Crox 语言规范
版本（日期）：12/18/2013 10:31 AM

数据类型：

JSON，但不支持 *数组字面量*、*对象字面量*、null

词法：

保留字：（js 的 *ReservedWord*、*FutureReservedWord* 和 null）abstract boolean break byte case catch char class const continue debugger default delete do double else enum export extends final finally float for function goto if implements import in instanceof int interface let long native new package private protected public return short static super switch synchronized this throw throws transient try typeof var void volatile while with yield null
标识符(id)：以字母或下划线开头，后面 0 或多个字母、数字、下划线。但不能是保留字、布尔。root 是一个特殊的标识符，表示数据。
字符串(string)：/"(?:[^"\\]|\\[\s\S])*"|'(?:[^'\\]|\\[\s\S])*'/
数值(number)：/\d+(?:\.\d+)?(?:e-?\d+)?/
布尔(boolean)：true false
空白：/\s+/ 无意义
文本(text)：任意字符序列，但不能包含 {{，并且，从倒数第 1 个字符向后看，不能是 {{
符号：! % && ( ) * + - . / < <= = > >= [ ] || === !==
{{ }} {{{ }}} {{#if {{#each {{/if}} {{/each}} {{else}}

语法：
用 BNF 表示，参考：巴科斯范式
http://zh.wikipedia.org/wiki/%E5%B7%B4%E7%A7%91%E6%96%AF%E8%8C%83%E5%BC%8F

```
#start program;
program:
    statements
;
statements:

|   statements statement
;
statement:
    "{{#if" expr "}}" statements "{{/if}}"
|   "{{#if" expr "}}" statements "{{else}}" statements "{{/if}}"
|   "{{#each" expr string string? "}}" statements "{{/each}}"
|   "{{" set id "=" expr "}}"
|   "{{" expr "}}"
|   "{{{" expr "}}}"
```

```
|   text
|   "{{" include string "}}"
;
id:
    realId
|   set
;
PrimaryExpression:
    string
|   number
|   boolean
|   id
|   "(" expr ")"
;
MemberExpression:
    PrimaryExpression
|   MemberExpression "." id
|   MemberExpression "[" expr "]"
;
UnaryExpression:
    MemberExpression
|   "!" UnaryExpression
|   "-" UnaryExpression
;
MultiplicativeExpression:
    UnaryExpression
|   MultiplicativeExpression "*" UnaryExpression
|   MultiplicativeExpression "/" UnaryExpression
|   MultiplicativeExpression "%" UnaryExpression
;
AdditiveExpression:
    MultiplicativeExpression
|   AdditiveExpression "+" MultiplicativeExpression
|   AdditiveExpression "-" MultiplicativeExpression
;
RelationalExpression:
    AdditiveExpression
|   RelationalExpression "<" AdditiveExpression
|   RelationalExpression ">" AdditiveExpression
|   RelationalExpression "<=" AdditiveExpression
|   RelationalExpression ">=" AdditiveExpression
;
EqualityExpression:
    RelationalExpression
```

```
|   EqualityExpression eq RelationalExpression
|   EqualityExpression ne RelationalExpression
;
LogicalAndExpression:
    EqualityExpression
|   LogicalAndExpression "&&" EqualityExpression
;
LogicalOrExpression:
    LogicalAndExpression
|   LogicalOrExpression "||" LogicalAndExpression
;
expr:
    LogicalOrExpression
;
```

其中：不出现在产生式头部的 *名字*，都是 *终结符*。

语义：

按照 js 语义（包括字面量取值、运算符优先级、运算规则等）
其他，说明如下：

```
statement: "{{#each" expr string string? "}}" statements "{{/each}}";
```

其中，两个 `string` 会声明一个名字为其值的变量，第一个表示 值，第二个表示 索引（可选）。

```
statement: "{{" set id "=" expr "}}";
```

该语句会声明一个变量 id，并赋值。

```
statement: "{{" expr "}}";
```

输出 html 转义的 expr 的值。

```
statement: "{{{" expr "}}}";
```

直接输出 expr 的值。

```
statement: "{{" include string "}}"
```

include 是基于文件的，其中的 string 是个路径字符串。