# General Info

- TypeScript
  - Superset of JavaScript
  - Adds features intended for ECMAScript 6
  - Can be transcompiled into JavaScript for use in current browsers
- HHVM – HipHop Virtual Machine
  - Develop by Facebook
  - Virtual machine that executes programs written in Hack and PHP
  - Just in time compiler than converts PHP syntax to machine code
  - HHVM → https://en.wikipedia.org/wiki/HipHop_Virtual_Machine
  - Hack → http://hacklang.org/
- Beautifier
  - http://jsbeautifier.org/

# One-Dimensional Arrays

- **Array** → Collection of values that can be treated as a unit or individually.

  var a = new Array(4);

- **Indexing** → We access an element using [ ]

  First element associated with index 0 (e.g., a[0])

- An element of an array can be of any type and and an array can hold different types of elements
- The length property represents the length of the array (e.g., a.length)
- We can print the contents of an array by using alert

# Definition of One-Dim Arrays

- Via array literal → comma separated list of elements within square brackets

```
var a = [2, 3, 5];
var b = []; // empty array
```

- Specified in the Array constructor

```
var c = new Array();
var e = new Array(4);           // defines array of size 4
```

- **Example:** ArraysOneDim.html

# Two-Dimensional Arrays

- JavaScript does not support actual two-dimensional arrays
- You can simulate two-dimensional arrays by using an array of arrays
- About two-dimensional arrays
  - You can pass them and return them from functions like one-dimensional arrays
  - Any modifications in the function will be permanent
  - You can have ragged arrays
- **Example:** ArraysTwoDim.html

# Functions as Data

- JavaScript functions are objects that can be passed and return from functions
- **Example:** FunctionsAsData*.html

# Random Values

- Example: RandomValues.html

# Numeric Values

- Example: NumericValues.html

# Null and Undefined

- **null →** indicates no value
- undefined
  - Value associated with uninitialized variables
    - var x; // in a function
  - When a function that is expected to return a value does not returns one (IMPORTANT case)
  - Value associated with object properties that do not exist
- == considers **null** and **undefined** equal
- === considers **null** and **undefined** different

# NaN

- **NaN** → Not-A-Number (Same as Number.NaN)
  - Unequal to any number including itself
  - Use **isNaN** function → determines (returns true or false) whether an argument is not a number. It attempts to convert the argument to a number
- The following comparisons return false

  NaN == NaN, NaN === NaN
- Remember → !isNaN() allow us to determine whether an expression is a number
  - Notice: isNaN(20) → False
  - You may want to write a function call isNumber that returns !isNaN(x)
- **Example:** NaN.html

# About prompt

- Returns null when cancel is selected
- **Example:** Null.html, ValidityCheck.html

# Debugging/General Info

- http://www.cs.umd.edu/~nelson/classes/utilities/JavaScript/JavaScriptDebugging/