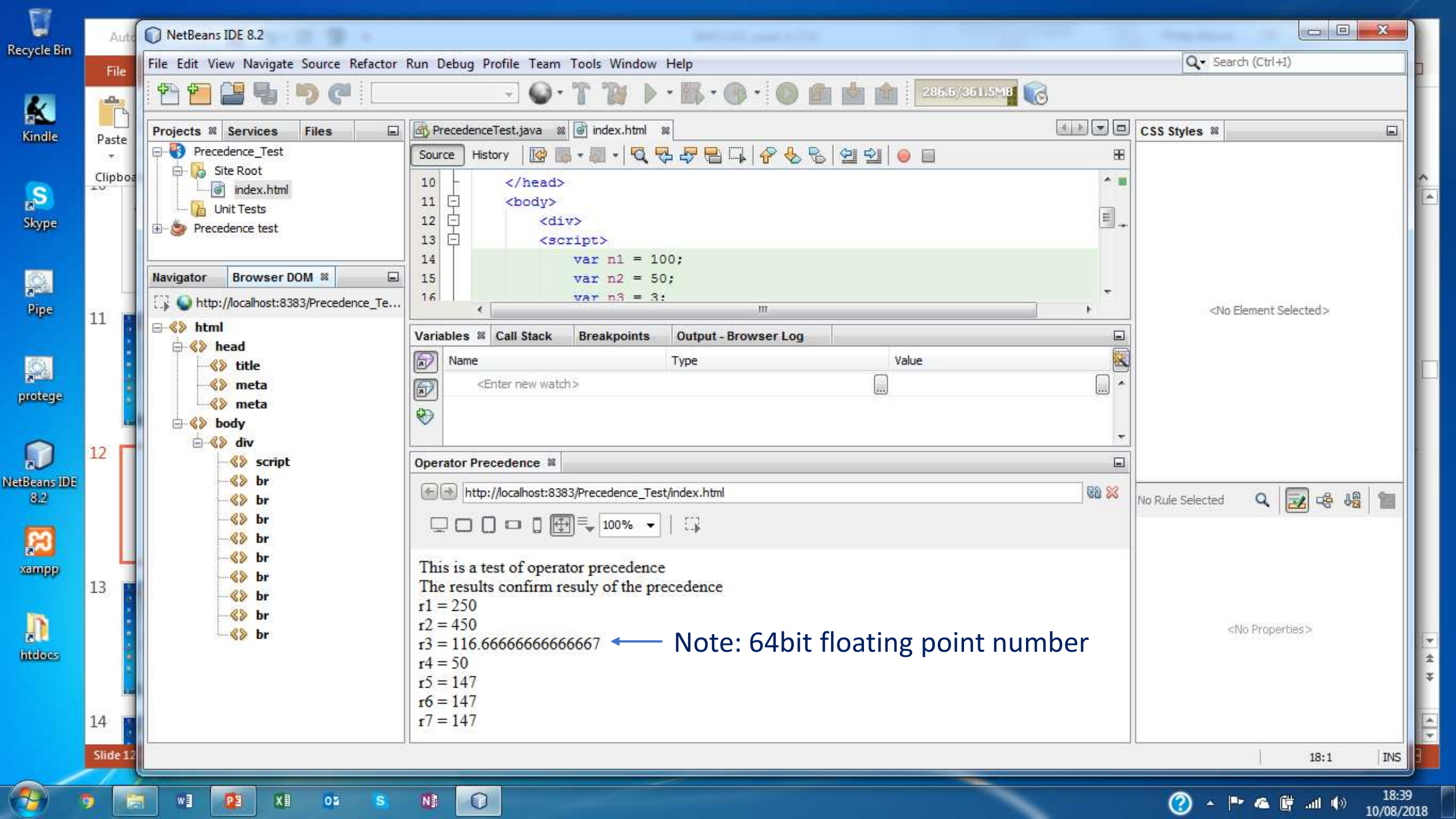
- A JavaScript expression is formed by combining values which can be any combination of the following:
  - Strings / string literals / variables / object properties / array elements / function invocations
  - Parenthesis **(** … **)** can be used to group sub-expressions
    - This can be used to change the default evaluation (processing) order
    - While the JavaScript syntax may be correct the program logic may be incorrect resulting in the wrong answer
- In JavaScript (in all programming languages) there is operator precedence

Operator Precedence Examples

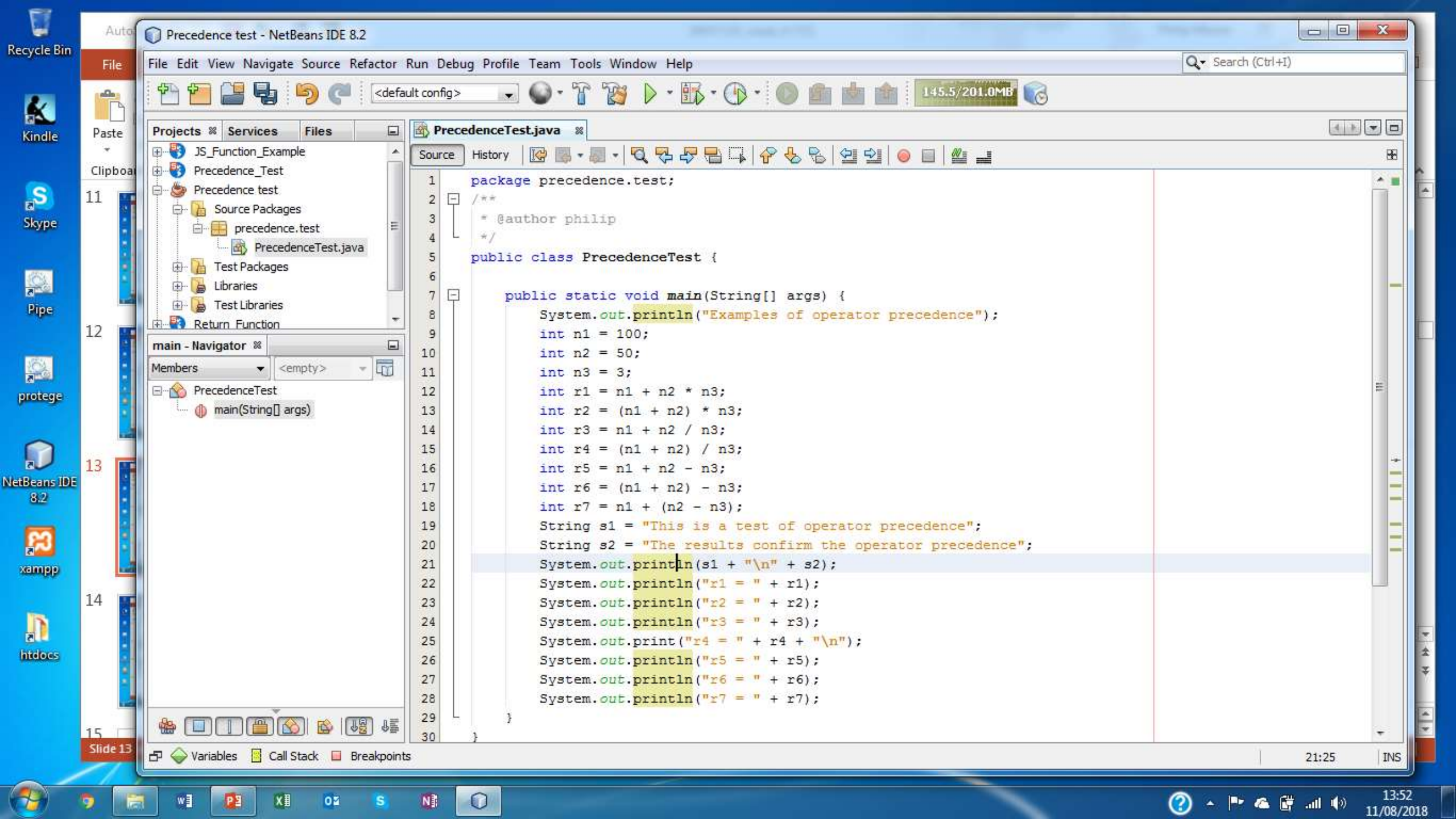
- Consider the following examples:
  - `100 + 50 * 3 = 250`
    - multiplication has precedence over addition
  - `(100 + 50) * 3 = 450`
    - multiplication has precedence over addition (but) parenthesis has precedence over multiplication
  - `100 + 50 / 3 = 116.66666666666667`
    - division has precedence over addition
  - `(100 + 50) / 3 = 50`
    - division has precedence over addition (but) parenthesis has precedence over division
  - `100 + 50 - 3 = 147`

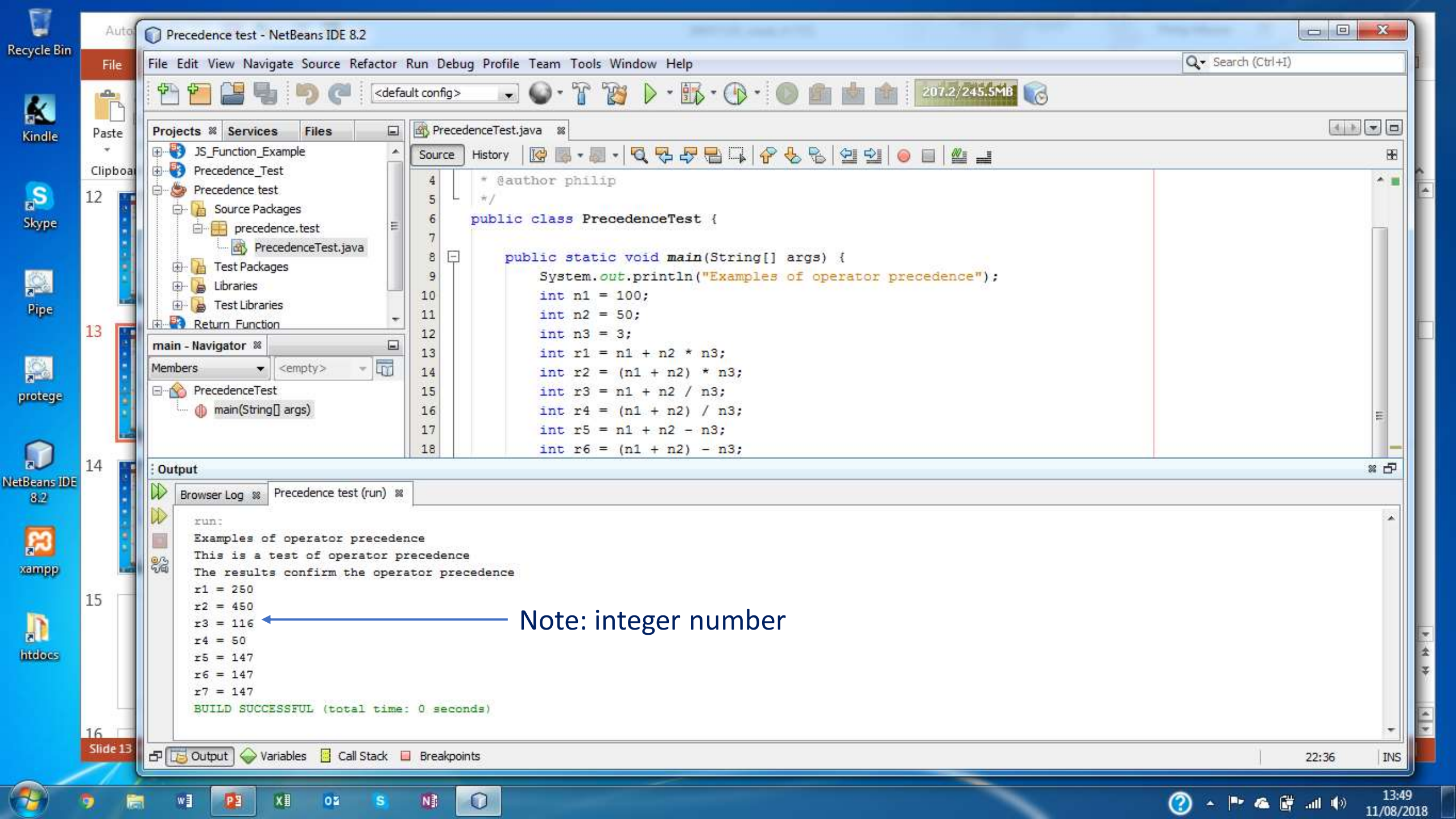
A JavaScript Implementation


Precedence_Test - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
265.3/339.5MB
Projects
Services
Files
Precedence_Test
Site Root
index.html
Unit Tests
Precedence test
Navigator
JavaScript
n1 : Number
n2 : Number
n3 : Number
r1 : Number
r2 : Number
r3 : Number
r4 : Number
r5 : Number
r6 : Number
r7 : Number
s1 : String
s2 : String
HTML
html
head
title
meta
meta
body
div
script
PrecedenceTest.java
index.html
Source
History
12
13 <script>
14 var n1 = 100;
15 var n2 = 50;
16 var n3 = 3;
17 var r1 = n1 + n2 * n3;
18 var r2 = (n1 + n2) * n3;
19 var r3 = n1 + n2 / n3;
20 var r4 = (n1 + n2) / n3;
21 var r5 = n1 + n2 - n3;
22 var r6 = (n1 + n2) - n3;
23 var r7 = n1 + (n2 - n3);
24 var s1 = "This is a test of operator precedence";
25 var s2 = "The results confirm resuly of the precedence";
26 document.write(s1 + "
" + s2 + "
");
27 document.write("r1 = " + r1 + "
");
28 document.write("r2 = " + r2 + "
");
29 document.write("r3 = " + r3 + "
");
30 document.write("r4 = " + r4 + "
");
31 document.write("r5 = " + r5 + "
");
32 document.write("r6 = " + r6 + "
");
33 document.write("r7 = " + r7 + "
");
34 </script>
Variables
Call Stack
Breakpoints
Output - Browser Log
Name
Type
Value
<Enter new watch>
18:1
INS
Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
Paste
Clipboa
File
Slide 11
18:40
10/08/2018


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
Precedence_Test
Site Root
index.html
Unit Tests
Precedence test
Navigator Browser DOM
http://localhost:8383/Precedence_Te...
html
head
title
meta
meta
body
div
script
br
br
br
br
br
br
br
br
br
PrecedenceTest.java
index.html
Source History
10 </head>
11 <body>
12
13 <script>
14 var n1 = 100;
15 var n2 = 50;
16 var n3 = 3;
Variables Call Stack Breakpoints Output - Browser Log
Name Type Value
<Enter new watch>
Operator Precedence
http://localhost:8383/Precedence_Test/index.html
100%
This is a test of operator precedence
The results confirm resuly of the precedence
r1 = 250
r2 = 450
r3 = 116.66666666666667
r4 = 50
r5 = 147
r6 = 147
r7 = 147
Note: 64bit floating point number
CSS Styles
<No Element Selected>
No Rule Selected
<No Properties>
18:1
INS
Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
Paste
File
Slide 12
18:39
10/08/2018

A Java Implementation


Precedence test - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
145.5/201.0MB
Projects
Services
Files
JS_Function_Example
Precedence_Test
Precedence test
Source Packages
precedence.test
PrecedenceTest.java
Test Packages
Libraries
Test Libraries
Return_Function
main - Navigator
Members
<empty>
PrecedenceTest
main(String[] args)
PrecedenceTest.java
Source
History
1 package precedence.test;
2 /**
3 * @author philip
4 */
5 public class PrecedenceTest {
6
7 public static void main(String[] args) {
8 System.out.println("Examples of operator precedence");
9 int n1 = 100;
10 int n2 = 50;
11 int n3 = 3;
12 int r1 = n1 + n2 * n3;
13 int r2 = (n1 + n2) * n3;
14 int r3 = n1 + n2 / n3;
15 int r4 = (n1 + n2) / n3;
16 int r5 = n1 + n2 - n3;
17 int r6 = (n1 + n2) - n3;
18 int r7 = n1 + (n2 - n3);
19 String s1 = "This is a test of operator precedence";
20 String s2 = "The results confirm the operator precedence";
21 System.out.println(s1 + "\n" + s2);
22 System.out.println("r1 = " + r1);
23 System.out.println("r2 = " + r2);
24 System.out.println("r3 = " + r3);
25 System.out.print("r4 = " + r4 + "\n");
26 System.out.println("r5 = " + r5);
27 System.out.println("r6 = " + r6);
28 System.out.println("r7 = " + r7);
29 }
30 }
Variables
Call Stack
Breakpoints
21:25
INS
Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
Paste
Clipboard
Slide 13
13:52
11/08/2018


Precedence test - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
207.2/245.5MB
Projects
Services
Files
JS_Function_Example
Precedence_Test
Precedence test
Source Packages
precedence.test
PrecedenceTest.java
Test Packages
Libraries
Test Libraries
Return_Function
main - Navigator
Members
<empty>
PrecedenceTest
main(String[] args)
PrecedenceTest.java
Source
History
4 * @author philip
5 */
6 public class PrecedenceTest {
7
8 public static void main(String[] args) {
9 System.out.println("Examples of operator precedence");
10 int n1 = 100;
11 int n2 = 50;
12 int n3 = 3;
13 int r1 = n1 + n2 * n3;
14 int r2 = (n1 + n2) * n3;
15 int r3 = n1 + n2 / n3;
16 int r4 = (n1 + n2) / n3;
17 int r5 = n1 + n2 - n3;
18 int r6 = (n1 + n2) - n3;
Output
Browser Log
Precedence test (run)
run:
Examples of operator precedence
This is a test of operator precedence
The results confirm the operator precedence
r1 = 250
r2 = 450
r3 = 116
r4 = 50
r5 = 147
r6 = 147
r7 = 147
BUILD SUCCESSFUL (total time: 0 seconds)
Output
Variables
Call Stack
Breakpoints
22:36
INS
Note: integer number
Slide 13
13:49
11/08/2018

Comparison and Conditional Operators

Comparison Operators

- Comparisons and Conditions
  - A full overview of comparison operators and conditional statements may be found in the course resources
- *Comparison* operators relate to
  - Comparison and Logical operators are used to test for *true* or *false*
- *Conditional* operators refer to
  - Conditional statements are used to perform different actions based on different conditions
- The following slides show worked examples

Comparison Operators (1)

