Elements of Programming with JavaScript

CMSC 122

What is a Computer Program?

- A finite sequence of instructions that solves a particular kind of problem.
- Ask: what kinds of problems can be solved by a computer?
  - Well-defined criteria for success.
  - Expressible (explainable) to the computer—meaning that the problem can be represented in some formal language.
  - Solution can be obtained in an “acceptable” amount of time and space.
  - These kinds of questions are at the foundation of Computer Science.
- Computer programs are machine-executable interpretations of algorithms.

How do we express programs?

- Primarily linguistic, meaning we need a language usable by human beings
- A language that is then consistently translated into a language suitable to a particular kind of machine (a kind of mathematical model).
- Modern programming languages based on John von Neumann's stored-program model as well as mathematical-logical abstractions, Turing, Church, and many others.

What kinds of languages are used today?

Literally, we have hundreds of languages . . . why?

- So-called "object-oriented" languages
- "Scripting" versus "compiled" languages.

But, all languages have certain things in common:

- Require formal grammars.
- Require underlying "semantics"

All languages are *equally* powerful as far as the kinds of problems they solve.

JavaScript is a scripting language

- Usually scripting languages are “interpretative,” meaning interactive: read, eval, print.
- No so with JavaScript.
- Browser is both the source of the data and events that drive the evaluation and it is also the container where the results of these scripts are displayed.
- For those who have experience with programming languages . . . JavaScript is *not* related to Java!

Programs embody algorithms, so . . .

Understanding programming requires that we understand algorithms. . . .

- Assume that we have a well-defined problem . . .
- Ask what kinds of "moves" can we make in solving the problem, meaning:
    - What kinds of "transformations" can be expressed?
    - What kinds of behaviors can our program express?
- Ask: how do we determine success or failure? How do we know when we're done?

An example

Give an algorithm for “brushing one’s teeth.”

- Ask: what are the “moves?”
- How can these moves be arranged?
- How do we know if/when we’re done, or that we have somehow failed?

Outline this algorithm in class, identifying its components.

Four basic moves

Concurrent	: Actions that have *no* obvious “sequence,” meaning that the outcome is not dependent upon the order in which they are performed.
Sequence	: Actions that must be done in order.
Conditional	: Actions that are taken in response to some state of things or a change in that state.
Repetitive	: Actions that are repeated.

Programming Languages implement the moves

- We will systematically tie the programming constructions that we study to these “basic moves.”
- These basic moves are combined by mathematical and logical principles into larger, more complex entities.
  - Mathematics describes the “structure” of each object and how they are combined to create more complex objects and behaviors.
  - Logic gives us the laws of consistent thinking as expressed within some language; logic ensures that the structures and behaviors that we create are correct and consistent.