Introduction to JavaScript

Brian Dickinson

University of Rochester

Fall 2019
Java and JavaScript

- Java and JavaScript are completely unrelated \(^1\)
  - Java is to JavaScript as car is to carpet
  - Java is to JavaScript as ham is to hamster
- JavaScript may have chosen its name to benefit from Java’s popularity

\(^1\)except that they are both programming languages
What is JavaScript?

- Interpreted Scripting Language
- Dynamically Typed
- Weakly Typed
- Originally Client-Side

\(^2\) now often server side as well using node.js
Dynamic vs Static Typing

<table>
<thead>
<tr>
<th>Static</th>
<th>explicit declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>checked at compile time</td>
</tr>
<tr>
<td>Dynamic</td>
<td>implicit declaration</td>
</tr>
<tr>
<td></td>
<td>checked at run time</td>
</tr>
</tbody>
</table>

Java

Double grade = 93.4;
grade = "A";

exit status 1
Main.java:5: error: incompatible types:
    String cannot be converted to Double

JavaScript

var a = 93.4;
a = "A";
Javascript Primitive Types

- string
- number
- boolean
- null
- undefined
<table>
<thead>
<tr>
<th>Operator</th>
<th>Operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
<td>$3 + 4 \rightarrow 7$</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td>$3 - 4 \rightarrow -1$</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>$3 \times 4 \rightarrow 12$</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td>$3/4 \rightarrow 0.75$</td>
</tr>
<tr>
<td>**</td>
<td>Exponentiation</td>
<td>$3 \times 4 \rightarrow 81$</td>
</tr>
<tr>
<td>%</td>
<td>Modulus</td>
<td>$3%4 \rightarrow 3$</td>
</tr>
</tbody>
</table>
## Assignment Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Equivalent Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>( x = y )</td>
</tr>
<tr>
<td>+=</td>
<td>( x += y )</td>
</tr>
<tr>
<td>-=</td>
<td>( x -= y )</td>
</tr>
<tr>
<td>*=</td>
<td>( x *= y )</td>
</tr>
<tr>
<td>/=</td>
<td>( x /= y )</td>
</tr>
<tr>
<td>**=</td>
<td>( x **= y )</td>
</tr>
<tr>
<td>%=</td>
<td>( x %= y )</td>
</tr>
</tbody>
</table>
# Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>==</code></td>
<td>equal to</td>
<td><code>10 == &quot;10&quot; → true</code></td>
</tr>
<tr>
<td><code>===</code></td>
<td>equal to and same type</td>
<td><code>10 == 10 → false</code></td>
</tr>
<tr>
<td><code>!=</code></td>
<td>not equal</td>
<td><code>8 != &quot;8&quot; → false</code></td>
</tr>
<tr>
<td><code>!==</code></td>
<td>not equal or not same type</td>
<td><code>8 !== &quot;8&quot; → true</code></td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>greater than</td>
<td><code>7 &gt; 8 → false</code></td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>less than</td>
<td>&quot;A&quot; &lt; &quot;B&quot; → true</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>greater than or equal to</td>
<td><code>8 &gt;= 8 → true</code></td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>less than or equal to</td>
<td>&quot;B&quot; &lt;= &quot;A&quot; → false</td>
</tr>
</tbody>
</table>
## Logical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;&amp;</code></td>
<td>logical and</td>
</tr>
<tr>
<td>`</td>
<td></td>
</tr>
<tr>
<td><code>!</code></td>
<td>logical not</td>
</tr>
</tbody>
</table>
JavaScript Variables

```javascript
var firstName = "Brian";
var grade = 93;
var passing = true;
var placeHolder;
var attendance = [true, true, false, true];
```
**Variable Declarations**

- **var** - generic function or global scope variable declaration
- **let** - block scope variable declaration
- **const** - block scope variable declaration that cannot be reassigned

```javascript
const numStudents = 45;
var grade = 87;
{
    let grade = 53;
    console.log(grade);
}
console.log(grade)
numStudents = 44;
```

```
53
87
TypeError: Assignment to constant variable.
```
Objects

```javascript
var user = {
    firstName: "Brian",
    lastName: "Dickinson",
    userName: "bdicken3",
    email: "bdicken3@cs.rochester.edu"
};

console.log(user.firstName.length)
```