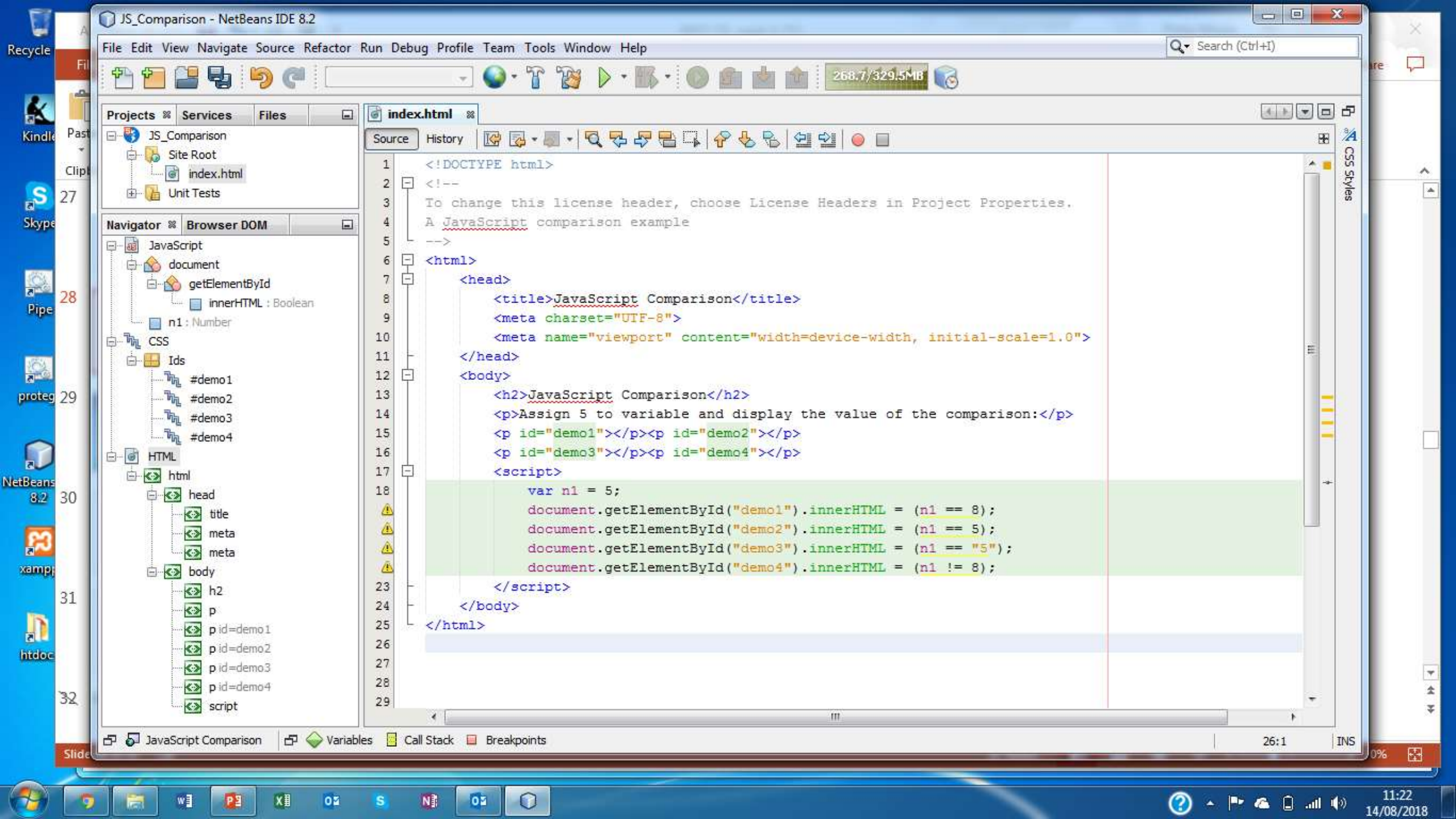
- Comparison operators are used in logical statements to determine equality or difference between variables or values
  - The table below shows comparison operators

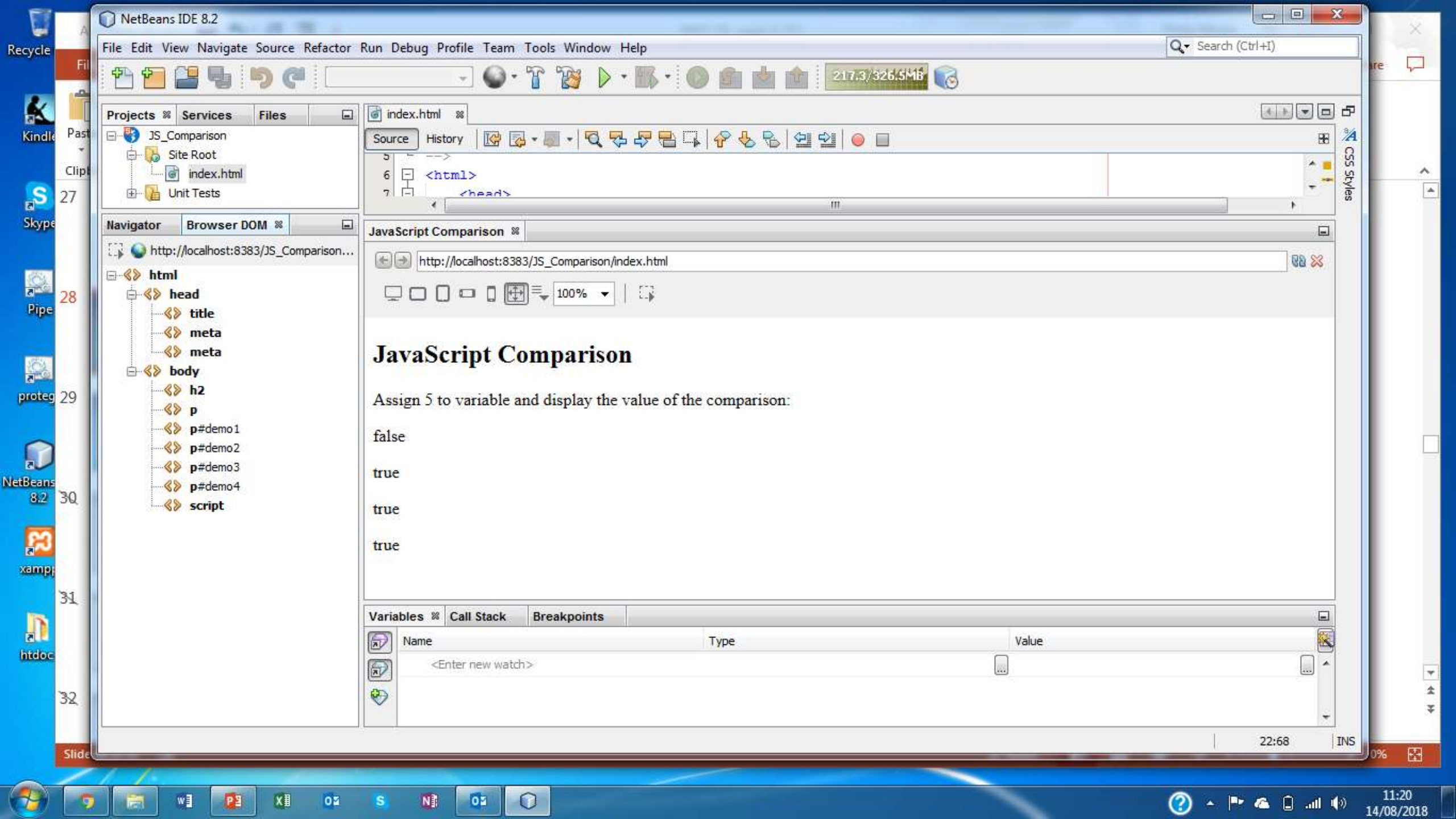
Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 8	false
		x === 5	true
		x === "5"	true

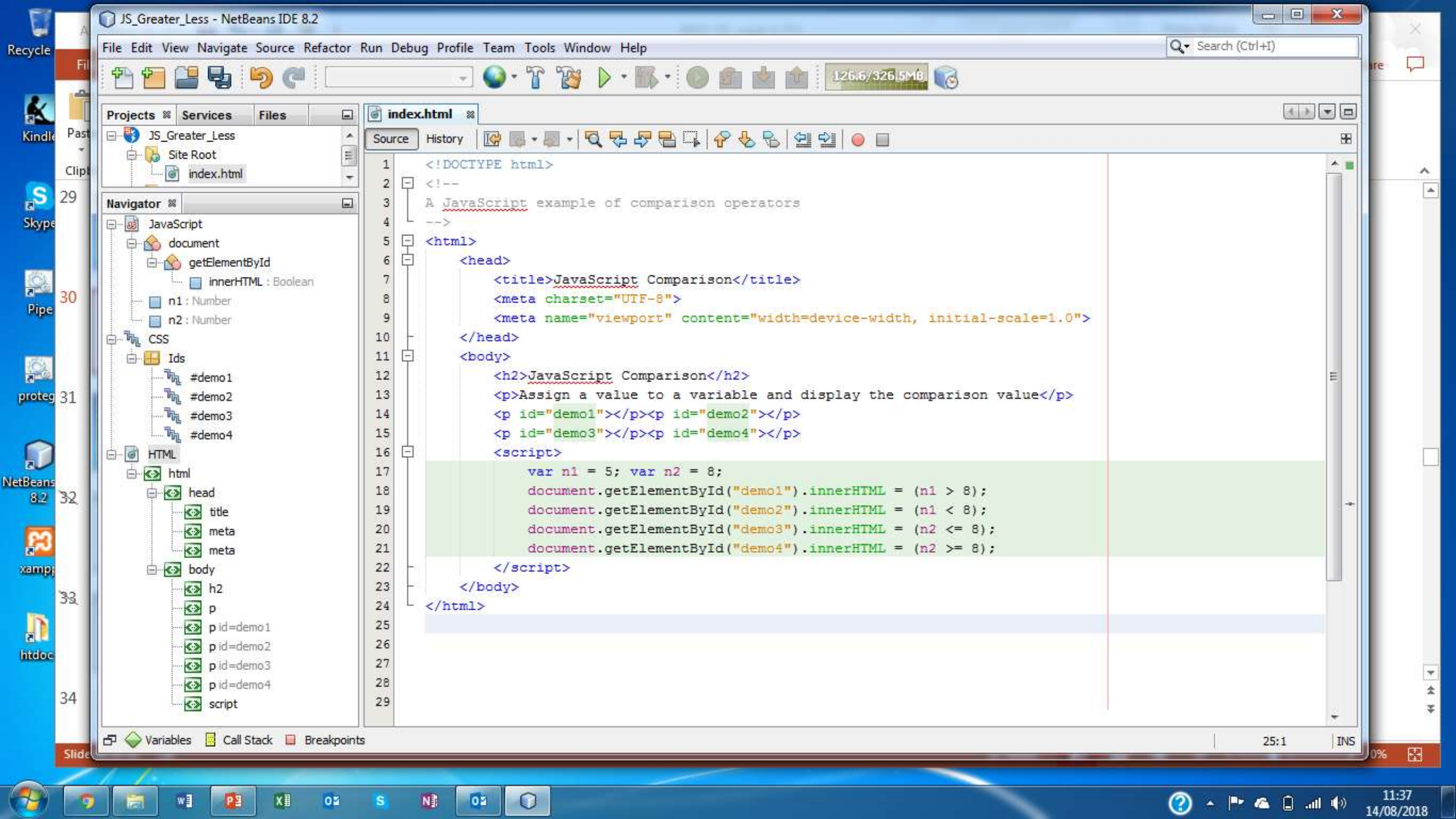
Conditional Operators (2)

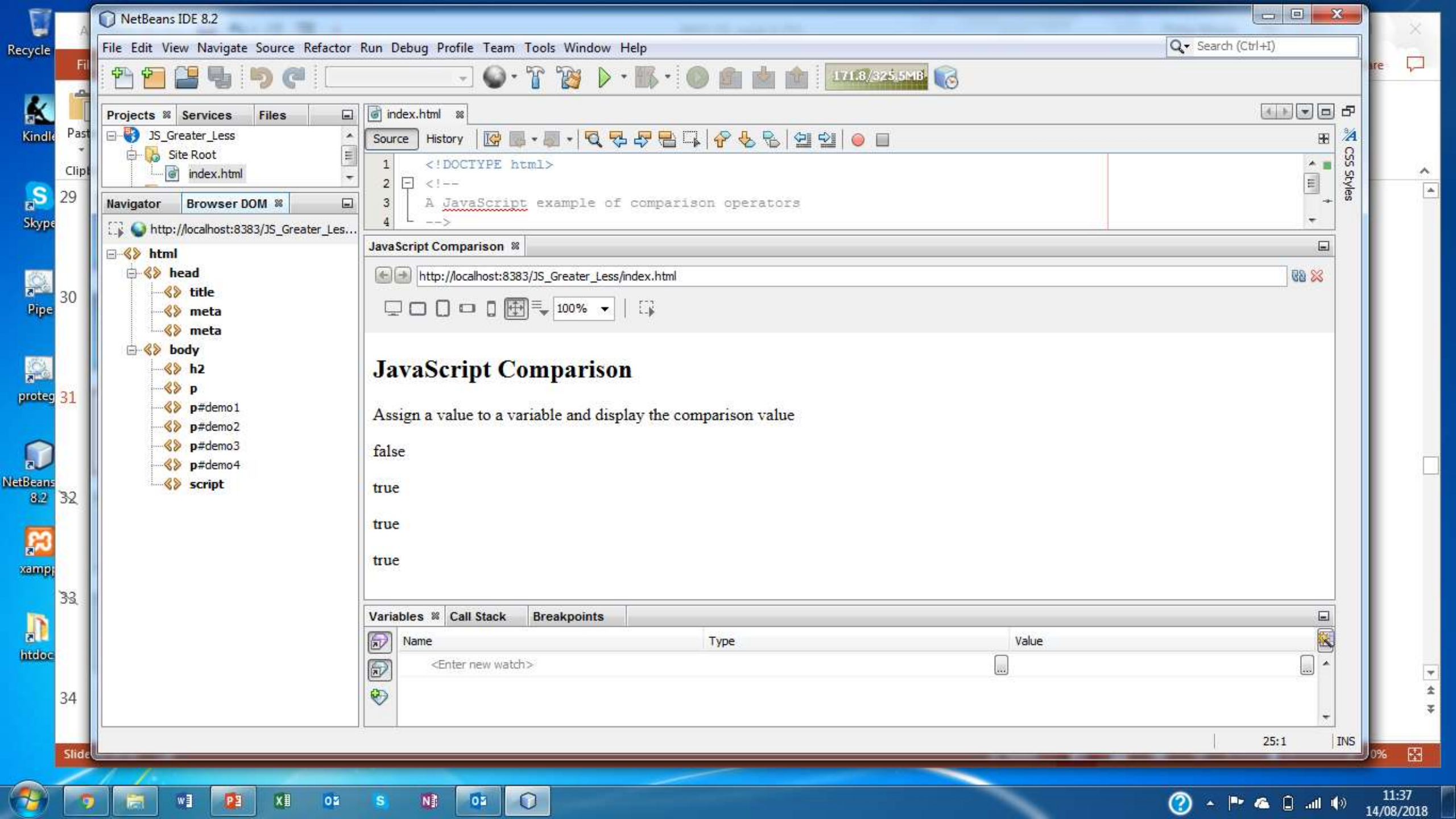
- Comparisons
  - The range of JavaScript comparison and conditional operators many be found in the course resources
  - Below we show conditional operators

Operator	Description	Example
==	equal to	if (day == "Monday")
>	greater than	if (salary > 9000)
<	less than	if (age < 18)
>=	greater than or equal to	if(age >= 18)
<=	less than or equal to	if(age <=18)


JS_Comparison - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
268.7/329.5MB
Projects
Services
Files
JS_Comparison
Site Root
index.html
Unit Tests
Navigator
Browser DOM
JavaScript
document
getElementById
innerHTML : Boolean
n1 : Number
CSS
Ids
#demo1
#demo2
#demo3
#demo4
HTML
html
head
title
meta
meta
body
h2
p
p id=demo1
p id=demo2
p id=demo3
p id=demo4
script
index.html
Source
History
CSS Styles
1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 A JavaScript comparison example
5 -->
6 <html>
7 <head>
8 <title>JavaScript Comparison</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 Comparison</h2>
14 Assign 5 to variable and display the value of the comparison:
15
16
17 <script>
18 var n1 = 5;
document.getElementById("demo1").innerHTML = (n1 == 8);
document.getElementById("demo2").innerHTML = (n1 == 5);
document.getElementById("demo3").innerHTML = (n1 == "5");
document.getElementById("demo4").innerHTML = (n1 != 8);
23 </script>
24 </body>
25 </html>
26
27
28
29
JavaScript Comparison
Variables
Call Stack
Breakpoints
26:1
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_Comparison
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_Comparison...
html
head
title
meta
meta
body
h2
p
p#demo1
p#demo2
p#demo3
p#demo4
script
index.html
Source History
<html>
JavaScript Comparison
http://localhost:8383/JS_Comparison/index.html
100%
JavaScript Comparison
Assign 5 to variable and display the value of the comparison:
false
true
true
true
Variables Call Stack Breakpoints
Name Type Value
<Enter new watch>
22:68
INS
CSS Styles


JS_Greater_Less - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
126.6/326.5MB
Projects
Services
Files
JS_Greater_Less
Site Root
index.html
Navigator
JavaScript
document
getElementById
innerHTML : Boolean
n1 : Number
n2 : Number
CSS
Ids
#demo1
#demo2
#demo3
#demo4
HTML
html
head
title
meta
meta
body
h2
p
p id=demo1
p id=demo2
p id=demo3
p id=demo4
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of comparison operators
4 -->
5 <html>
6 <head>
7 <title>JavaScript Comparison</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 Comparison</h2>
13 Assign a value to a variable and display the comparison value
14
15
16 <script>
17 var n1 = 5; var n2 = 8;
18 document.getElementById("demo1").innerHTML = (n1 > 8);
19 document.getElementById("demo2").innerHTML = (n1 < 8);
20 document.getElementById("demo3").innerHTML = (n2 <= 8);
21 document.getElementById("demo4").innerHTML = (n2 >= 8);
22 </script>
23 </body>
24 </html>
25
26
27
28
29
Variables
Call Stack
Breakpoints
25:1
INS

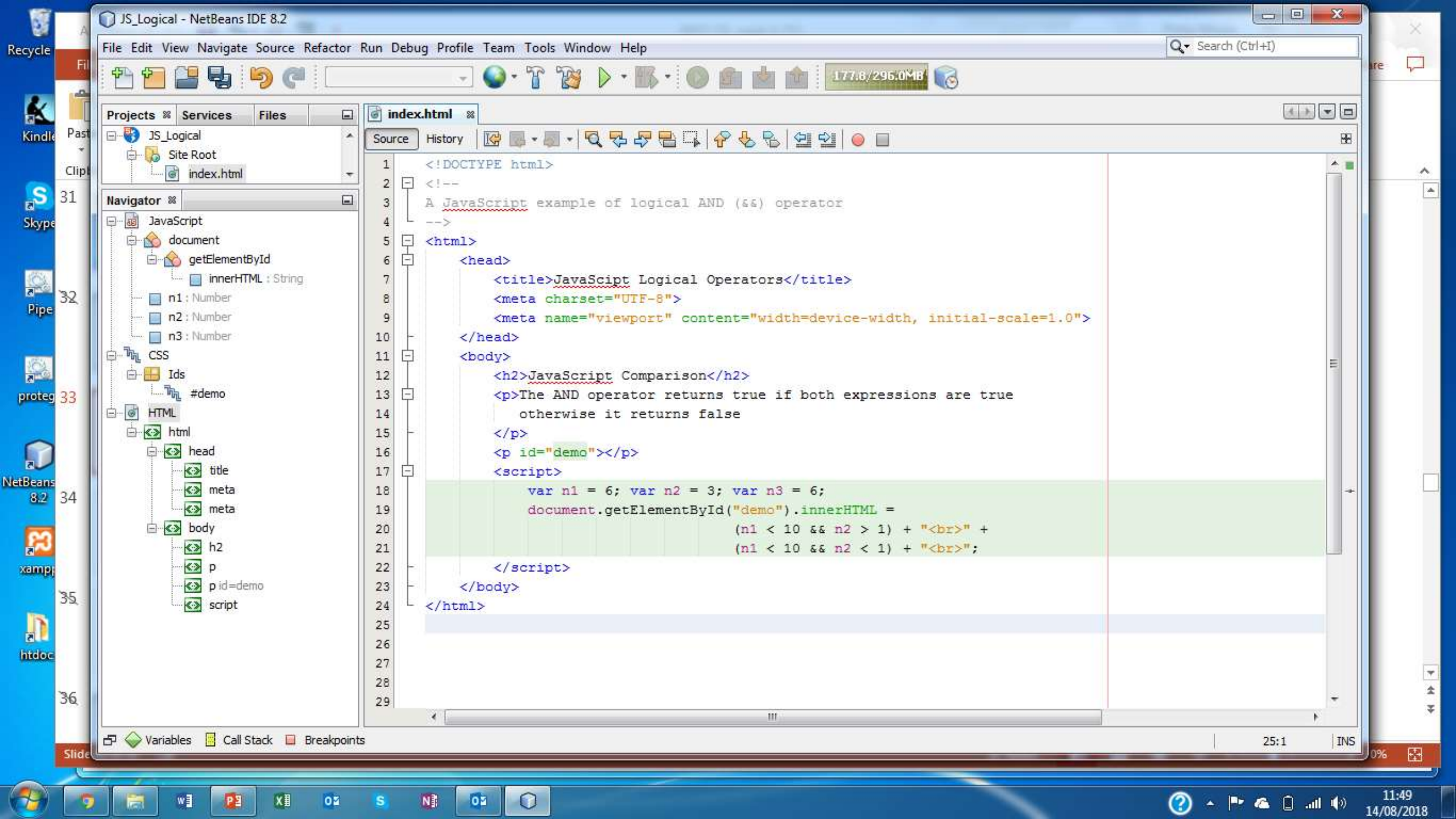

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
171.8/325.5MB
Projects
Services
Files
JS_Greater_Less
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_Greater_Les...
html
head
title
meta
meta
body
h2
p
p#demo1
p#demo2
p#demo3
p#demo4
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of comparison operators
4 -->
CSS Styles
JavaScript Comparison
http://localhost:8383/JS_Greater_Less/index.html
100%
JavaScript Comparison
Assign a value to a variable and display the comparison value
false
true
true
true
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
25:1
INS

