

# Documentation For

`main.py`

---

April 09, 2025

Generated by Documentation Generator

# Table of Contents

| Section           |
|-------------------|
| Classes           |
| Functions         |
| If Statements     |
| For Statements    |
| While Statements  |
| Try Statements    |
| Switch Statements |

## Classes

| Name             | Methods | Description  |
|------------------|---------|--|
| TranslateRequest |         | Defines the request model for translation.         |
| CodeInput        |         | Defines the input model for custom code execution. |

# Functions

| Name            | Params  | Returns | Description  |
|-----------------|---|---------|--|
| clean_spaces    | ['text']  | str     | Replaces non-breaking spaces with regular spaces.      |
| validate_syntax | ['line', 'alias_map', 'syntax_map', 'enclosures', 'settings'] | None    | Validates a line of code against a defined syntax.     |
| translate       | ['request']   | None    | Translates custom code to Python and executes it.      |
| run_custom_code | ['input']   | None    | Runs custom code line by line after translation.       |
| define_language | ['blocks']  | None    | Defines a custom language with keywords and datatypes. |
| home            | []  | None    | Returns a welcome message.                             |
| create_language | ['language']  | None    | Saves a new language definition.                       |
| list_languages  | []  | None    | Lists all defined languages.                           |

## If Statements

| Line No. | Condition   | Description   |
|----------|---|---|
| 45       | <code>line.startswith(alias)</code>   | Runs custom code line by line after translation.                        |
| 167      | <code>not line.strip()</code>   | Returns a welcome message.  |
| 170      | <code>result.get('success')</code>  | Saves a new language definition.  |
| 190      | <code>block['type'].startswith('custom_')</code>  | Lists all defined languages.  |
| 48       | <code>not expected_syntax</code>  | Iterates through aliases and validates syntax for a line of code.       |
| 57       | <code>input_parts[0] != alias</code>  | Handles translation and execution of custom code, catching exceptions.  |
| 61       | <code>'parameters' in syntax_parts or 'statements' in syntax_parts</code>                   | Iterates through lines of code and runs each line.                      |
| 129      | <code>not validate_syntax(line.strip(), alias_map, syntax_map, enclosures, settings)</code> | Executes translated code and handles exceptions.                        |
| 141      | <code>not matched</code>  | Checks if a line starts with a given alias and validates its syntax.    |
| 192      | <code>block['type'] == 'datatype'</code>  | Iterates through lines of code, validates syntax, and translates them.  |
| 62       | <code>len(input_parts) &lt; 2</code>  | Skips empty lines.  |
| 78       | <code>settings.get('enforce_indentation') and 'statements' in syntax_parts</code>           | Processes custom keywords and datatypes in language definition blocks.  |
| 87       | <code>len(input_parts) != len(syntax_parts)</code>  | Checks if a syntax definition exists for a given keyword.               |
| 134      | <code>re.match(f'^{re.escape(alias)}\\b', line)</code>                                      | Checks for parameters/statements and validates their enclosures.        |
| 197      | <code>(start, end) in used_enclosures</code>  | Checks syntax validity and returns an error if invalid.                 |
| 74       | <code>not matched</code>  | Returns an error if no alias is matched for a line.                     |
| 82       | <code>'statements' in syntax_parts</code>   | Checks if the block defines a datatype and handles enclosure conflicts. |
| 92       | <code>expected == 'keyword' and actual != alias</code>                                      | Checks if there are enough parts in a line for parameters/statements.   |

| Line No. | Condition  | Description  |
|----------|--|--|
| 198      | not (dtype in ['char', 'string'] and any((d in ['char', 'string'] and datatype_map[d]['start'] == start and (datatype_map[d]['end'] == end) for d in datatype_map))) | Iterates through parts of a line and checks their enclosures.              |
| 70       | part.startswith(wrap['start']) and part.endswith(wrap['end'])  | Checks for indentation enforcement or statement enclosures.                |
| 95       | expected not in ['parameters', 'statements'] and expected != actual  | Checks for mismatched part counts between expected syntax and actual line. |

## For Statements

| Line No. | Condition  | Description  |
|----------|--|--|
| 44       | (alias, keyword) in alias_map.items()                | Replaces non-breaking spaces with regular spaces.                                  |
| 166      | line in input.code.splitlines()                      | Defines the request model for translation.   |
| 189      | block in blocks                                      | Defines the input model for custom code execution.                                 |
| 128      | line in raw_code.strip().splitlines()                | Defines a custom language with keywords and datatypes.                             |
| 133      | (alias, keyword) in alias_map.items()                | Iterates through language definition blocks and updates keyword and datatype maps. |
| 66       | part in input_parts[1:]                              | Checks if a code execution was successful and handles the result.                  |
| 91       | (expected, actual) in zip(syntax_parts, input_parts) | Checks if the first part of a line matches the expected alias.                     |
| 69       | (dtype, wrap) in enclosures.items()                  | Iterates through aliases and matches them to lines of code.                        |

## While Statements

No data available.



## Try Statements

| Line No. | Caught Error Name | Description  |
|----------|-------------------|--|
| 113      | e                 | Validates a line of code against a defined syntax. |
| 174      | e                 | Translates custom code to Python and executes it.  |

# Switch Statements

No data available.