C Language Reference Manual

Stripped Language: AWK and LUA
Support Data Types: Double, Int, String, Set
Strongly Typed
Types inferred: 0.0, 0, “”, []
Automatic Coercion: int -> double
Operators: + - * / == = && ll []
string + string; string + int/double/set
highest precedence; returns string
set + set: concat sets
double + double: add
int + int: add
set[#] for access
Identifiers: name that is not reserved
Control Flow

```
if(expr){/*expr = 1 true, expr = 0*/

}else if(expr){

}else{

}
while(expr){

}
```

Expressions: id = expression;
print “”
General Structure

```
open “” /*Optionally opens file*/
{/*Begin*/}
[/*Pattern*/]{/*Action*/}
        [/*Subpattern*/]{/*Subaction*/}
}
[/*Pattern*/]{/*Action*/}
        [/*Subpattern*/]{/*Subaction*/}
}
{/*End*/}
```

Begin scope continues for entire pattern
Pattern scope continues for rest of program
End scope begins at start of end block and ends at closing parentheses

Functions: think of a named pattern

```
name [/*params*/]{
        return 0; //default return type
}
```

Defined in Begin and End Blocks
Patterns
[“regex”]
[“css selectors”]
Line by Line?
Access: this[#]

Open Questions
How to we handle multiple files?
Do we want to provide import statements?
What do we compile to? I suggest Java byte code since we know Java, there are good frameworks for parsing regex and xml, and it will be a lot faster than python.

Timeline

1. Determine target language (Mon)
2. Resolve open questions (Mon)
3. Fix syntax/Add + remove conventions (Mon)
4. Language Reference Manual Split (2 weeks)
    1. Introduction + Syntax
    2. Expressions
    3. Declarations
    4. Grammar