Logical Operators

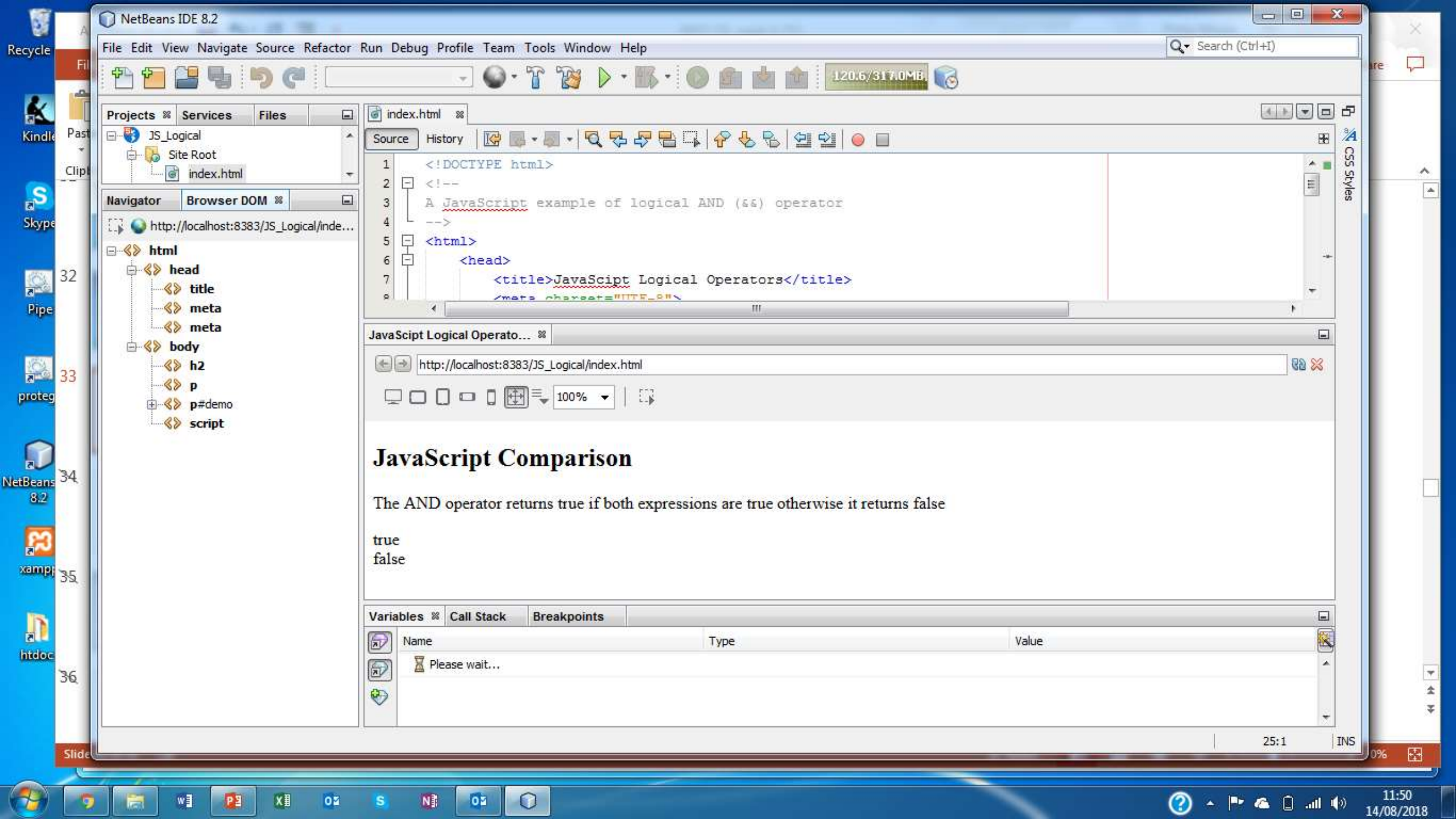
Logical Operators

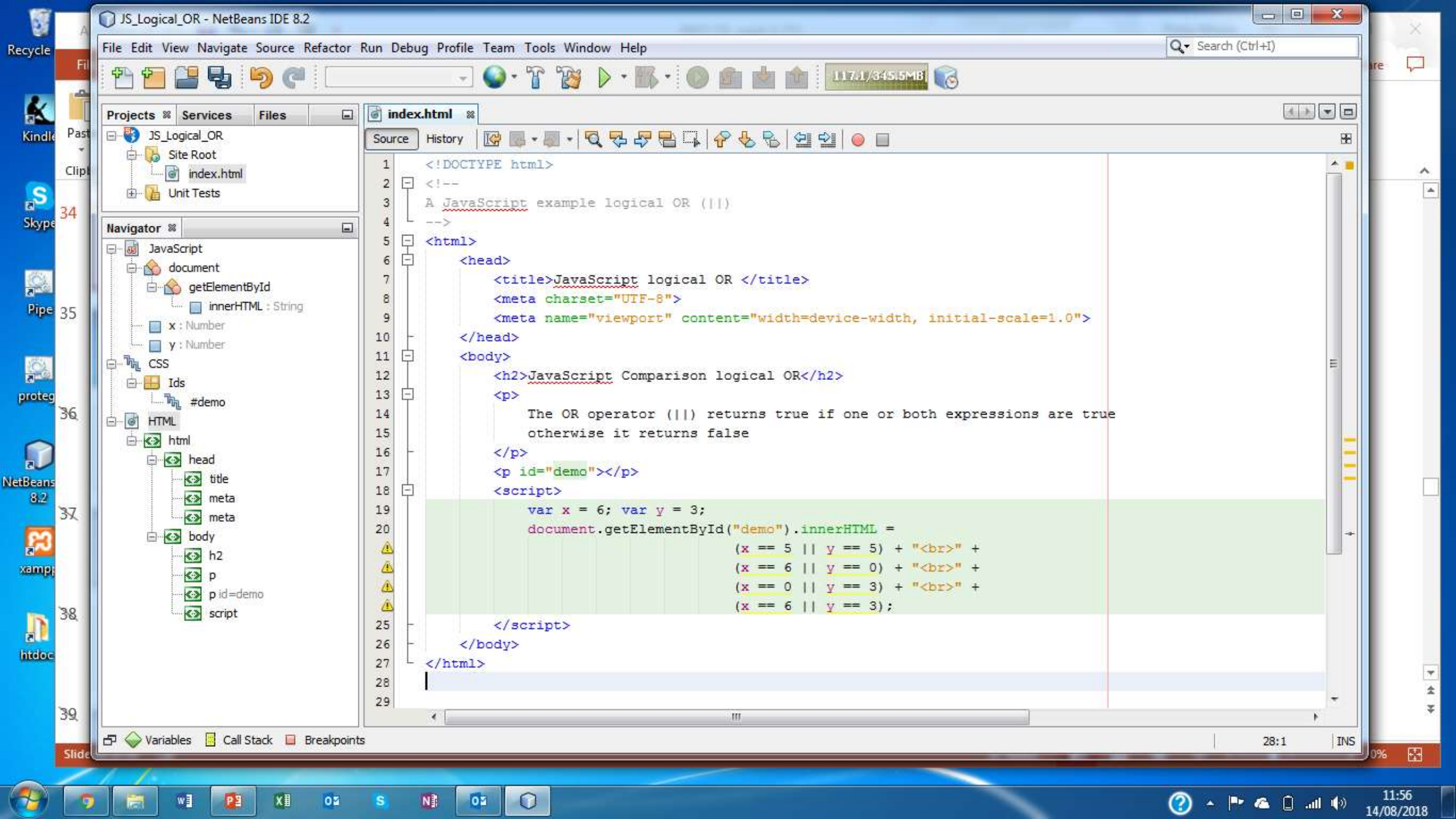
- Logical operators are used to determine the logic between variables or values
- The table below explains the logical operators

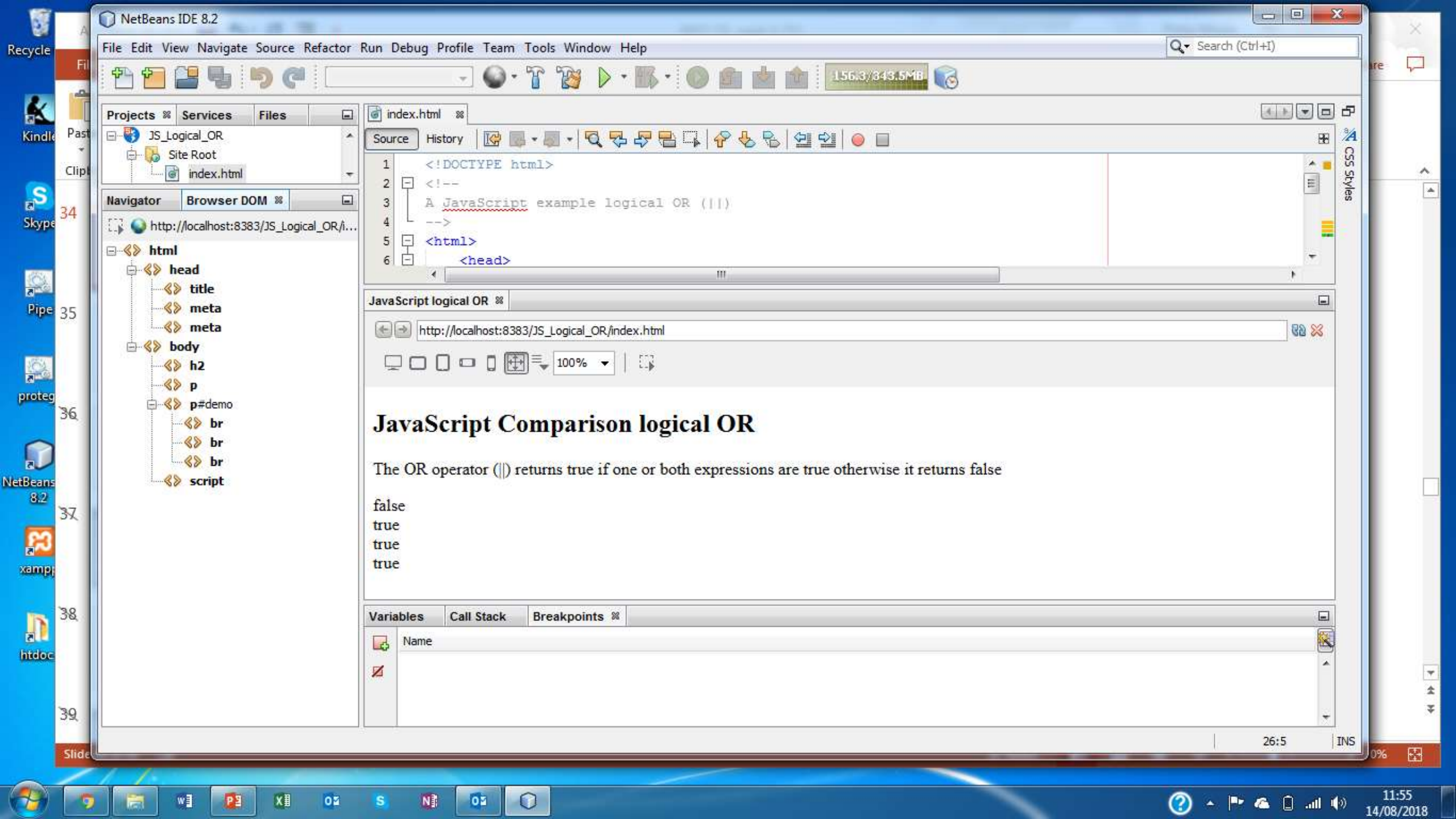
Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x == 5 || y == 5) is false
!	not	!(x == y) is true


JS_Logical - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
177.8/296.0MB
Projects
Services
Files
JS_Logical
Site Root
index.html
Navigator
JavaScript
document
getElementById
innerHTML : String
n1 : Number
n2 : Number
n3 : Number
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
h2
p
p id=demo
script
index.html
Source
History
<!DOCTYPE html>
<!--
A JavaScript example of logical AND (&&) operator
-->
<html>
<head>
<title>JavaScipt Logical Operators</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h2>JavaScript Comparison</h2>
The AND operator returns true if both expressions are true
otherwise it returns false

<script>
var n1 = 6; var n2 = 3; var n3 = 6;
document.getElementById("demo").innerHTML =
(n1 < 10 && n2 > 1) + "
" +
(n1 < 10 && n2 < 1) + "
";
</script>
</body>
</html>
Variables
Call Stack
Breakpoints
25:1
INS
11:49
14/08/2018


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
120.6/317.0MB
Projects
Services
Files
JS_Logical
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_Logical/inde...
html
head
title
meta
meta
body
h2
p
p#demo
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of logical AND (&&) operator
4 -->
5 <html>
6 <head>
7 <title>JavaScipt Logical Operators</title>
CSS Styles
JavaScipt Logical Operato...
http://localhost:8383/JS_Logical/index.html
100%
JavaScript Comparison
The AND operator returns true if both expressions are true otherwise it returns false
true
false
Variables
Call Stack
Breakpoints
Name
Type
Value
Please wait...
25:1
INS
11:50
14/08/2018


JS_Logical_OR - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
117.1/345.5MB
Projects
Services
Files
JS_Logical_OR
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : String
x : Number
y : Number
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 example logical OR (||)
4 -->
5 <html>
6 <head>
7 <title>JavaScript logical OR </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 Comparison logical OR</h2>
13
14 The OR operator (||) returns true if one or both expressions are true
15 otherwise it returns false
16
17
18 <script>
19 var x = 6; var y = 3;
20 document.getElementById("demo").innerHTML =
(x == 5 || y == 5) + "
" +
(x == 6 || y == 0) + "
" +
(x == 0 || y == 3) + "
" +
(x == 6 || y == 3);
25 </script>
26 </body>
27 </html>
28
29
Variables
Call Stack
Breakpoints
28:1
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_Logical_OR
Site Root
index.html
Navigator Browser DOM
http://localhost:8383/JS_Logical_OR/i...
html
head
title
meta
meta
body
h2
p
p#demo
br
br
br
script
index.html
Source History
<!DOCTYPE html>
<!--
A JavaScript example logical OR (||)
-->
<html>
<head>
CSS Styles
JavaScript logical OR
http://localhost:8383/JS_Logical_OR/index.html
100%
JavaScript Comparison logical OR
The OR operator (||) returns true if one or both expressions are true otherwise it returns false
false
true
true
true
Variables Call Stack Breakpoints
Name
26:5
INS

