

BioBIKE Language Syntax

General Consideration of Syntax

Evaluation and Variables

A. Evaluation process

When the BBL web-listener is handed over code, it goes through the following thought processes:

1. Overall syntax check:
 - a. Do all parentheses occur as matching pairs?^{*}
Example (valid): (SEQUENCE-OF all4312 FROM x TO (+ x 100))
Example (invalid): (SEQUENCE-OF all4312
 - b. Does each open [have a corresponding close]?
Example (valid): *all-organisms*[1 -> 3]
Example (invalid): *all-organisms*[1
 - c. Does each open { have a corresponding close }
Example (valid): {{A7120 all4312} {Npun npf0001}}
Example (invalid): {{A7120 all4312} {Npun npf0001}
2. Extract the first form and ignore the rest
 - a. If the first non-space symbol is an open parenthesis, read until the corresponding close parenthesis and ignore the rest.
Example: (DEFINE x AS 7) (DISPLAY (+ x 1))
|<-extracted->| |<---ignored--->|
 - b. Otherwise read until the end of the symbol
Example: 1 + 1
|<-ignored...>
3. Evaluate all symbols
 - a. Numbers, and literal strings and symbols are evaluated to themselves
Example: "x" evaluates to "x"
Example: 47 evaluates to 47
 - b. Other symbols are evaluated as variables or as functions, depending on their position
Example: (+ 1 x) x is interpreted as the **variable** called x
Example: (x + 1) x is interpreted as the **function** called x
 - c. If a symbol is deemed to be a variable or a function and that variable or function has not been defined, an error is raised
Example: (+ 1 x) ERROR, unless x previously defined as a **variable**
Example: (x + 1) ERROR, unless x previously defined as a **function**

^{*} If the line ends in at least one close parenthesis, the BBL-reader will add or subtract as many parentheses necessary to make parentheses match. This is **not** done with square and curly brackets.

4. Evaluate functions

- a. Start with logically most inner functions and work outwards

Example: `(+ 1 (COUNT-OF "A" IN (CHROMOSOME-OF A7120)))`
|<- evaluate 1st -->|

- b. Check syntax of function, raising error if syntax faulty

Example: `(CHROMOSOME-OF 47)` ERROR: argument can't be a number

- c. Replace function with object returned by function

Example: `(+ 1 (COUNT-OF "A" IN (CHROMOSOME-OF A7120)))`
--> `(+ 1 (COUNT-OF "A" IN A7120.chromosome))`
--> `(+ 1 1878490)`
--> `1878491`

It is important to see from this sequence of events that:

BioBIKE examines only one form at a time

That one form may be a single number or dozens of lines of code within matching parentheses, but learn this lesson well. A common error amongst new users is to expect things like this to work:

```
(DEFINE x AS 47)
(- x 18)
```

Users wonder why they never see 29 nor any error message or warning. But you now know why.

B. Defining variables

Variables at minimum help organize your thoughts. At maximum they allow you to do in general what you could not hope to do in a lifetime of specific instructions.

Variables may be defined explicitly with `DEFINE` or implicitly with `ASSIGN` or `INCREMENT`. If an attempt is made to `ASSIGN` or `INCREMENT` a variable that has not previously been defined, then it is created on the spot. All of these functions returns the value it ends up assigning.

Examples:

```
<1>> (DEFINE x AS 47)
:: 47
[The variable x is defined and given the value 47]
<2>> (DEFINE my-favorite-genes AS {all4312 npf0001})
:: (#\$A7120.all4312 #\$Npun.NpF0001)
[The variable my-favorite-genes is defined and given the value a list consisting
of two genes]
<3>> (DEFINE glnA-genes AS
      (ORTHOLOGS-OF (GENE-DESCRIBED-BY "glnA" IN A7120)))
:: (#\$S7942.ser0686 #\$P9313.PMT0601 #\$S6803.slr1756 #\$Npun.NpR5387 . . .)
[The variable glnA-genes is defined and given the value a list consisting of all
genes orthologous (similar in a way we'll no doubt discuss later) to a gene of
Anabaena PCC 7120 that has the name "glnA"]
```

To illustrate how defining variables helps to organize your thoughts and make it easier to see later what you did, consider this situation:

You want to find out what genes, if any, are peculiar to those cyanobacteria that live in the ocean. Perhaps these will include genes to resist the high concentration of salt, etc. Then you want to download these marine-specific genes, along with a descriptive phrase.

Here's one way to do it:

```
<1>> *all-organisms*
:: (#\$synechococcus\_elongatus\_pcc7942 #\$prochlorococcus\_marinus\_mit9313
   #\$synechocystis\_pcc6803 #\$nostoc\_punctiforme\_atcc29133 . . .)
   Entering the name of an object asks BioBIKE to return the value of the object. Here it gives us a list of all the organisms known to the system.

<2>> (ITEMS .habitat IN-EACH-OF *all-organisms*)
:: ("fresh water" "marine" "fresh water" "terrestrial" . . .)
   ITEMS is used to go inside of informational structures. Here it asks for the retrieval from every organism the information stored under the category "habitat".

<3>> (DEFINE marine-organisms AS *all-organisms*[2 6 7 8 11 12])
:: (#\$prochlorococcus\_marinus\_mit9313 #\$trichodesmium\_erythraeum . . .)
   The variable marine-organisms is defined as the 2nd, 6th, 7th, 8th, 11th, and 12th organism within the list of all organisms known to the system.

<4>> (DEFINE non-marine-organisms AS
      (SET-DIFFERENCE *all-organisms* marine-organisms))
:: (#\$anabaena\_variabilis\_atcc29413 #\$thermosynechococcus\_elongatus\_bp1...)
   The variable non-marine-organisms is defined as what's left when you remove the organisms of marine-organisms from the set of all available organisms.

<5>> (DEFINE marine-genes AS
      (COMMON-ORTHOLOGS-OF marine-organisms NOT-IN non-marine-organisms))
:: (#\$P9313.p-PMT1856 #\$P9313.p-PMT0643 #\$P9313.p-PMT0605
   #\$P9313.p-PMT0340 #\$P9313.p-PMT0339 #\$P9313.p-PMT0268)
   The variable marine-genes is defined as the set of all genes common to marine organisms not found in non-marine organisms.

<6>> (DEFINE marine-genes-info AS
      (INTERLEAVE marine-genes (DESCRIPTIONS-OF marine-genes)))
:: ((#\$P9313.p-PMT1856 "possible Photosystem II reacti")
   (#\$P9313.p-PMT0643 "\"Integral membrane protein, DU") . . .)
   The variable marine-genes-info is defined a set of paired values -- the name of a gene and its description -- for each gene in the set called marine-genes.

<7>> (WRITE-TAB-DELIMITED-FILE "marine-genes-info.txt" marine-genes-info)
   The list of paired values in marine-genes-info is written out in tab-delimited format (readable by Excel) into a file called "marine-genes-info.txt".
```

You *could* have accomplished the same end without defining any variables, but I'll bet you'd never be able to understand the code a week later.