COMP302 Lecture 2 07 September 2016

➔ trade-offs in various features
- they all do the same thing !
  - same expressiveness, no language more powerful, though some better for specific uses
- differences in terms of usability : how easy they are for us to use, and to express complex ideas
  - 3 features :
    - Sequence things together "one thing happens after the other"
    - Conditionals "test whether things are true, or make decisions"
    - Looping "iteration, recursion. Repeating things, an arbitrary number of times"
  - Plus DATA "0" – baseline
    - defining successors or predecessors
- appropriate abstraction

With that I can make a language

---

Languages in this course

- JavaScript: main language. Common, useful. Technically a scripting language. Functional language.
  - Drawbacks: peculiar syntax/semantics choices. Dynamic typing (though convenient when making a quick program, it encourages you to be less attentive).
- Other functional languages:
  - LISP/Scheme. LISP is a bit like the classical functional language.
- Mediawiki-style "template

---

JavaScript

Scripting. OO-language. Name is bad cause reminds of Java but unrelated.
- "LiveScript" (1995) –> renamed JavaScript to use Java's popularity

Meant to be used inside browser, added to HTML → web enhancement language. Now has grown to be more than that.
Also called ECMAScript because it was standardized.
- Evolving standard.

Browser variations
- ECMAScript 5-6
  - Firefox does 5.1+
  - Chrome does 6
  - Safari does 6
  - Edge does 5.1
  - Opera 5?

JS → interpreted language. In source code. Traditional way of executing language:

Program → Compiler → Machine code(.exe)

JS does :

JS source → Browser

- JS engine
  - interprets, JIT (just in time compiler)

(((Another way is node.js. Command line version of it. JS engine taken out and made a standalone program. )))

Embedded: in a web-page *<script> ..... </script>*. Sometimes not convenient (for example if you want to update the code independent of the webpage.
Other way to avoid that is *<script src = "source.js"> </script>*.

We will mostly use JS for computation, as a functional language.

Will not be used in class

- API for manipulating the webpage itself structure (accessing and manipulating). Known as DOM. Tree light representation of the entire webpage structure. Change/copy/delete elements.
- Lots of libraries, for doing different tasks. Might make tasks easier. One very well known: jQuery.



To declare variables:
*var f;* or *var f = 1;* var has scope within entire function, not just the block
*let f* has scope only within its block.

Types

- Numbers: 64-bit floats (no integer types). All base 10 by default.
- Booleans: true/false.
- Strings: *"abc"* or *'abc'*. Escape characters *'abc\n'*.
  - Immutable, cannot change its value once it's declared. You can create strings from other ones. Can concatenate.
    - *"abc".charAt(0) → "a".*
    - *"hello".substr(2) → "llo".*

TYPE CONVERSION
*5+3 → 8*
*"5" + "3" → "53"*
*"5" + 3 → "53" due JS trying to unify the types*
*"""" + 3 → "3"*
*"3" + 0 → "30"*
*"3" – 0 → 3*
*String(3) → "3"*
*Number("3") → 3*
"" → false
false → "false"
"..." → true
true → "true"
"" → false → "false" → true (need to be careful)

COMPLEX DATA

Objects

- Associative arrays. Key-value pair mappings. We can manipulate, change the keys, etc.
- Arrays are a type of Objects, their keys are just a string of integers.