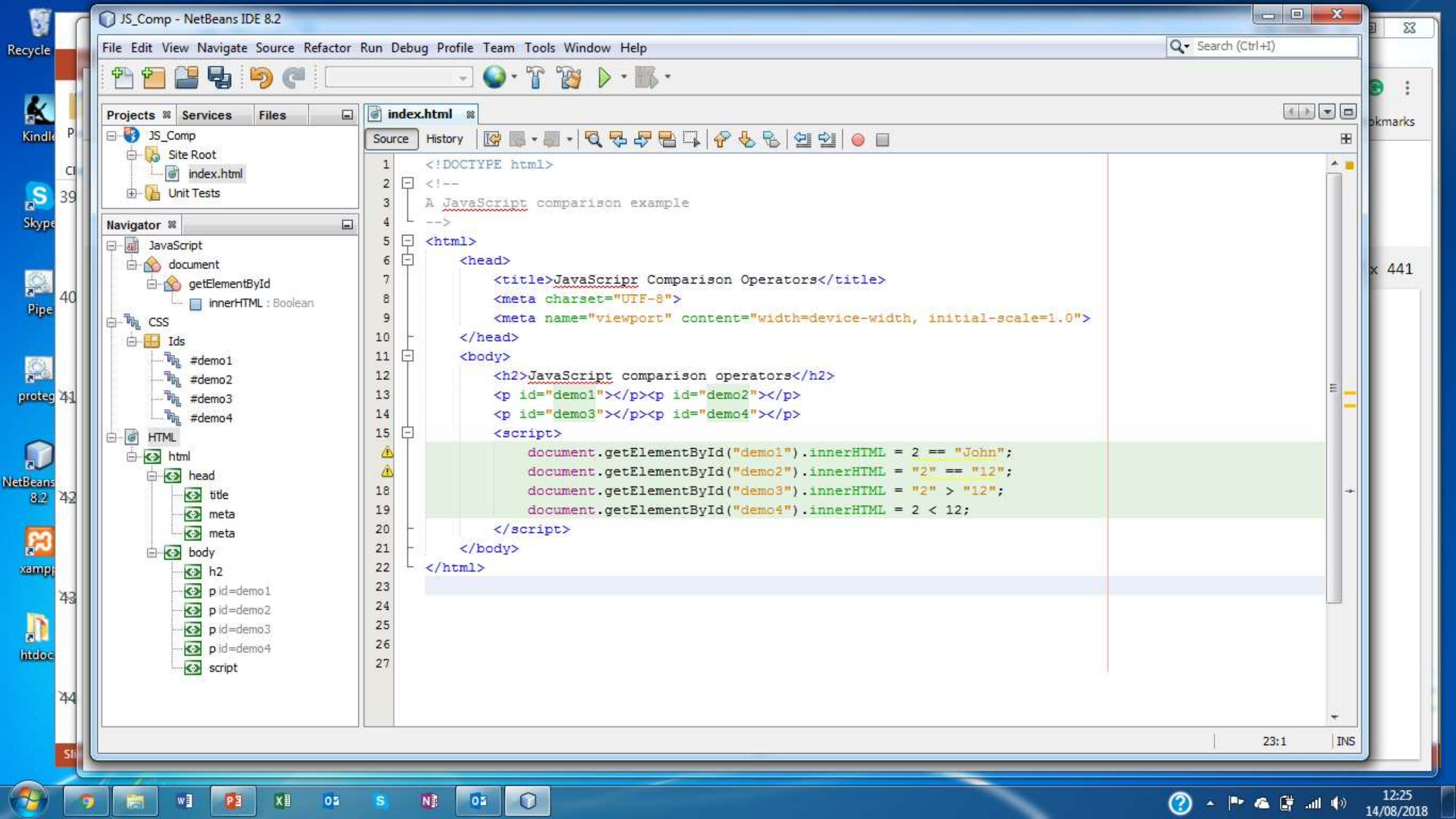
Comparing Different Types

Comparing Different Types

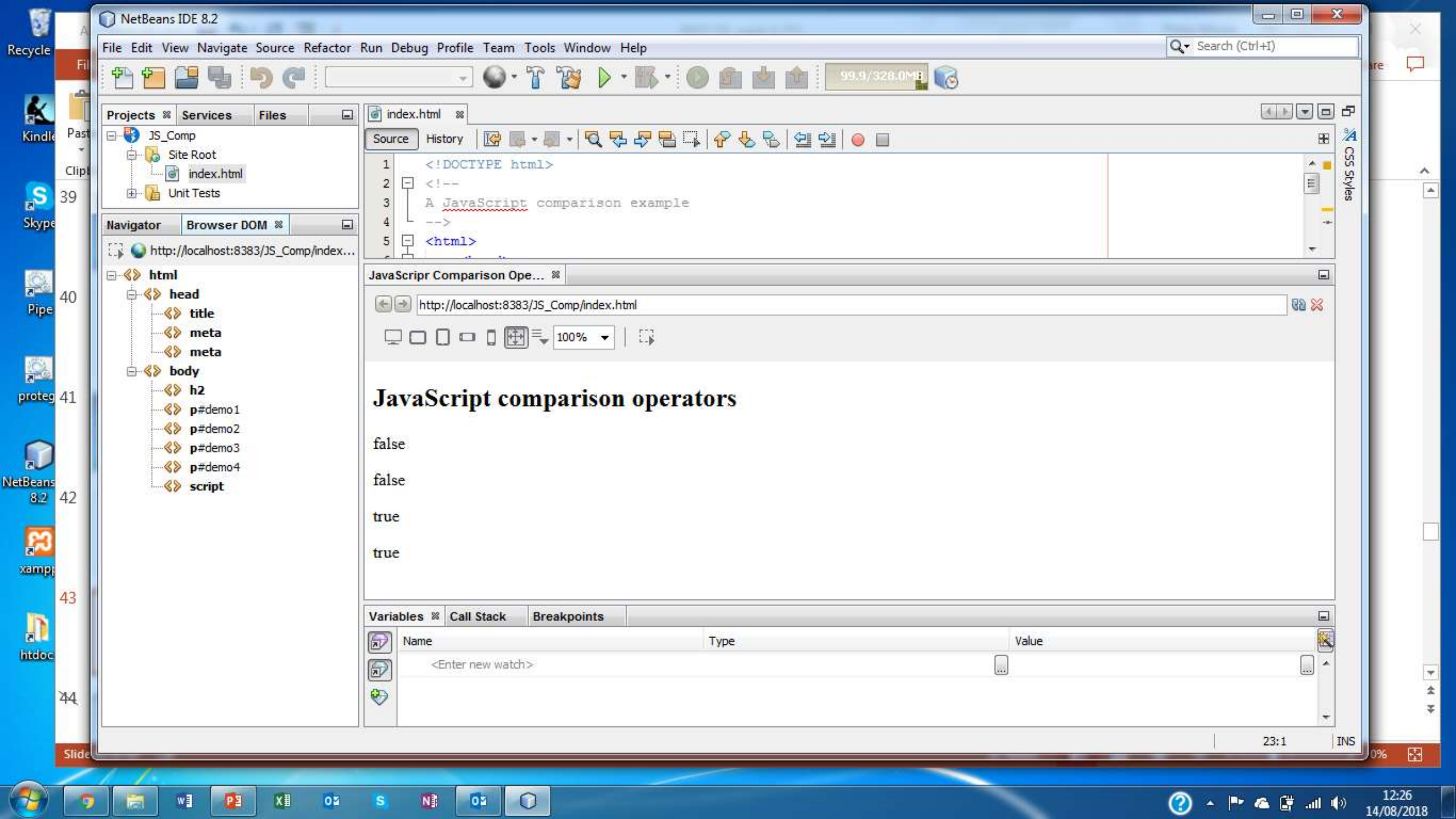
- Comparing data of different types may give unexpected results.
- When comparing a string with a number, JavaScript will convert the string to a number when doing the comparison.
- An empty string converts to 0 – a non-numeric string converts to **NaN** which is always false.
- For example

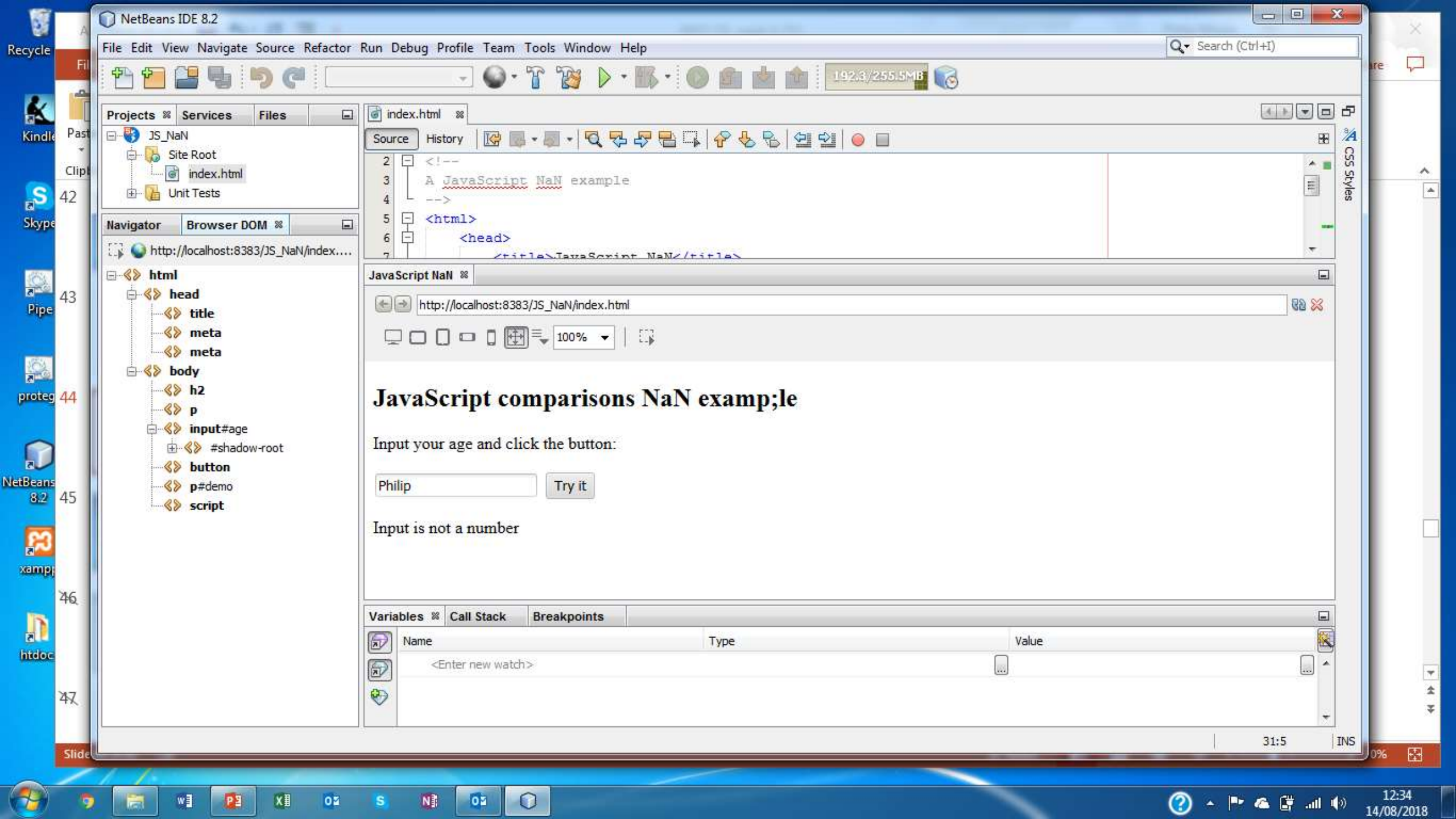
```
2 < 12 = true
“2” > “12”  = true
```

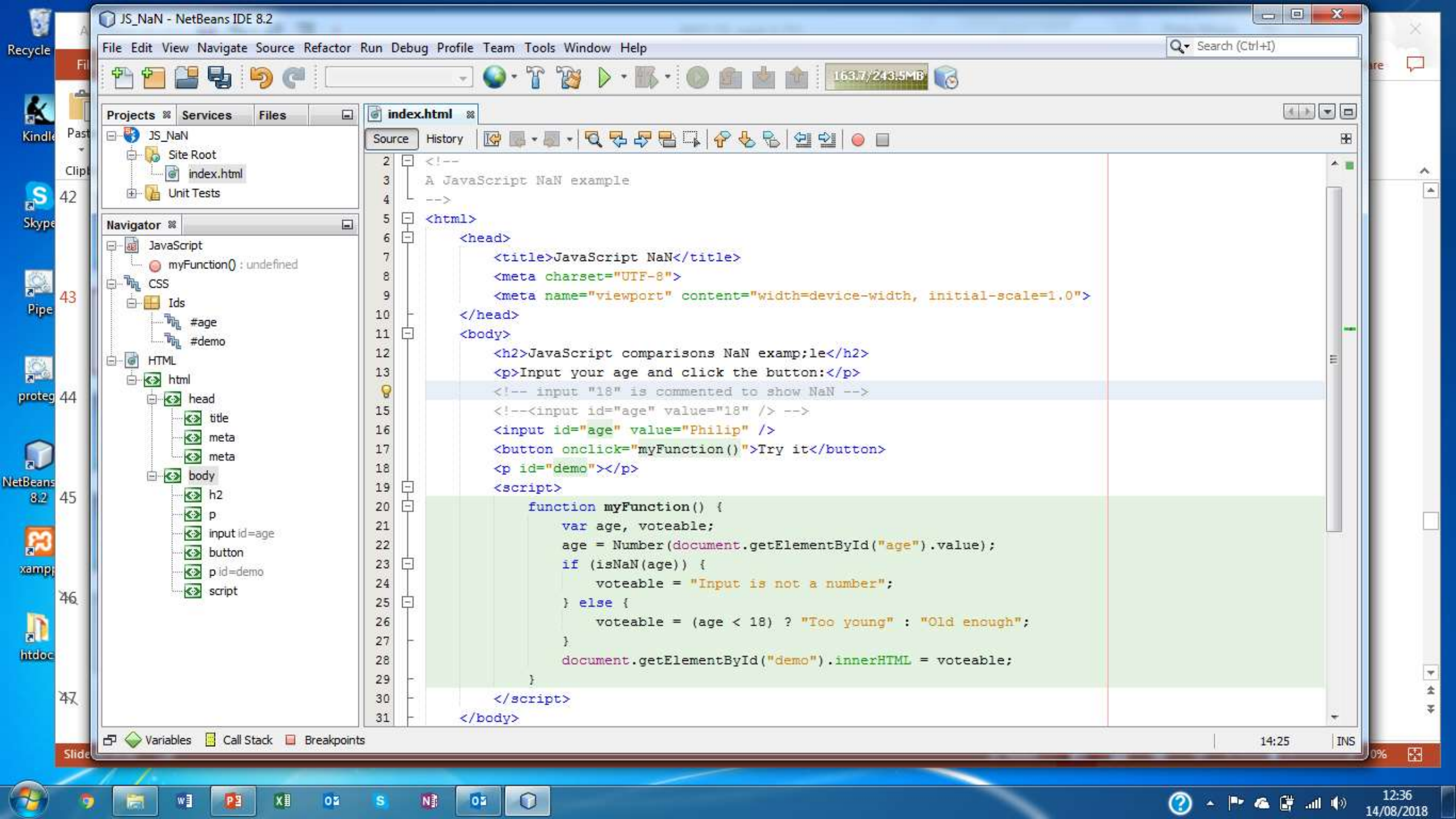
- When comparing two strings, "2" will be greater than "12" because (alphabetically) 1 is less than 2
- To secure a correct result variables should be converted to the proper type before comparison


JS_Comp - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Comp
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML : Boolean
CSS
Ids
#demo1
#demo2
#demo3
#demo4
HTML
html
head
title
meta
meta
body
h2
p id=demo1
p id=demo2
p id=demo3
p id=demo4
script
index.html
Source History
<!DOCTYPE html>
<!--
A JavaScript comparison example
-->
<html>
<head>
<title>JavaScripr Comparison Operators</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h2>JavaScript comparison operators</h2>

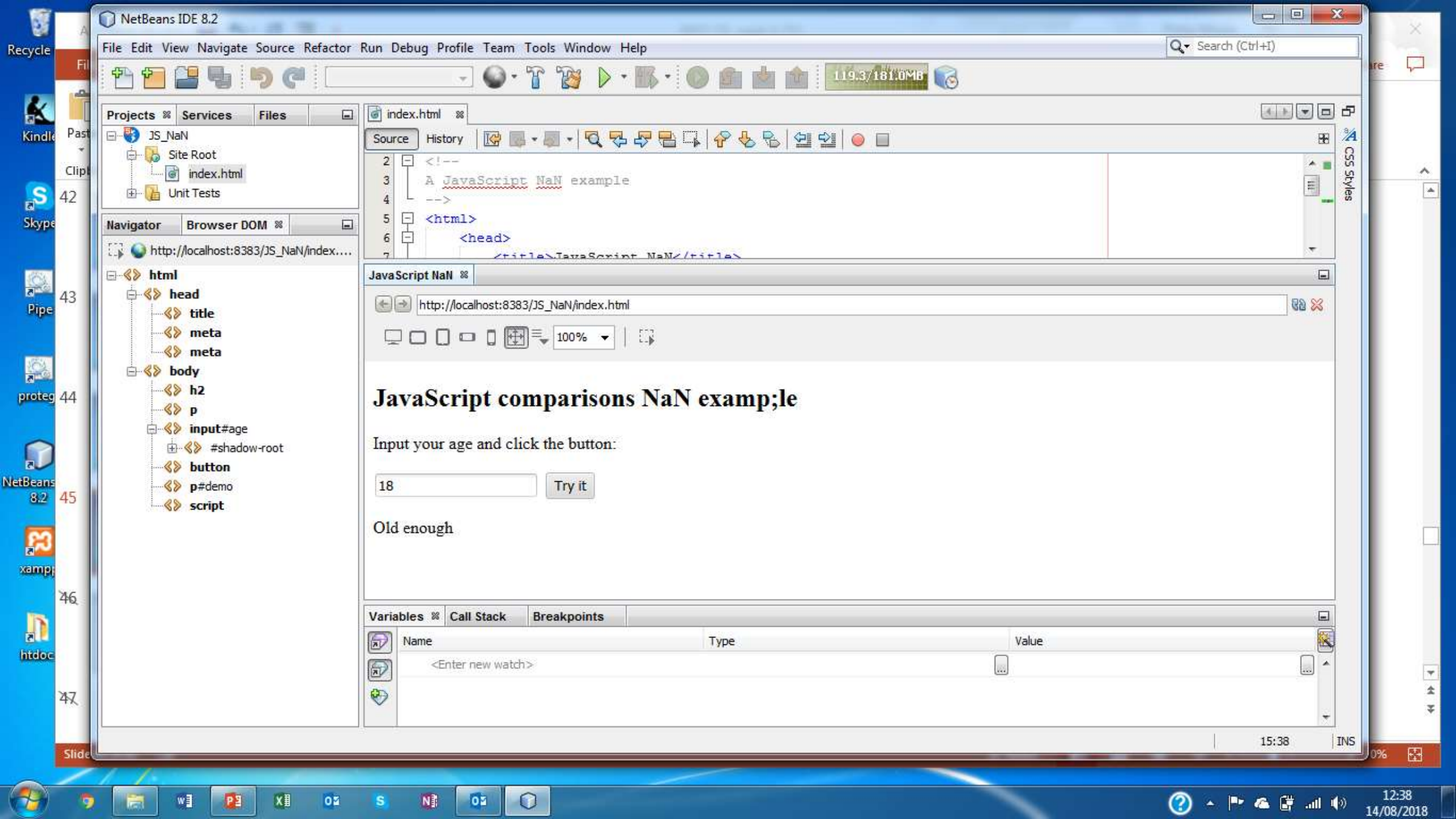
<script>
document.getElementById("demo1").innerHTML = 2 == "John";
document.getElementById("demo2").innerHTML = "2" == "12";
document.getElementById("demo3").innerHTML = "2" > "12";
document.getElementById("demo4").innerHTML = 2 < 12;
</script>
</body>
</html>
23:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
99.9/328.0MB
Projects
Services
Files
JS_Comp
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_Comp/index...
html
head
title
meta
meta
body
h2
p#demo1
p#demo2
p#demo3
p#demo4
script
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript comparison example
4 -->
5 <html>
CSS Styles
JavaScripr Comparison Ope...
http://localhost:8383/JS_Comp/index.html
100%
JavaScript comparison operators
false
false
true
true
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
23:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
192.3/255.5MB
Projects
Services
Files
JS_NaN
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_NaN/index....
html
head
title
meta
meta
body
h2
p
input#age
#shadow-root
button
p#demo
script
index.html
Source
History
2 <!--
3 A JavaScript NaN example
4 -->
5 <html>
6 <head>
CSS Styles
JavaScript NaN
http://localhost:8383/JS_NaN/index.html
100%
JavaScript comparisons NaN examp;le
Input your age and click the button:
Philip
Try it
Input is not a number
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
31:5
INS
12:34
14/08/2018


JS_NaN - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_NaN
Site Root
index.html
Unit Tests
Navigator
JavaScript
myFunction() : undefined
CSS
Ids
#age
#demo
HTML
html
head
title
meta
meta
body
h2
p
input id=age
button
p id=demo
script
index.html
Source History
<!--
A JavaScript NaN example
-->
<html>
<head>
<title>JavaScript NaN</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h2>JavaScript comparisons NaN examp;le</h2>
Input your age and click the button:
<!-- input "18" is commented to show NaN -->
<!--<input id="age" value="18" /> -->
<input id="age" value="Philip" />
<button onclick="myFunction()">Try it</button>

