Transpilation

- I am careful to choose notation that I can easily transpile
  - for example, inheriting the year can be handled by a simple textual rewrite (1. every time we encounter a *year* variable, push it onto a stack, 2. every time we encounter a *mm-dd* date, rewrite it as yyyy-mm-dd using the top of the stack)
  - [N.B. it might be fruitful to create many stacks, one for each kind of inherited item, e.g. a year-stack, a statement-date stack, a from-account-stack, etc. Or, one can use dynamic scoping (as per early Lisps) …]
  - [N. B. stacks —> concatenative language design]

Objects and Bags

- I will use ASON Objects { $name_1$: … $name_2$: … …} to represent named entities
- I will use ASON Arrays as bags [… … …]
  - the concept of Array is too low-level and implementation-oriented
  - I don’t care if the items are indexed
  - The main feature I want: collection of unnamed items
- Bags contain any kind of data recursively, e.g. bags can contain constants, bags, objects
  - e.g. [1 2 3]
  - e.g. [1 {name: 2} 3]
  - e.g. [1 [2 3 {$name_1$: 4} {$name_2$: 5} 6] 7 8 9]