| Group | Module | Test Code | Accompanying files | About |
|---|---|---|---|---|
| utils | between_with_step.pl | between_with_step.plt | TLDR_between_with_step.txt | Like between/3, but takes a (possibly negative) step value |
| | between_x.pl | between_x.plt | TLDR_between_x.txt | A between/3 which accepts unbound variables as Arg1 and Arg2 and generates possible intervals [Arg1,Args] in which Arg3 lies on backtracking |
| | clashfree_id_selection.pl | clashfree_id_selection.plt | | |
| | difflist_length.pl | difflist_length.plt | | |
| | in_prefix.pl | in_prefix.plt | | |
| | lenient_length.pl | | TLDR_lenient_length.txt | Like length/2 but takes an additional option to select whether it should behave like SWI-Prolog's length/2, additionally throw if it receives a negative integer as Length, or never throw and just fail on bad arguments. |
| | list_of_numbered_pairs.pl | | | |
| | openlist_append.pl | openlist_append.plt | | |
| | partition_freely.pl | partition_freely.plt | | |
| | probe_length.pl | probe_length.plt | TLDR_probe_length.txt | Probe the length of an unknown term (generally a proper list or an open list, but not necessarily). Tells you what it found. This is a complement to length/2, which only handles proper lists. |
| | random_atom.pl | random_atom.plt | | Utilities to generate random atoms or strings (generally for tests) |
| | randomly_insert.pl | randomly_insert.plt | | |
| | randomly_select.pl | randomly_select.plt | | Randomly select a key from a dict, where the dict values indicate the relative probability of selection by length-of-atom. |
| | replace0.pl | replace0.plt | | |
| | rotate_list.pl | rotate_list.plt | | Rotate a list by N positions |
| | splinter0.pl | splinter0.plt | | Splinter a list by index, i.e. decompose into a prefix, an element, a suffix. |
| | vector_nth0.pl | vector_nth0.plt | | A “vector” version of nth0 |
| | vector_replace0.pl | vector_replace0.plt | | A “vector replace” to replace values in a list |
| support | meta_helpers.pl | | | Provides meta predicates like if_then_else, switch, unless to make code easier to read |
| | safe_format.pl | | | Wraps format/3 into a catch to prevent surprises in running code due to stupid formatting mistakes. |
| | throwme_nonmodular.pl | | | A non-module (the file has to be included) providing skeleton code for throwing exceptions cleanly, in the sense that exception messages are composed in dedicated predicates, not inline. |
| strings | justify.pl | justify.plt | | Straightforward "string justification": left, right, center, with cutting |
| | string_of_spaces.pl | string_of_spaces.plt | string_of_spaces_performance.plt | Quickly generate strings made entirely of the character 0x20 ("SPACE") |
| | string_overwrite.pl | string_overwrite.plt | string_overwrite_performance.plt | Overwrite a string/atom "Lower" with another string/atom "Upper" |
| | stringy.pl | stringy.plt | | A few very simple predicates that try to make "manipulation of strings" and "manipulation of atoms" a bit more uniform. |
| | tablify.pl | tablify.plt | | Print data in tabular form |
| terms | Experimental | | | |