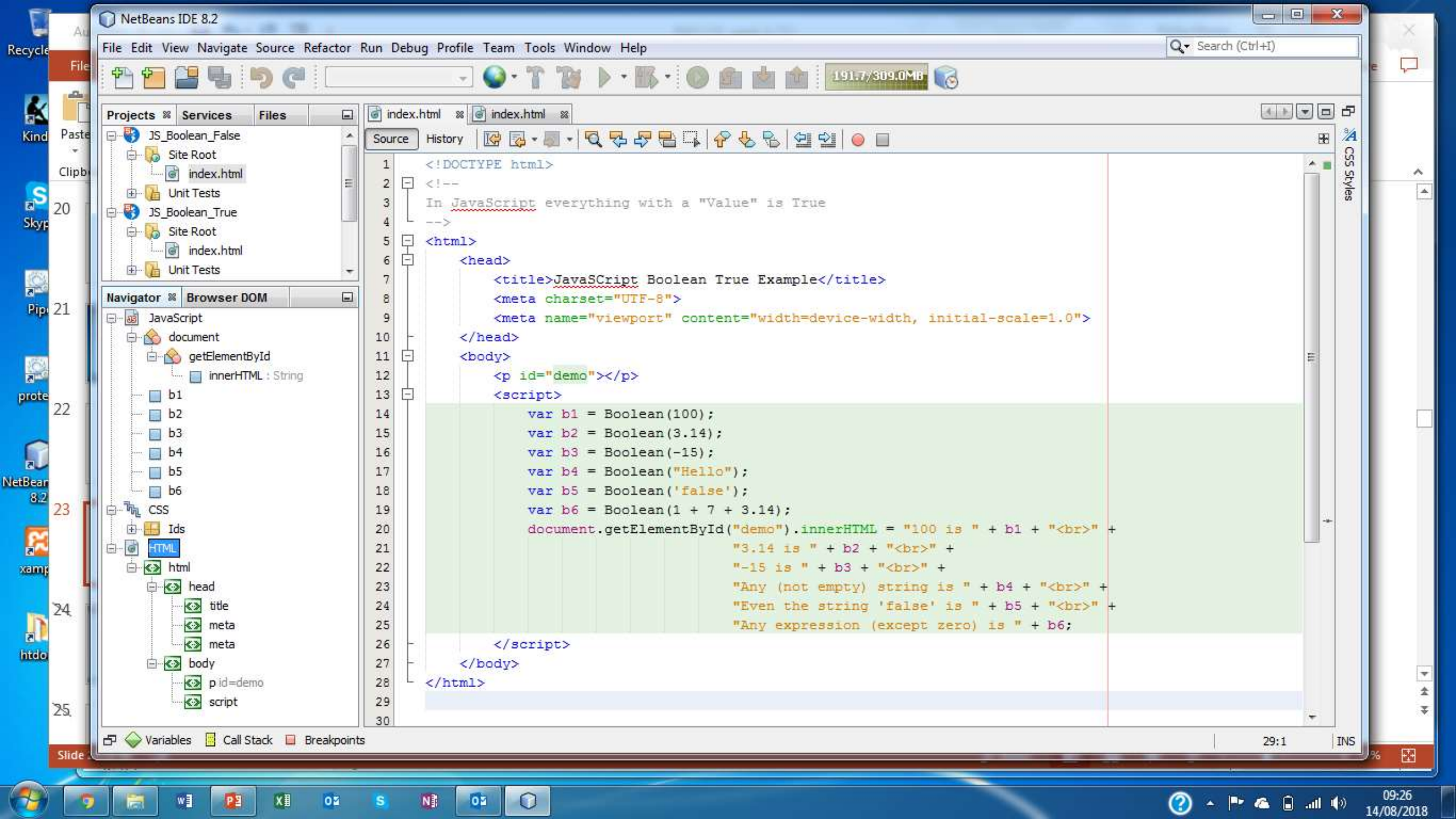
<script>
function myFunction() {
var age, voteable;
age = Number(document.getElementById("age").value);
if (isNaN(age)) {
voteable = "Input is not a number";
} else {
voteable = (age < 18) ? "Too young" : "Old enough";
}
document.getElementById("demo").innerHTML = voteable;
}
</script>
</body>
Variables Call Stack Breakpoints
14:25 INS

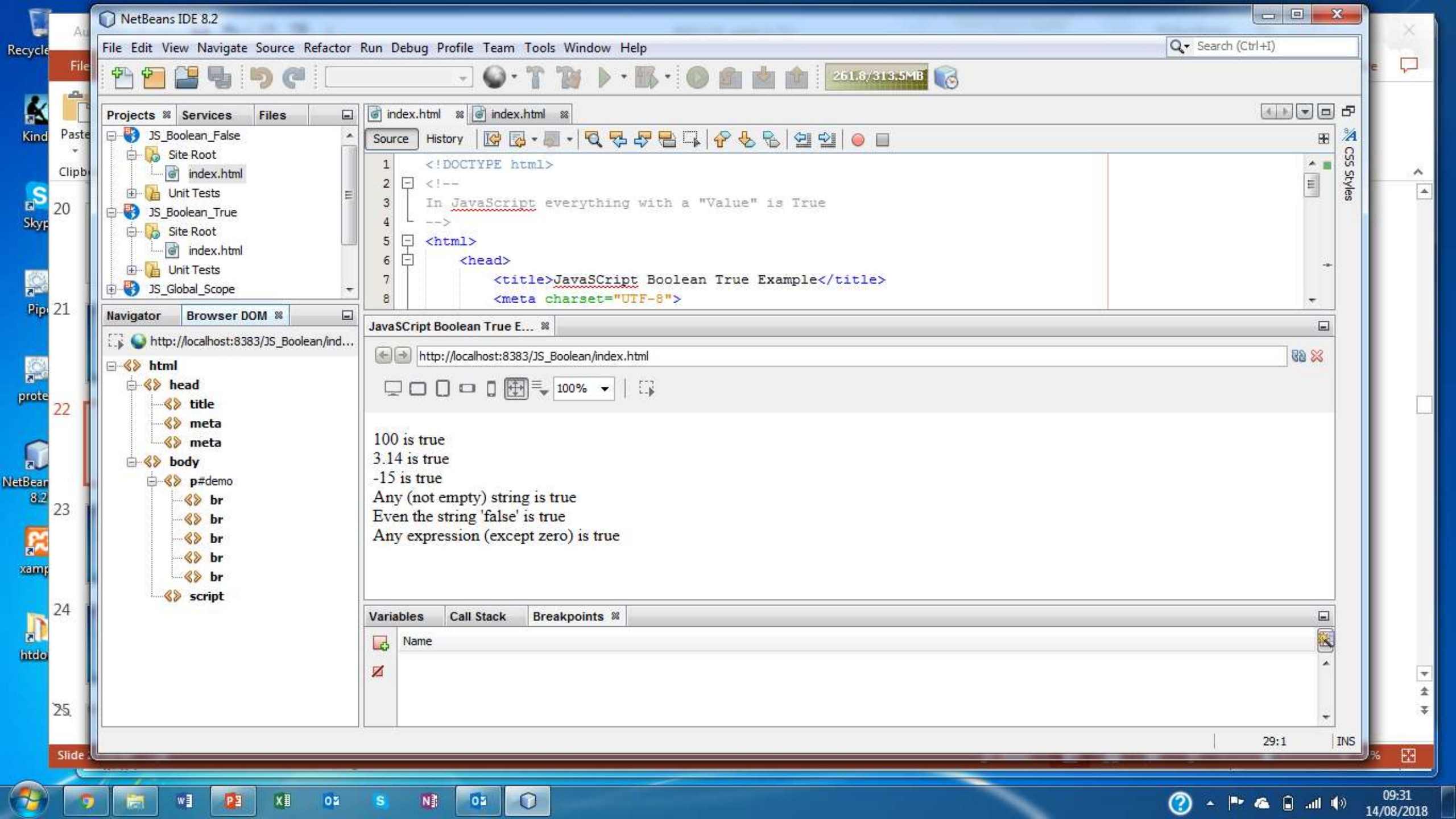

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
119.3/181.0MB
Projects
Services
Files
JS_NaN
Site Root
index.html
Unit Tests
Navigator
Browser DOM
http://localhost:8383/JS_NaN/index....
html
head
title
meta
meta
body
h2
p
input#age
#shadow-root
button
p#demo
script
index.html
Source
History
2 <!--
3 A JavaScript NaN example
4 -->
5 <html>
6 <head>
CSS Styles
JavaScript NaN
http://localhost:8383/JS_NaN/index.html
100%
JavaScript comparisons NaN examp;le
Input your age and click the button:
18
Try it
Old enough
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
15:38
INS

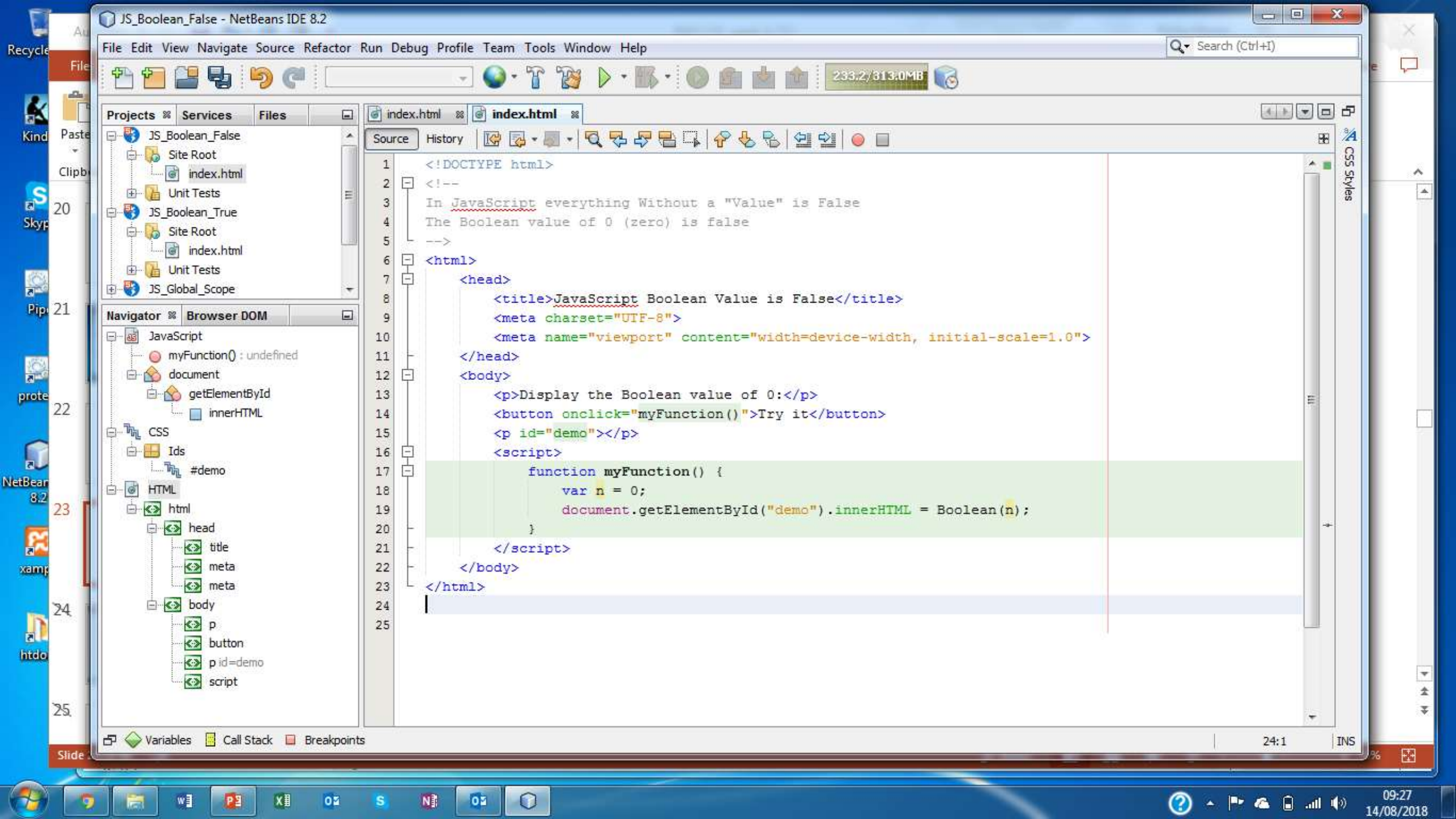
Boolean Values

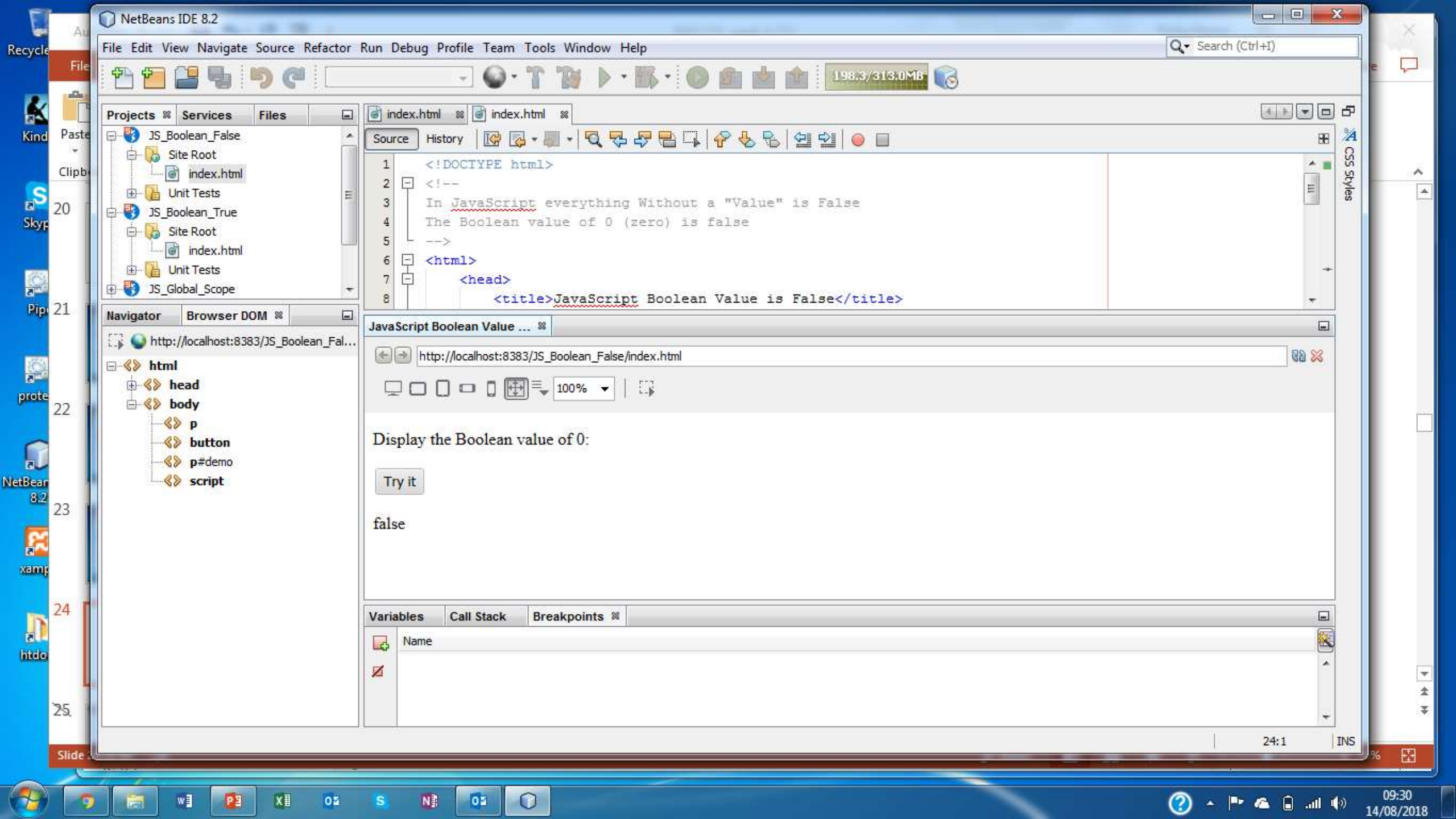
Boolean Values

- A Boolean value represents one of two values: **true** or **false**
- Programming requires a data type that can only have one of two values
  - **YES** / **NO**
  - **ON** / **OFF**
  - **TRUE** / **FALSE**
- JavaScript has a **Boolean** data type which can only take the values **true** or **false**
- For example:
  - **(10 > 9)** (returns **true**
  - **(10 >= 10)** (returns **true**)
  - **(10 >= 11)** (returns **false**)
  - **(10 < 9)** (returns **false**)
  - **(10 <= 10)** (returns **true**)


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
191.7/309.0MB
Projects Services Files
JS_Boolean_False
Site Root
index.html
Unit Tests
JS_Boolean_True
Site Root
index.html
Unit Tests
Navigator Browser DOM
JavaScript
document
getElementById
innerHTML : String
b1
b2
b3
b4
b5
b6
CSS
Ids
HTML
html
head
title
meta
meta
body
p id=demo
script
index.html
index.html
Source History
CSS Styles
1 <!DOCTYPE html>
2 <!--
3 In JavaScript everything with a "Value" is True
4 -->
5 <html>
6 <head>
7 <title>JavaScript Boolean True Example</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12
13 <script>
14 var b1 = Boolean(100);
15 var b2 = Boolean(3.14);
16 var b3 = Boolean(-15);
17 var b4 = Boolean("Hello");
18 var b5 = Boolean('false');
19 var b6 = Boolean(1 + 7 + 3.14);
20 document.getElementById("demo").innerHTML = "100 is " + b1 + "
" +
21 "3.14 is " + b2 + "
" +
22 "-15 is " + b3 + "
" +
23 "Any (not empty) string is " + b4 + "
" +
24 "Even the string 'false' is " + b5 + "
" +
25 "Any expression (except zero) is " + b6;
26 </script>
27 </body>
28 </html>
29
30
Variables Call Stack Breakpoints
29:1 INS
09:26
14/08/2018


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
261.8/313.5MB
Projects
Services
Files
JS_Boolean_False
Site Root
index.html
Unit Tests
JS_Boolean_True
Site Root
index.html
Unit Tests
JS_Global_Scope
Navigator
Browser DOM
http://localhost:8383/JS_Boolean/ind...
html
head
title
meta
meta
body
p#demo
br
br
br
br
br
script
index.html
index.html
Source
History
1 <!DOCTYPE html>
2 <!--
3 In JavaScript everything with a "Value" is True
4 -->
5 <html>
6 <head>
7 <title>JavaScript Boolean True Example</title>
8 <meta charset="UTF-8">
CSS Styles
JavaScript Boolean True E...
http://localhost:8383/JS_Boolean/index.html
100%
100 is true
3.14 is true
-15 is true
Any (not empty) string is true
Even the string 'false' is true
Any expression (except zero) is true
Variables
Call Stack
Breakpoints
Name
29:1
INS


JS_Boolean_False - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
233.2/313.0MB
Projects Services Files
JS_Boolean_False
Site Root
index.html
Unit Tests
JS_Boolean_True
Site Root
index.html
Unit Tests
JS_Global_Scope
Navigator Browser DOM
JavaScript
myFunction() : undefined
document
getElementById
innerHTML
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 In JavaScript everything Without a "Value" is False
4 The Boolean value of 0 (zero) is false
5 -->
6 <html>
7 <head>
8 <title>JavaScript Boolean Value is False</title>
9 <meta charset="UTF-8">
10 <meta name="viewport" content="width=device-width, initial-scale=1.0">
11 </head>
12 <body>
13 Display the Boolean value of 0:
14 <button onclick="myFunction()">Try it</button>
15
16 <script>
17 function myFunction() {
18 var n = 0;
19 document.getElementById("demo").innerHTML = Boolean(n);
20 }
21 </script>
22 </body>
23 </html>
24
25
CSS Styles
Variables Call Stack Breakpoints
24:1 INS
09:27
14/08/2018


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
198.3/313.0MB
Projects Services Files
JS_Boolean_False
Site Root
index.html
Unit Tests
JS_Boolean_True
Site Root
index.html
Unit Tests
JS_Global_Scope
Navigator Browser DOM
http://localhost:8383/JS_Boolean_Fal...
html
head
body
p
button
p#demo
script
index.html
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 In JavaScript everything Without a "Value" is False
4 The Boolean value of 0 (zero) is false
5 -->
6 <html>
7 <head>
8 <title>JavaScript Boolean Value is False</title>
CSS Styles
JavaScript Boolean Value ...
http://localhost:8383/JS_Boolean_False/index.html
100%
Display the Boolean value of 0:
Try it
false
Variables Call Stack Breakpoints
Name
24:1
INS