5
var letter_grade;
if (numerical_grade >= 90) {
    letter_grade = "A";
} else if (numerical_grade >= 80) {
    letter_grade = "B";
} else if (numerical_grade >= 70) {
    letter_grade = "C";
} else if (numerical_grade >= 60) {
    letter_grade = "D";
} else {
    letter_grade = "F";
}
var letter_grade;
switch (Math.floor(numerical_grade/10)) {
    case 10:
    case 9:
        letter_grade = "A";
        break;
    case 8:
        letter_grade = "B";
        break;
    case 7:
        letter_grade = "C";
        break;
    case 6:
        letter_grade = "D";
        break;
    default:
        letter_grade = "F";
}
While and Do While Loops

```javascript
var i = 0
while (i < 5) {
    console.log(i);
    i++;
}
```

```javascript
var i = 0;
do {
    console.log(i);
    i++;
} while (i < 5);
```
For and For In Loops

```javascript
for (i = 0; i < 5; i++) {
    console.log(i);
}

example = {"Hello": 0, "World": 1};
for (var k in example){
    console.log(k, example[k]);
}
```
function add(a, b) {
    return a+b;
}

var c = add(2,40);
console.log(c);
Anonymous Functions

- unnamed functions that are only used in one place
- most commonly attached as events

```javascript
document.getElementById("someButton").onclick = function()
{
    alert("You clicked the button!");
};
```

\(^3\) if the code is going to be reused you should create a named function
Where to Include JavaScript Files

- External Reference (usually best practice)
- Document Embedded (sometimes best option)
- Element Embedded (rarely a good idea)
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="script.js"></script>
  </head>
  
  . . .
</html>
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript">
        alert("Hello World");
    </script>
</head>
...
<!DOCTYPE html>
<html>

... 
</body>

<button onclick="alert(Date());">
  What time is it?
</button>

</body>

</html>
Document Object Model (DOM)

- DOM is a programming interface to modify page
  - content
  - structure
  - style
- DOM may be modified using JavaScript
DOM Traversal in JavaScript

// Direct Element Access Methods
var body = document.body;
var button = document.getElementById("demoButton");
var paragraphs = document.getElementsByTagName("p");
var cities = document.getElementsByClassName("city");

// DOM Tree Traversal Attributes
var buttonParent = button.parentNode;
var children = body.childNodes;
var firstChild = body.firstChild;
var lastChild = body.lastChild;
var secondChild = firstChild.nextSibling;
var penultimateChild = lastChild.previousSibling;
DOM Tree Traversal Reference

- **Root element:** `<html>`
  - **parentNode**
  - **firstChild**
    - **Element:** `<head>`
      - **previousSibling**
      - **nextSibling**
        - **lastChild**
          - **Element:** `<body>`
    - **childNodes** to `<html>`, and siblings to each other
// use link’s id to find it in the DOM
var link = document.getElementById("rocLink");

// modifying the link’s attributes
link.innerHTML = "URCS"; // change element contents
link.href = "https://cs.rochester.edu"; // change attribute
link.style.color = "red";
DOM Element Creation and Deletion

// first we create a new button element
var btn = document.createElement("button");

// then we modify this element to set desired attributes
btn.innerHTML = "Get Date/Time";
btn.onclick = function(){alert(Date())};

// next we use this button to replace the old one
var old = document.getElementById("dateButton");
var prnt = old.parentNode;
prnt.removeChild(old);
prnt.appendChild(btn);

// alternatively we could have used
prnt.replaceChild(btn, old);
Bibliography I

- **JSLint.**
  - Jslint.

- **Premshree Pillai.**
  - Introduction to static and dynamic typing.

- **programminghub.io.**
  - The truth about java and javascript.
repl.it.
Javascript repl.
https://repl.it/languages/javascript.

w3c.
Javascript best practices.
https://www.w3.org/wiki/JavaScript_best_practices.

w3schools.
Javascript - html dom methods.
https://www.w3schools.com/js/js_htmlldom_methods.asp.
w3schools.
Javascript best practices.
https://www.w3schools.com/js/js_best_practices.asp.

w3schools.
Javascript const.
https://www.w3schools.com/js/js_const.asp.

w3schools.
Javascript events.
https://www.w3schools.com/js/js_events.asp.
w3schools.
Javascript for loop.
https://www.w3schools.com/js/js_loop_for.asp.

w3schools.
Javascript html dom document.
https://www.w3schools.com/js/js_htmldom_document.asp.

w3schools.
Javascript html dom navigation.
https://www.w3schools.com/js/js_htmldom_navigation.asp.
w3schools. Javascript let.
https://www.w3schools.com/js/js_let.asp.

w3schools. Javascript objects.
https://www.w3schools.com/js/js_object_definition.asp.

w3schools. Javascript tutorial.
https://www.w3schools.com/js/default.asp.
Bibliography VI

w3schools.
Javascript while loop.
https://www.w3schools.com/js/js_loop_while.asp.