Numbers in Javascript

Relevant links

- Flanagan's book, section 3.1.
- MDN's reference for Numbers[1]
- MDN's reference for the Math object[2]
- IEEE 754 standard[3] (optional)
- The group working on the standard[4] (optional)
- Wikipedia page on IEEE 754 standard[5] (optional)

Numbers in Javascript

- Unlike most languages, in Javascript there is only one type of numbers, namely 64-bit floating point numbers defined in the IEEE 754 standard.
- Certain operations (e.g. bitwise) are performed treating the numbers as 32-bit integers instead.
- Number literals are much like in other languages. You can use `0x` for hexadecimals.
- Avoid using octal literals (those starting with a 0).
- The `Math` object contains a number of useful functions. Refer to the MDN documentation for these functions for details. You should make sure you become familiar with the functions for the following (avoid "experimental" functions):
  - Raising to a power
  - Maximum/Minimum
  - Random numbers
  - Various rounding
  - Logarithm and exponential
  - Trigonometric functions
- The `Math` object also contains a number of mathematical constants (pi, e).
- The `Number` object contains constants for the largest integer, as well as `Infinity` and `NaN` (not a number).
- Arithmetic operations never raise errors, instead producing either `Infinity`, `-Infinity` or `NaN`.
- `NaN` in particular is a very unusual value. It represents a numeric value that is indeterminate.
  - So it *is* a number, even though it stands for "not a number".
  - You can see that by typing `typeof NaN`.

[1] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number
[2] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math
[3] http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4610933
[4] http://grouper.ieee.org/groups/754/
[5] http://en.wikipedia.org/wiki/IEEE_floating_point

  - NaN is not equal to any value, including itself.
  - You can see that by trying NaN == NaN.

- Some behavior to watch out for (try to explain it): Number.MAX_VALUE == Number.MAX_VALUE + 1
- Make sure to read about rounding errors, that are unavoidable whenever you do floating point arithmetic.