Ternary Conditional Operator

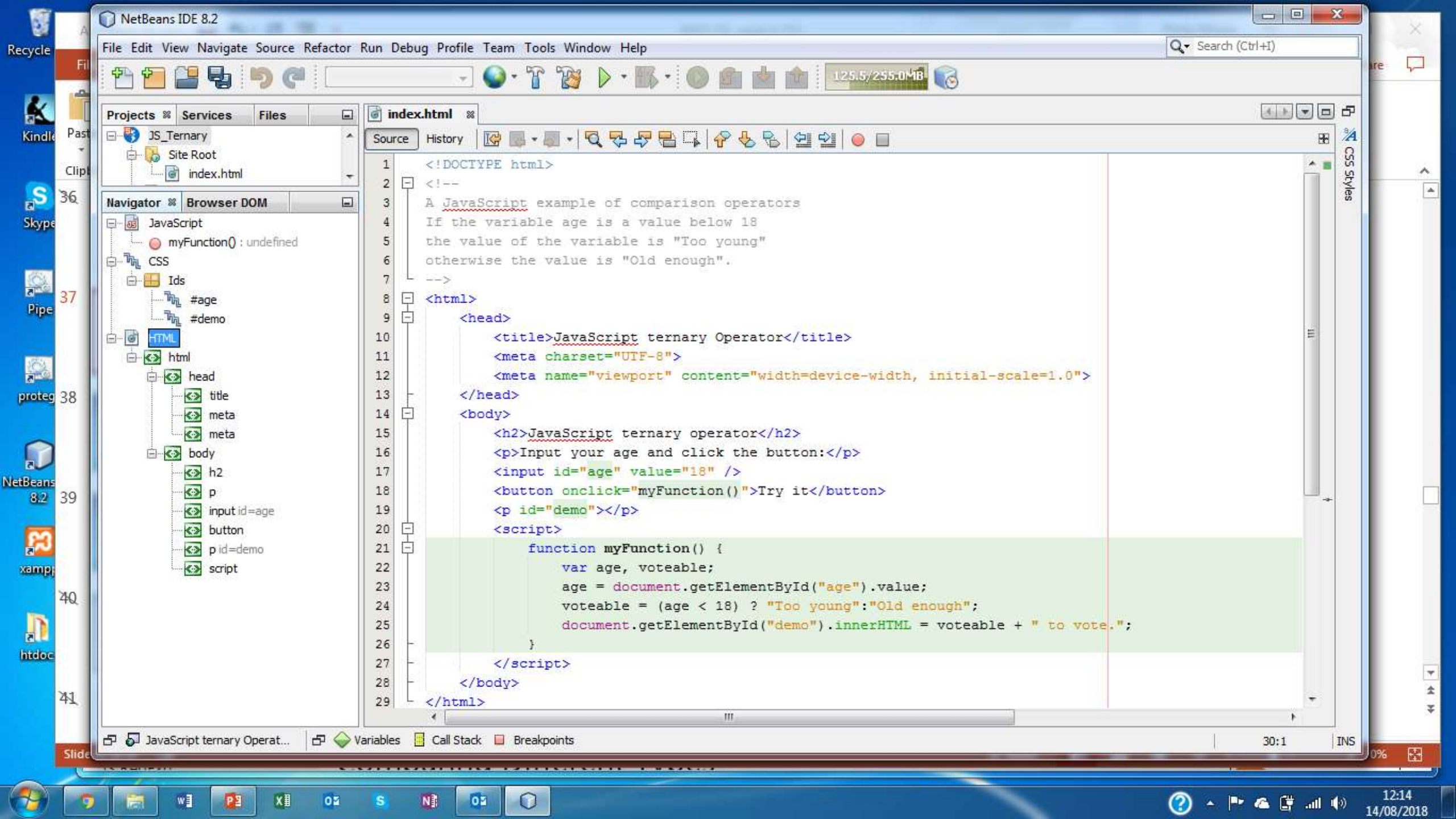
Conditional (Ternary) Operator

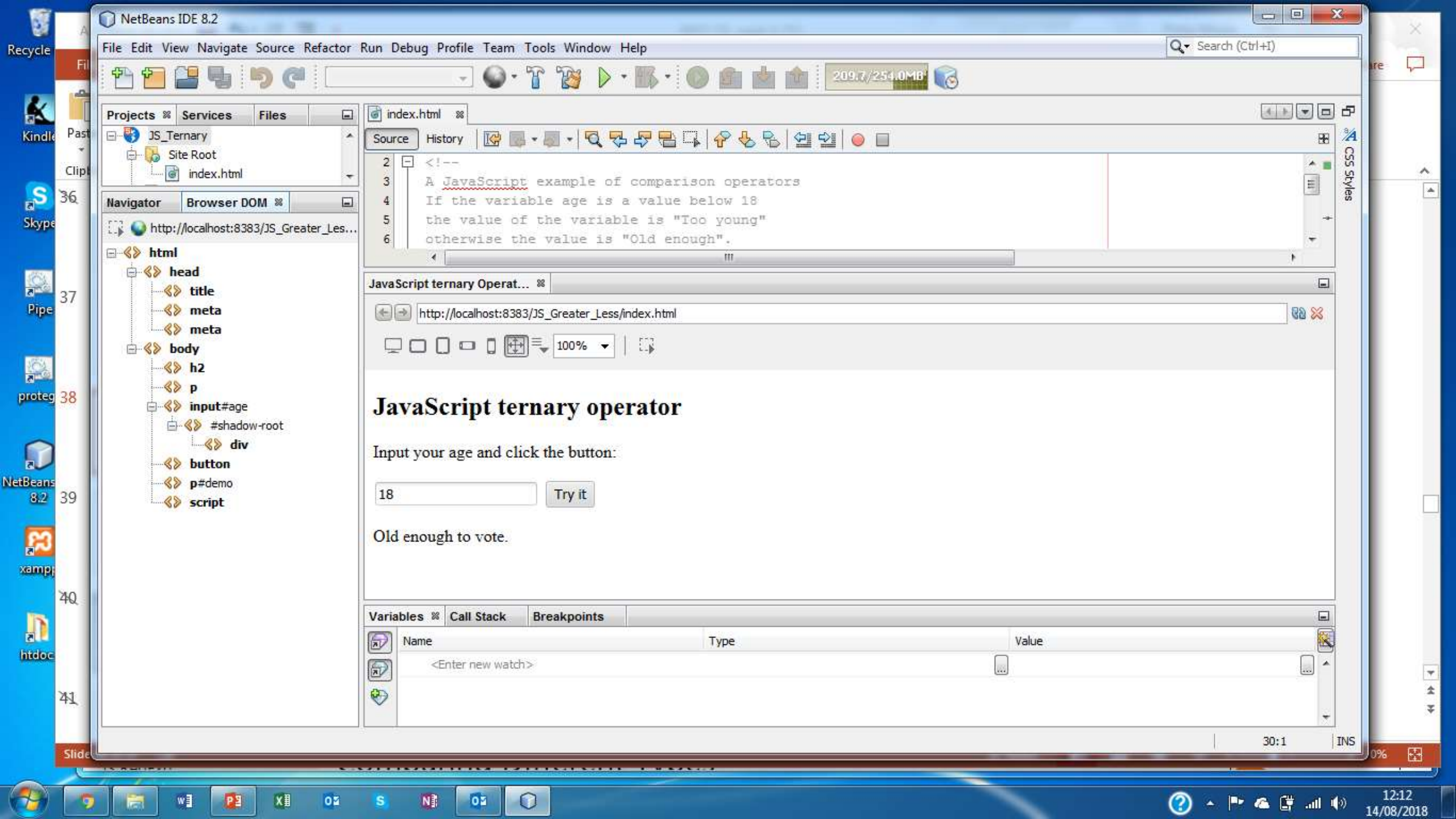
- JavaScript contains a conditional operator that assigns a value to a variable based on some condition
- The syntax

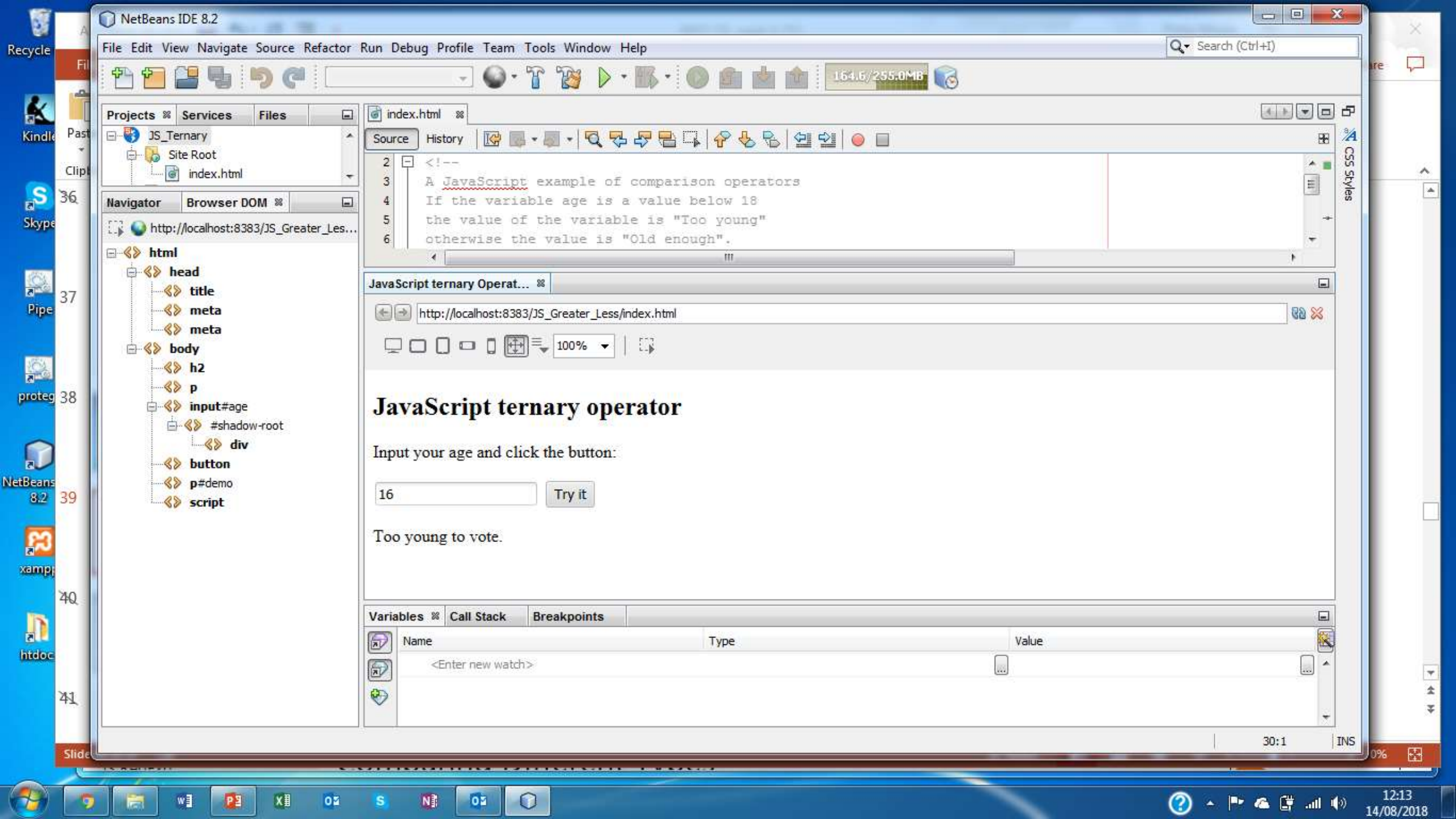
```
variablename = (condition) ? value1:value2
```

- For example

```
var voteable = (age < 18) ? "Too young":"Old
enough";
```


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
125.5/255.0MB
Projects Services Files
JS_Ternary
Site Root
index.html
Navigator Browser DOM
JavaScript
myFunction() : undefined
CSS
Ids
#age
#demo
HTML
html
head
title
meta
meta
body
h2
p
input id=age
button
p id=demo
script
index.html
Source History
CSS Styles
1 <!DOCTYPE html>
2 <!--
3 A JavaScript example of comparison operators
4 If the variable age is a value below 18
5 the value of the variable is "Too young"
6 otherwise the value is "Old enough".
7 -->
8 <html>
9 <head>
10 <title>JavaScript ternary Operator</title>
11 <meta charset="UTF-8">
12 <meta name="viewport" content="width=device-width, initial-scale=1.0">
13 </head>
14 <body>
15 <h2>JavaScript ternary operator</h2>
16 Input your age and click the button:
17 <input id="age" value="18" />
18 <button onclick="myFunction()">Try it</button>
19
20 <script>
21 function myFunction() {
22 var age, voteable;
23 age = document.getElementById("age").value;
24 voteable = (age < 18) ? "Too young":"Old enough";
25 document.getElementById("demo").innerHTML = voteable + " to vote.";
26 }
27 </script>
28 </body>
29 </html>
JavaScript ternary Operat...
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)
209.7/254.0MB
Projects
Services
Files
JS_Ternary
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_Greater_Les...
html
head
title
meta
meta
body
h2
p
input#age
#shadow-root
div
button
p#demo
script
index.html
Source
History
2 <!--
3 A JavaScript example of comparison operators
4 If the variable age is a value below 18
5 the value of the variable is "Too young"
6 otherwise the value is "Old enough".
CSS Styles
JavaScript ternary Operat...
http://localhost:8383/JS_Greater_Less/index.html
100%
JavaScript ternary operator
Input your age and click the button:
18
Try it
Old enough to vote.
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
30:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
164.6/255.0MB
Projects
Services
Files
JS_Ternary
Site Root
index.html
Navigator
Browser DOM
http://localhost:8383/JS_Greater_Les...
html
head
title
meta
meta
body
h2
p
input#age
#shadow-root
div
button
p#demo
script
index.html
Source
History
2 <!--
3 A JavaScript example of comparison operators
4 If the variable age is a value below 18
5 the value of the variable is "Too young"
6 otherwise the value is "Old enough".
CSS Styles
JavaScript ternary Operat...
http://localhost:8383/JS_Greater_Less/index.html
100%
JavaScript ternary operator
Input your age and click the button:
16
Try it
Too young to vote.
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
30:1
INS

Events
and
Event Handling

HTML Events

- HTML events are:
  - Web browser events (or)
  - User actions
- Typical HTML events include:
  - An HTML web-page has finished loading
  - An HTML input field has changed or has been updated
  - A HTML button (e.g., in a form) has been clicked
  - When an event happens users may wish do something
  - JavaScript allows users to execute code when events are detected
  - HTML allows event handler attributes (with JavaScript code) to be added to HTML elements

Common HTML Events

- Here is a list of some common reactive HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page
Event	Description

- A full list of HTML events may be found in the course resources

Event Handlers in JavaScript

- Event handlers can be used to:
  - Handle and verify user input and actions
  - Handle browser actions
    - When a page loads
    - When a page is closed
    - User actions such as when a user clicks a button
    - Feedback when a cursor hovers over a web-page element (mouseover)
    - etc …
- For further details in event handlers see the course resources

JavaScript Events and Methods

- Many different methods can be used to let JavaScript work with events:
  - HTML event attributes can execute JavaScript code directly
  - HTML event attributes can call JavaScript functions
  - User defined events can be assigned to a user defined event handler functions to HTML elements
  - Events can be prevented from being sent or handled
- etc ...

Event Handlers

- Events are triggered with single quotes and double quotes
  - With single quotes:

```
<element event='some JavaScript'>
```

  - With double quotes:

```
<element event="some JavaScript">
```

- In the following example, an onclick attribute (with code), is added to a button element:

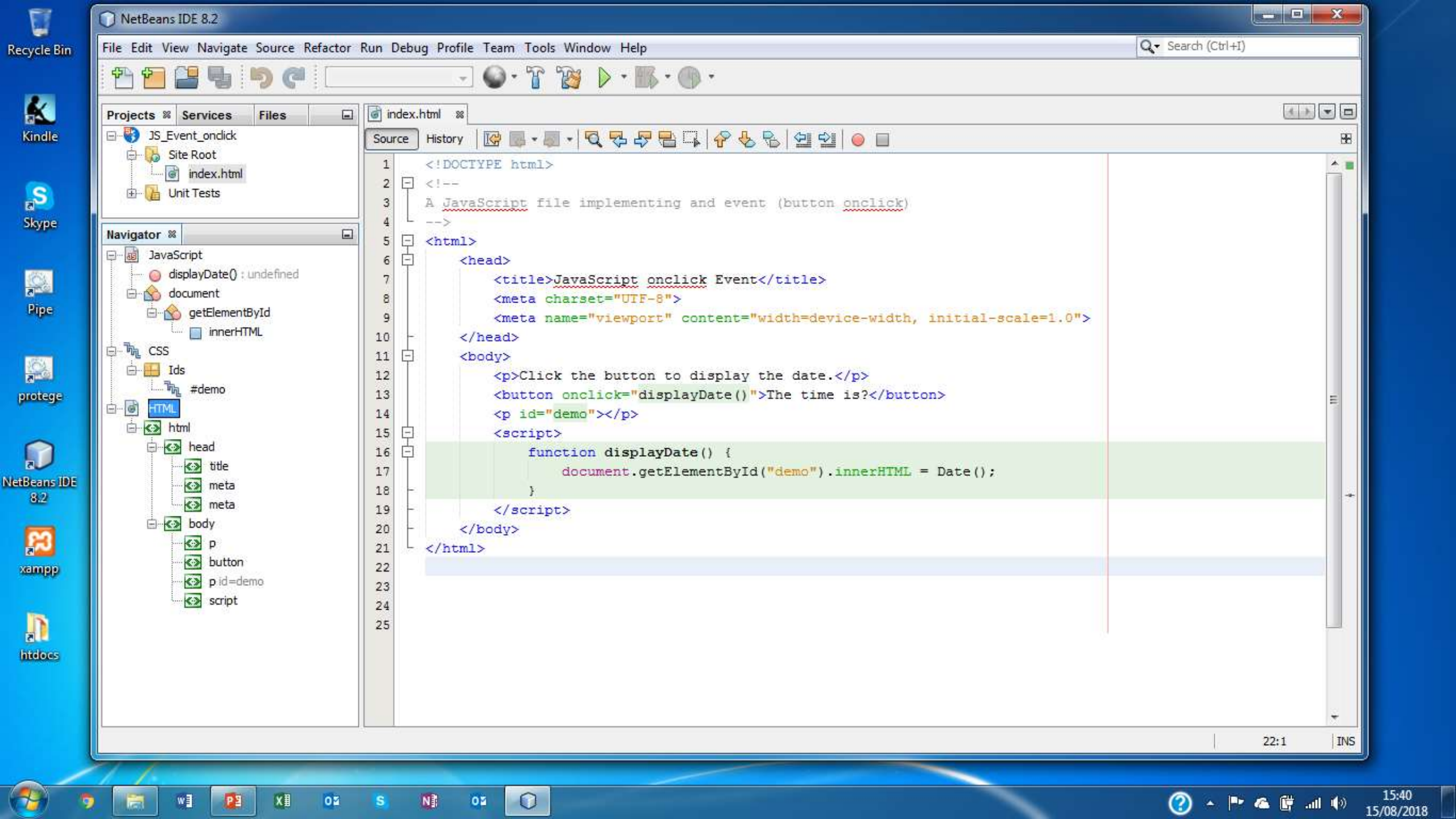
```
<button onclick="document.getElementById('demo').innerHTML =
Date()">The time is?</button>
```

