

Introduction to JavaScript

CSE 331

University of Washington

Michael Ernst

How to create a GUI for your Java application

According to Sun/Oracle:

- 1995-1998: AWT (interface to native system GUI)
- 1998-2008: Swing (pure Java GUI)
- 2008-2018: JavaFX (supports HTML5 features)
- 2018- : 🙄(ツ)🙄

According to Java developers:

- HTML5 = HTML + JavaScript
- 57% of *all* Java apps use JavaScript (Java Magazine, Nov/Dec 2018)

Java and JavaScript



Oak (1994) \Rightarrow Java (1995)

- Designed for Internet of Things (IoT)
- Success when integrated with browser
- Then used as a general-purpose programming language

Mocha (10 days in 1995) \Rightarrow LiveScript \Rightarrow JavaScript (1996)

- Write code directly in a web page
- No separate file, no compilation step
- Integration with HTML
- Named to piggyback on Java's popularity

A web page = HTML + CSS + JavaScript

- HTML: Contents and structure of the web page
 - Headings
 - Text
 - Images
- CSS: Visual appearance
- JavaScript: interactive behavior
 - Dynamically modify an HTML page
 - React to user input
 - Validate user input
 - Create cookies
 - Communicate with a server (written in any language)
 - JavaScript user interaction does not require any communication with the server

Learning a new programming language

- Important skill
- Learn a small subset of JavaScript
- More relevant than learning Swing

To learn a new programming language:

- Syntax
- Libraries
- Language concepts

Syntax

- Expression and statement syntax is similar to Java
 - Arithmetic
 - for, while, if, ...
 - Define and call functions
- Some syntax is nicer; see next slide

Data structures

Arrays (lists):

```
let students = ["Harry", "Hermione", "Ron"];
students[1]
students.length
students[9] = "Neville" creates undefined elements at indices 3..8
```

Dictionaries (maps):

```
const teams = {instinct: yellow, mystic: red, valor: blue};
teams["instinct"]
teams.instinct
teams["silliness"] = 'polkadot'
```

Sets

- Use a dict (how Java implements HashSet internally)
- `const colors = new Set(['red', 'green', 'blue']);`
Methods: add, has, delete, clear, size

Dynamic typing = run-time checks

```
let x = 42;  
... x++ ...  
x = 'hello';  
...
```

```
const myNumber = 22;  
const myString = "hello";  
myNumber + myString
```

No compiler or type-checker

- Easy to develop fast!
- Easy to make mistakes! (run-time crash or odd behavior)
 - recommendation: `"use strict";`
 - recommendation: use a linter like JSHint

Null vs. undefined

The Java compiler requires that all variables are initialized. JavaScript does not.

- Undefined = the variable has not yet been assigned
 - Maybe it hasn't even been declared
- Null = the variable has been assigned to the `null` value

Automatic conversion to boolean

```
if (expr) { ... } else { ... }
```

- Treated as false if `expr` evaluates to:
false, 0, empty string (""), NaN, null, or undefined
- Treated as true otherwise

Objects

```
const greatestClass = {department:"CSE", number:331}
greatestClass.number => 331
greatestClass.professor = "Ernst";
delete greatestClass.department;
```

```
const greatestClass = {
  department: "CSE",
  number: 331,
  toString: function() {
    return this.department + this.number;
  }
}
```

Objects only, no classes (ECMAScript 6 has classes)

- No class declarations
- Add fields and methods dynamically
- Clone a *prototype* object to share behavior

The browser renders the DOM, not HTML

- DOM = Document Object Model
- Initially: same as HTML
- JavaScript can change it

