

## 2024-04-04 RT - Recursive Text

### RT

I began thinking about an über-syntax - an “IR” if you wish - that I was tentatively calling “RT” (Recursive Text).

I haven’t settled on any yet, but, I currently think that it will take the form:

«*operator operand operand...*»

Where the brackets are hard-chosen to be Unicode, to leave valuable ASCII characters alone. Unicode has so many characters that we can afford to choose some and assign fixed meanings to them, without using up valuable character-space real-estate.

Note that Ohm (PEG) allows parsing arbitrary strings of characters (as long as there is something unique in the strings, to differentiate them), so we might write stuff like:

«I want a *hamburger with 2 patties and ketchup and mustard*»

Where italics are use to represent *operands*. The above phrase might be transpiled into:

kitchen.order (burger, 2 \* patty, [ketchup, mustard])

Which might be further reduced to something like:

(order kitchen ('burger (\* 2 'patty (list 'ketchup 'mustard))))

The fact that OhmJS can be used to parse XML / HMTL / etc. might gives us syntax like:

**I want a** *hamburger with 2 patties* **and** *ketchup* **and** *mustard*

which looks - in machine-readable text form - as:

<mark>I want a</mark> <i>hamburger with 2 patties</i> <mark>and</mark>  
<I>ketchup</i> <mark>and</mark> <I>mustard</i>

And, of course, we could parse even more interesting syntaxes composed only of HTML (and CSS) or project-specific syntaxes based on XML / HTML / etc.

## Appendix - See Also

### **See Also**

**References** <https://guitarvydas.github.io/2024/01/06/References.html>

**Blog** <https://guitarvydas.github.io/>

**Blog** <https://publish.obsidian.md/programmingsimplicity>

**Videos** <https://www.youtube.com/@programmingsimplicity2980>

[see playlist “programming simplicity”]

**Discord** <https://discord.gg/Jjx62ypR> (Everyone welcome to join)

**X (Twitter)** @paul\_tarvydas

**More writing (WIP):** <https://leanpub.com/u/paul-tarvydas>