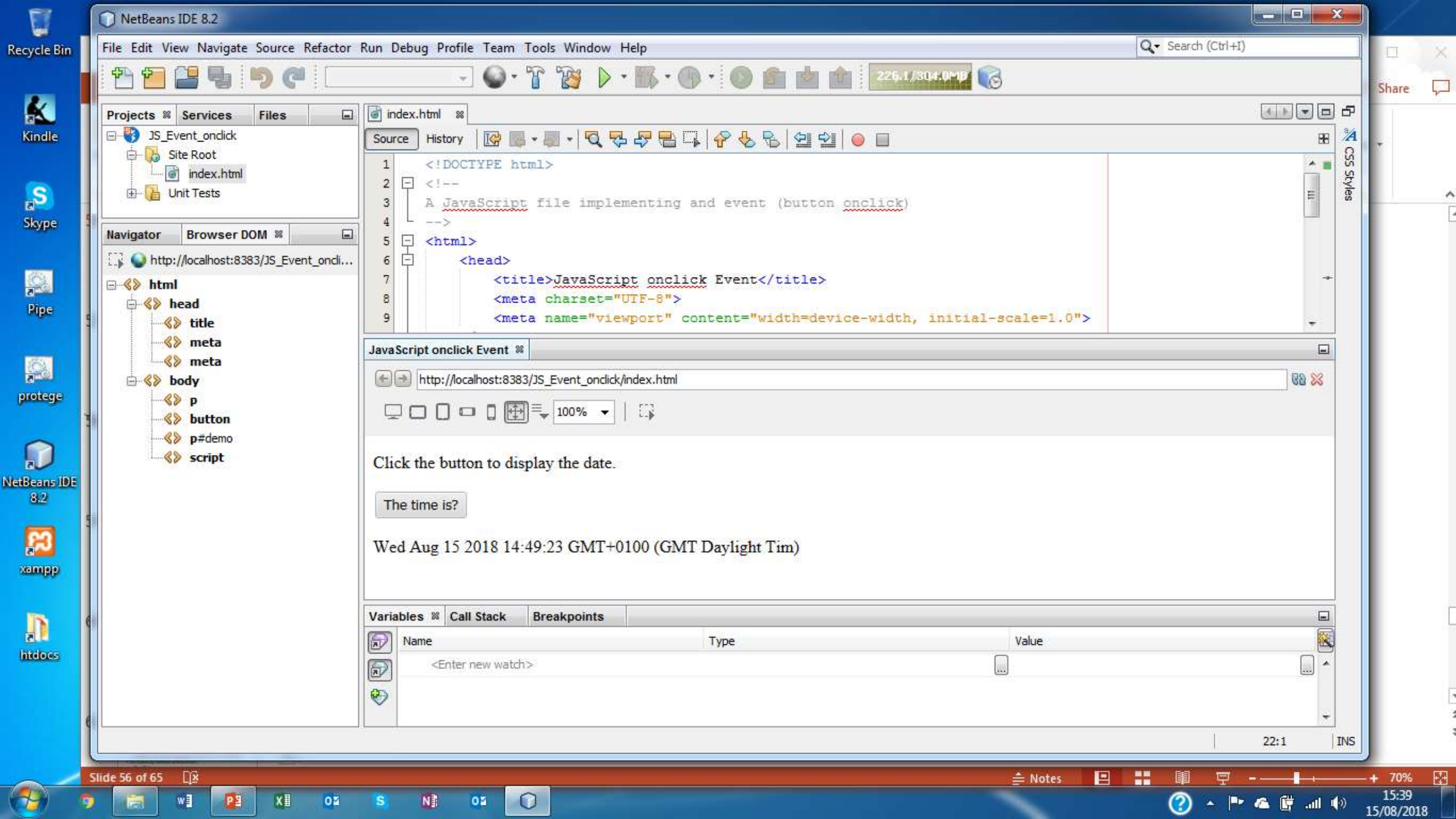
Event Handlers

- JavaScript code can be complex is often several lines long
- It is more common to see event attributes calling functions:
- For example

```
<button onclick="displayDate()">The time is?</button
```

- The following example demonstrates
  - The HTML
  - A button (clicked to produce the current day, date, and time)
  - The HTML onclick method
  - The function **`displayDate()`** which does not return a value but directly outputs the date information


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Event_onclick
Site Root
index.html
Unit Tests
Navigator
JavaScript
displayDate() : undefined
document
getElementById
innerHTML
CSS
Ids
#demo
HTML
html
head
title
meta
meta
body
p
button
p id=demo
script
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file implementing and event (button onclick)
4 -->
5 <html>
6 <head>
7 <title>JavaScript onclick Event</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 the date.
13 <button onclick="displayDate()">The time is?</button>
14
15 <script>
16 function displayDate() {
17 document.getElementById("demo").innerHTML = Date();
18 }
19 </script>
20 </body>
21 </html>
22 23 24 25
22:1
INS
Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
15:40
15/08/2018


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Event_onclick
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_Event_oncli...
html
head
title
meta
meta
body
p
button
p#demo
script
index.html
Source History
1 <!DOCTYPE html>
2 <!--
3 A JavaScript file implementing and event (button onclick)
4 -->
5 <html>
6 <head>
7 <title>JavaScript onclick Event</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
CSS Styles
JavaScript onclick Event
http://localhost:8383/JS_Event_onclick/index.html
100%
Click the button to display the date.
The time is?
Wed Aug 15 2018 14:49:23 GMT+0100 (GMT Daylight Tim)
Variables Call Stack Breakpoints
Name Type Value
<Enter new watch>
22:1 INS
Slide 56 of 65
Notes
70%
15:39
15/08/2018

HTML DOM Events

- HTML DOM Events
  - Allow JavaScript to register different event handlers on elements in an HTML document
  - Events are normally used in combination with functions
  - Functions will not be executed before the event occurs
    - such as when a user clicks a button
- HTML DOM Events have properties and methods
  - For a full list of available events with properties and methods see the course resources

DOM Event Objects

- Event Objects
  - When an event occur in HTML, the event belongs to a certain event object, like a mouse click event belongs to the **`MouseEvent`** object
- The Event Object
  - All event objects
  - Are based on the Event Object
  - Event objects inherit all the properties and methods
- For details of the DOM Event Objects see the course resources

Catching Runtime Errors

`Try…catch`

`try…catch…finally`

Potential Runtime Errors

- JavaScript runtime error checking methods will **`catch`** a range of potential errors including
  - HTML Validation Errors
  - Range Errors
  - Reference Errors
  - Syntax Errors (note: this will not find logic errors)
  - Type Errors
  - URI (Uniform Resource Identifier) Errors
- A list of the possible errors with the syntax and examples may be found at
  - https://www.w3schools.com/js/js_errors.asp
- Examples of the syntax and examples are shown in the following slides

Non-Standard Error Object Properties

- Mozilla and Microsoft defines some non-standard error object properties:
  - *fileName* (Mozilla)
  - *lineNumber* (Mozilla)
  - *columnNumber* (Mozilla)
  - *stack* (Mozilla)
  - *description* (Microsoft)
  - *number* (Microsoft)
- Do not use these properties in public web sites as they will not work in all browsers

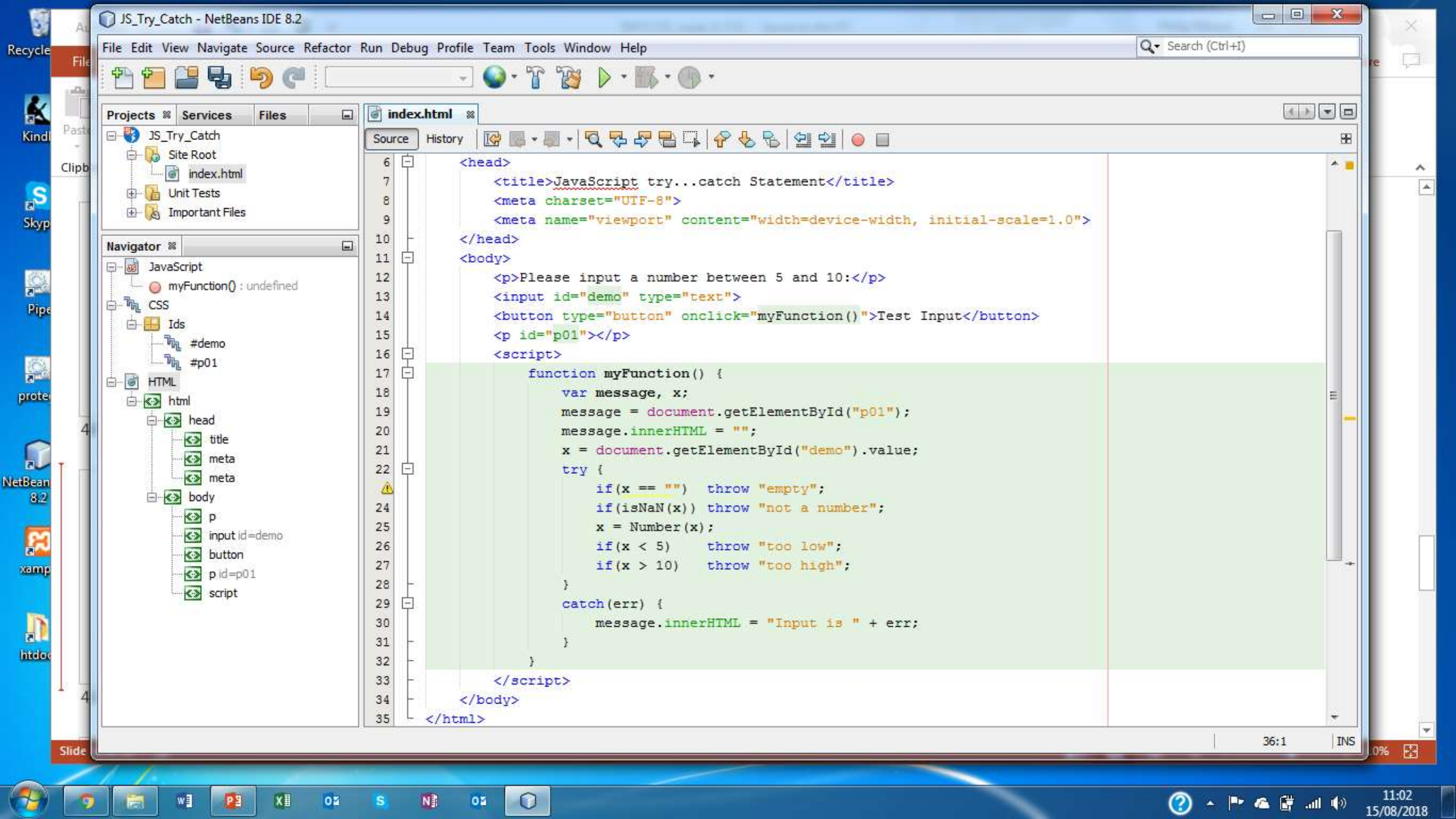
Catching Runtime Errors

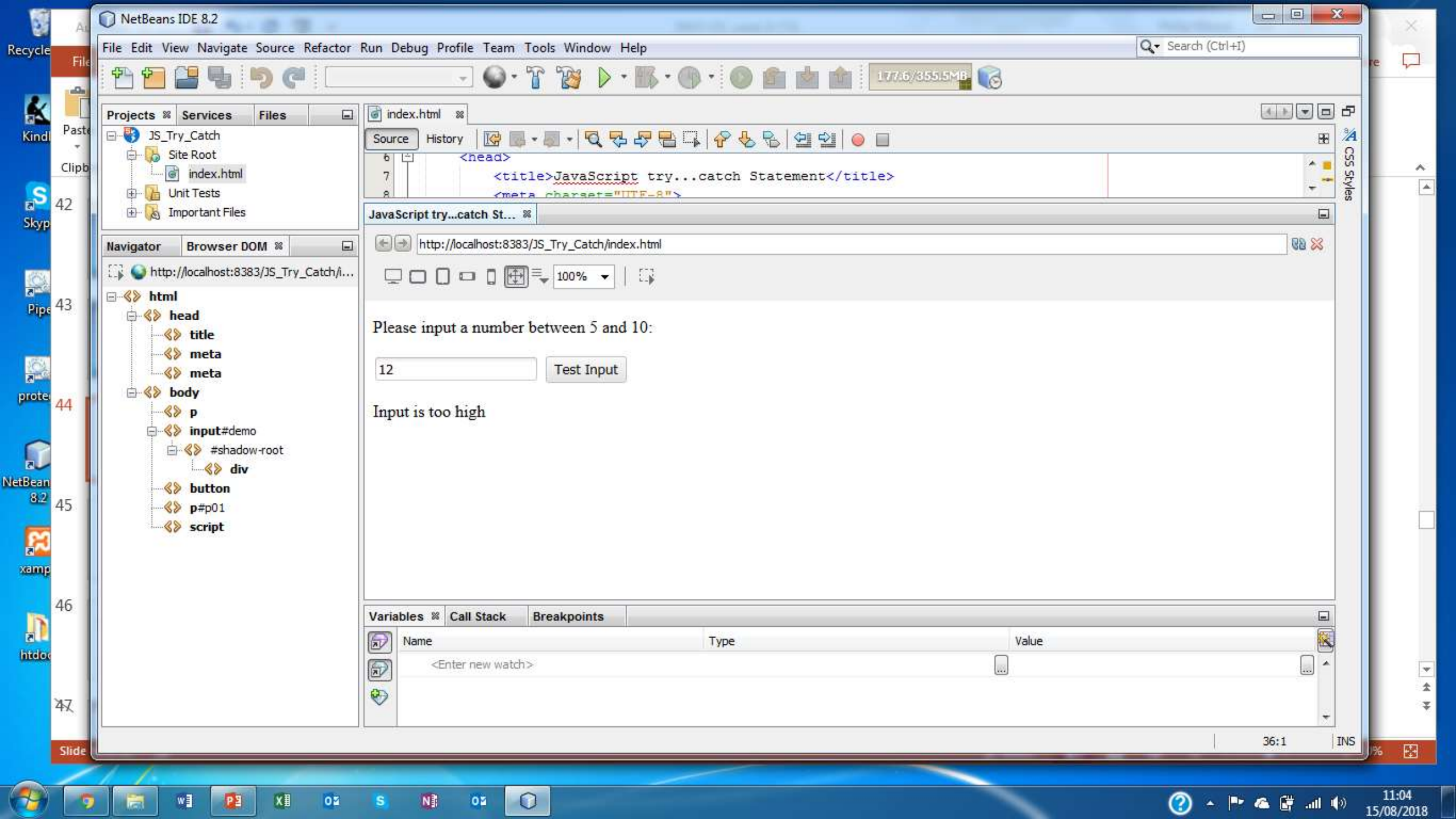
- When executing JavaScript code different runtime errors can occur
- Errors include
  - Invalid input by a user
  - There are other (possibly unforeseen) runtime errors
- JavaScript provides the throw and try and catch methods to catch errors
  - The **`try`** statement tests a block of code for runtime errors
  - The **`catch`** statement handles the runtime error
  - The **`throw`** statement creates custom errors
  - The **`finally`** statement execute JavaScript program code after the try and catch methods regardless of the result

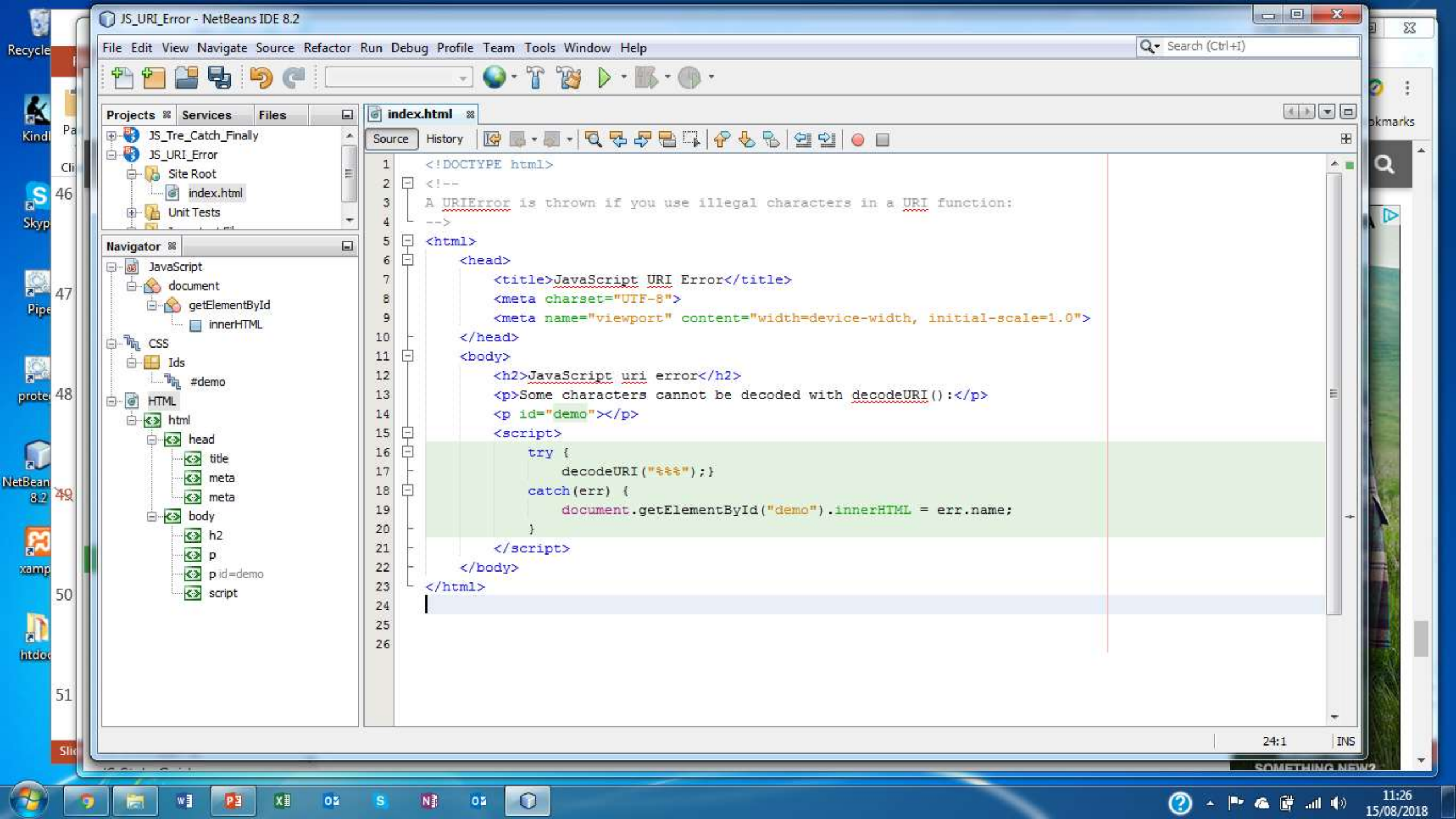
`try` … `catch` block

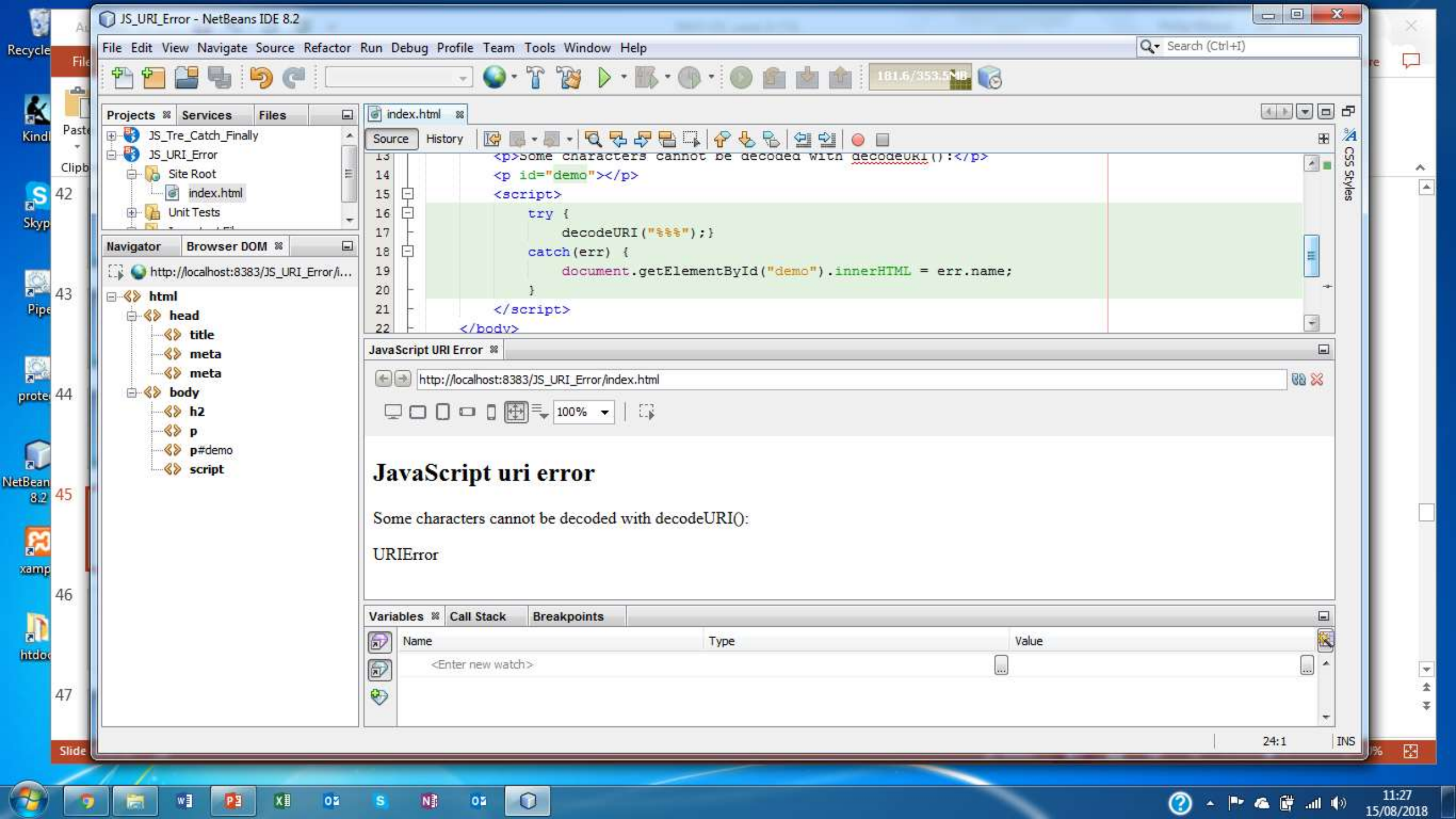
- The **`try`** statement defines a block of code to be tested for errors while it is being executed (at ***runtime***)
- The **`catch`** statement defines a block of code to be executed if an error occurs in the **`try`** block.
- The JavaScript statements **`try`** and **`catch`** come in pairs using the following syntax:

```
try {
    Block of code to try
}
catch(err) {
    Block of code to handle errors
}
```


JS_Try_Catch - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Try_Catch
Site Root
index.html
Unit Tests
Important Files
Navigator
JavaScript
myFunction() : undefined
CSS
Ids
#demo
#p01
HTML
html
head
title
meta
meta
body
p
input id=demo
button
p id=p01
script
index.html
Source History
6 <head>
7 <title>JavaScript try...catch Statement</title>
8 <meta charset="UTF-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 Please input a number between 5 and 10:
13 <input id="demo" type="text">
14 <button type="button" onclick="myFunction()">Test Input</button>
15
16 <script>
17 function myFunction() {
18 var message, x;
19 message = document.getElementById("p01");
20 message.innerHTML = "";
21 x = document.getElementById("demo").value;
22 try {
24 if(x == "") throw "empty";
25 if(isNaN(x)) throw "not a number";
26 x = Number(x);
27 if(x < 5) throw "too low";
28 if(x > 10) throw "too high";
29 }
30 catch(err) {
31 message.innerHTML = "Input is " + err;
32 }
33 }
34 </script>
35 </body>
</html>
36:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
177.6/355.5MB
Projects
Services
Files
JS_Try_Catch
Site Root
index.html
Unit Tests
Important Files
Navigator
Browser DOM
http://localhost:8383/JS_Try_Catch/i...
html
head
title
meta
meta
body
p
input#demo
#shadow-root
div
button
p#p01
script
index.html
Source
History
<head>
<title>JavaScript try...catch Statement</title>
JavaScript try...catch St...
http://localhost:8383/JS_Try_Catch/index.html
100%
Please input a number between 5 and 10:
12
Test Input
Input is too high
Variables
Call Stack
Breakpoints
Name
Type
Value
<Enter new watch>
CSS Styles
36:1
INS


JS_URI_Error - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Tre_Catch_Finally
JS_URI_Error
Site Root
index.html
Unit Tests
Navigator
JavaScript
document
getElementById
innerHTML
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 URIError is thrown if you use illegal characters in a URI function:
4 -->
5 <html>
6 <head>
7 <title>JavaScript URI Error</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 uri error</h2>
13 Some characters cannot be decoded with decodeURI():
14
15 <script>
16 try {
17 decodeURI("%%%");}
18 catch(err) {
19 document.getElementById("demo").innerHTML = err.name;
20 }
21 </script>
22 </body>
23 </html>
24
25
26
24:1
INS
11:26
15/08/2018

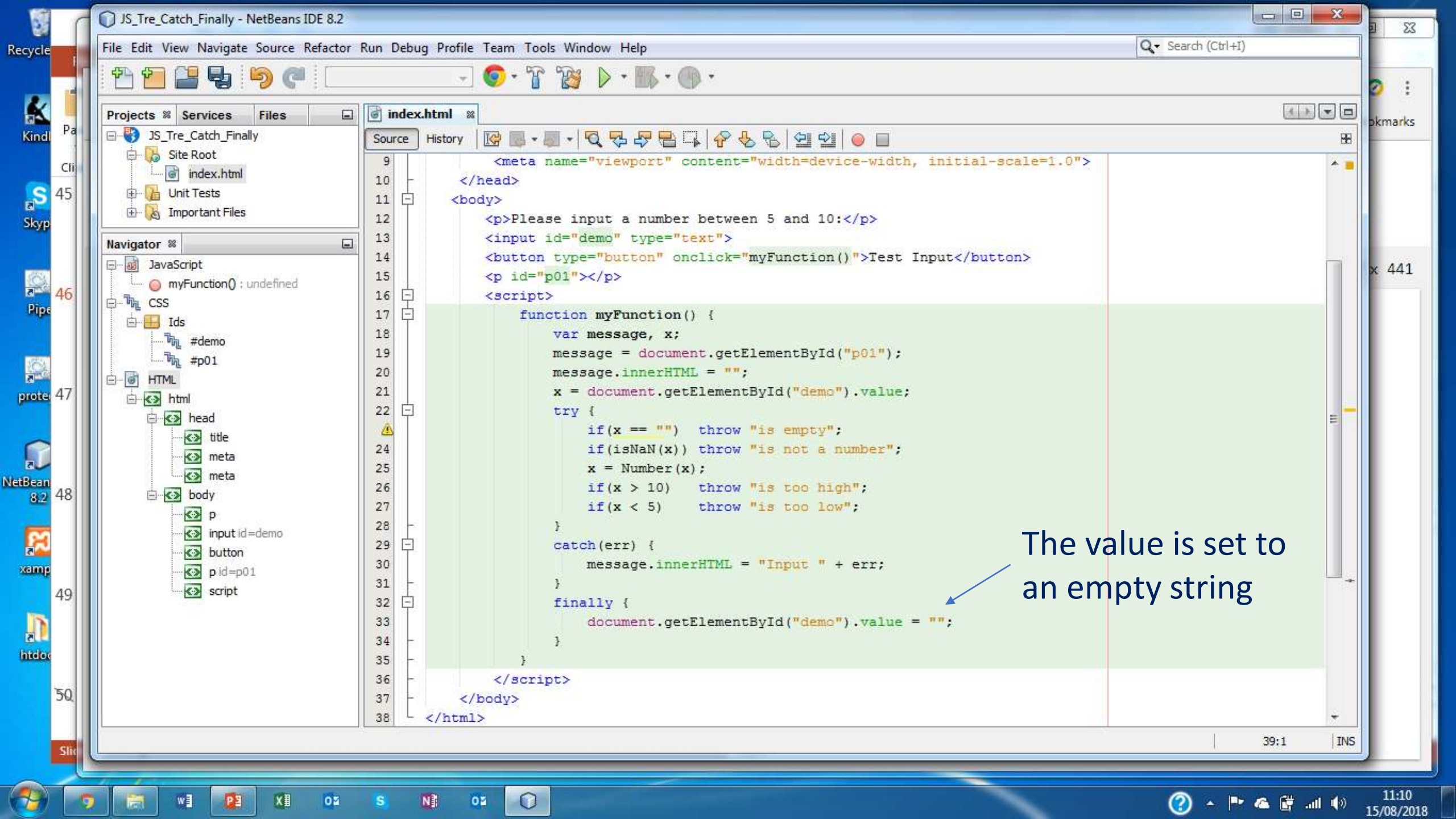

JS_URI_Error - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Tre_Catch_Finally
JS_URI_Error
Site Root
index.html
Unit Tests
Navigator Browser DOM
http://localhost:8383/JS_URI_Error/i...
html
head
title
meta
meta
body
h2
p
p#demo
script
index.html
Source History
Some characters cannot be decoded with decodeURI():

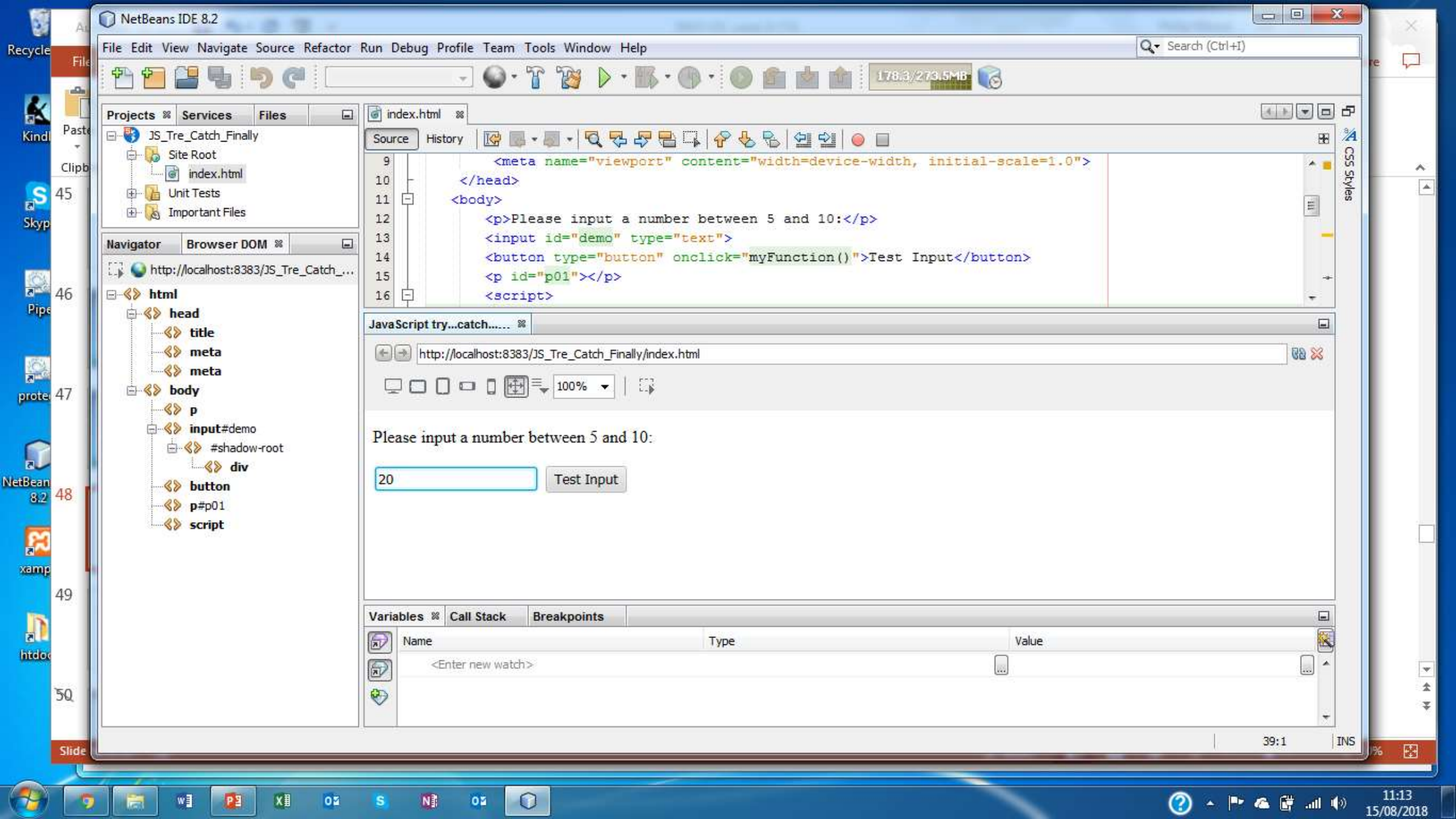
<script>
try {
decodeURI("%%%");}
catch(err) {
document.getElementById("demo").innerHTML = err.name;
}
</script>
</body>
CSS Styles
JavaScript URI Error
http://localhost:8383/JS_URI_Error/index.html
100%
JavaScript uri error
Some characters cannot be decoded with decodeURI():
URIError
Variables Call Stack Breakpoints
Name
Type
Value
<Enter new watch>
24:1
INS

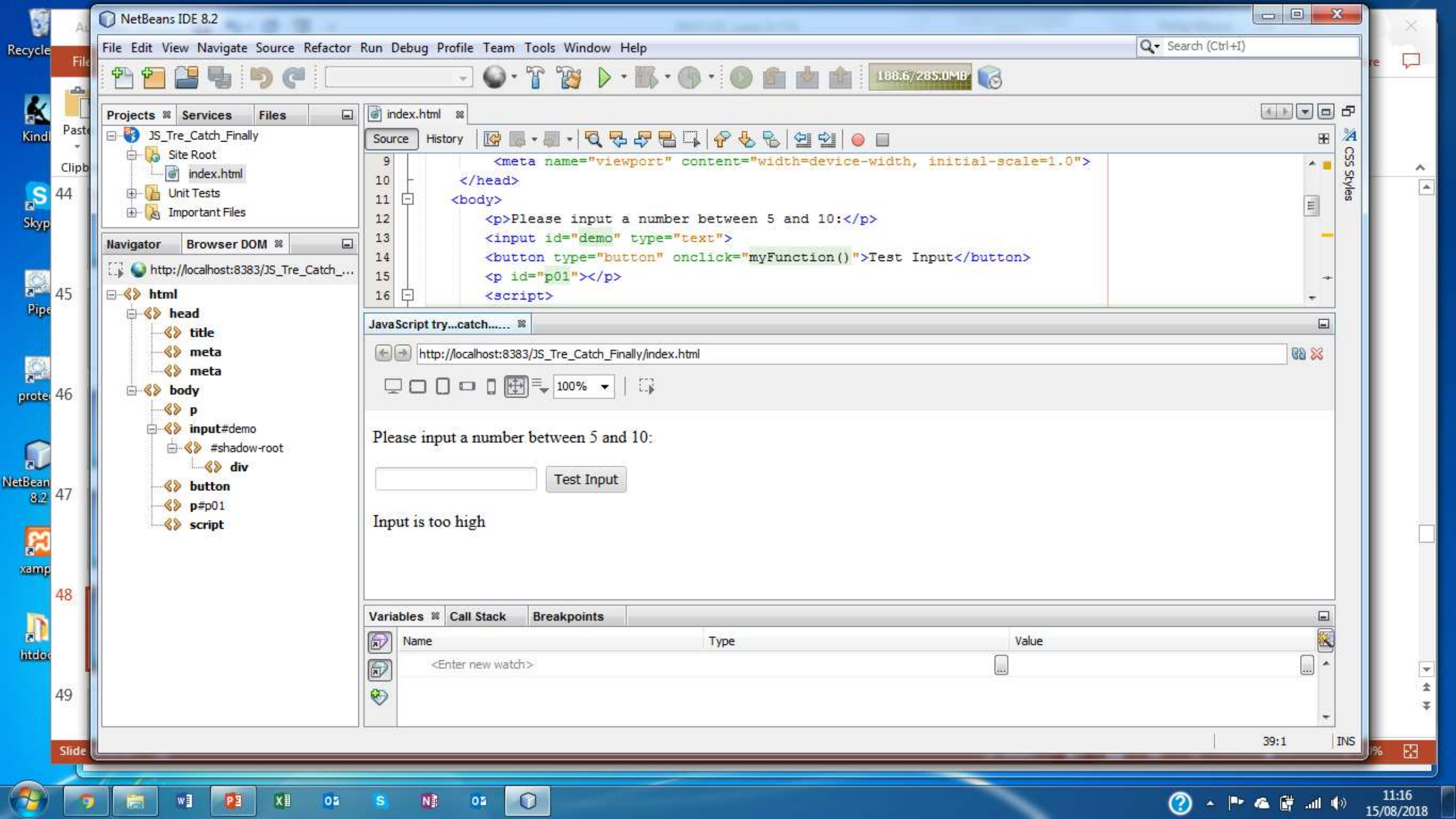
`try…catch…finally` block

- The syntax for a **`try … catch … finally`** block is:

```
try {
        statements
 }
catch ( arguments ) {
}
finally {
        statements
}
```


JS_Tre_Catch_Finally - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Services Files
JS_Tre_Catch_Finally
Site Root
index.html
Unit Tests
Important Files
Navigator
JavaScript
myFunction() : undefined
CSS
Ids
#demo
#p01
HTML
html
head
title
meta
meta
body
p
input id=demo
button
p id=p01
script
index.html
Source History
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 Please input a number between 5 and 10:
13 <input id="demo" type="text">
14 <button type="button" onclick="myFunction()">Test Input</button>
15
16 <script>
17 function myFunction() {
18 var message, x;
19 message = document.getElementById("p01");
20 message.innerHTML = "";
21 x = document.getElementById("demo").value;
22 try {
24 if(x == "") throw "is empty";
25 if(isNaN(x)) throw "is not a number";
26 x = Number(x);
27 if(x > 10) throw "is too high";
28 if(x < 5) throw "is too low";
29 }
30 catch(err) {
31 message.innerHTML = "Input " + err;
32 }
33 finally {
34 document.getElementById("demo").value = "";
35 }
36 }
37 </script>
38 </body>
</html>
The value is set to an empty string
39:1
INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
178.3/273.5MB
Projects Services Files
JS_Tre_Catch_Finally
Site Root
index.html
Unit Tests
Important Files
Navigator Browser DOM
http://localhost:8383/JS_Tre_Catch_...
html
head
title
meta
meta
body
p
input#demo
#shadow-root
div
button
p#p01
script
index.html
Source History
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 Please input a number between 5 and 10:
13 <input id="demo" type="text">
14 <button type="button" onclick="myFunction()">Test Input</button>
15
16 <script>
CSS Styles
JavaScript try...catch.....
http://localhost:8383/JS_Tre_Catch_Finally/index.html
100%
Please input a number between 5 and 10:
20
Test Input
Variables Call Stack Breakpoints
Name Type Value
<Enter new watch>
39:1 INS


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
188.6/285.0MB
Projects Services Files
JS_Tre_Catch_Finally
Site Root
index.html
Unit Tests
Important Files
Navigator Browser DOM
http://localhost:8383/JS_Tre_Catch_...
html
head
title
meta
meta
body
p
input#demo
#shadow-root
div
button
p#p01
script
index.html
Source History
9 <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 Please input a number between 5 and 10:
13 <input id="demo" type="text">
14 <button type="button" onclick="myFunction()">Test Input</button>
15
16 <script>
CSS Styles
JavaScript try...catch.....
http://localhost:8383/JS_Tre_Catch_Finally/index.html
100%
Please input a number between 5 and 10:
Test Input
Input is too high
Variables Call Stack Breakpoints
Name Type Value
<Enter new watch>
39:1 INS

Review

- In this tutorial we have extended our introduction to arrays and have:
  - Introduced reversing arrays
  - Reviewed expressions and operators and introduced comparison operators, conditional operators, and logical operators
  - Introduced comparing different types and boolean values
  - Introduced the ternary conditional operator
  - Considered events and event handling
  - Considered how runtime errors may be addressed