

---

# **mathparse Documentation**

*Release 0.0.1*

**Gunther Cox**

**Nov 23, 2018**



---

Contents:

---

<b>1</b>	<b>Parsing strings</b>	<b>3</b>
<b>2</b>	<b>The mathwords utility module</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



mathparse is a library for solving mathematical equations contained in strings



Methods for evaluating mathematical equations in strings.

**exception** `mathparse.mathparse.PostfixTokenEvaluationException`

Exception to be raised when a language code is given that is not a part of the ISO 639-2 standard.

`mathparse.mathparse.evaluate_postfix` (*tokens*)

Given a list of evaluable tokens in postfix format, calculate a solution.

`mathparse.mathparse.extract_expression` (*dirty\_string*, *language*)

Give a string such as: “What is 4 + 4?” Return the string “4 + 4”

`mathparse.mathparse.find_word_groups` (*string*, *words*)

Find matches for words in the format “3 thousand 6 hundred 2”. The words parameter should be the list of words to check for such as “hundred”.

`mathparse.mathparse.is_binary` (*string*)

Return true if the string is a defined binary operator.

`mathparse.mathparse.is_constant` (*string*)

Return true if the string is a mathematical constant.

`mathparse.mathparse.is_float` (*string*)

Return true if the string is a float.

`mathparse.mathparse.is_int` (*string*)

Return true if string is an integer.

`mathparse.mathparse.is_symbol` (*string*)

Return true if the string is a mathematical symbol.

`mathparse.mathparse.is_unary` (*string*)

Return true if the string is a defined unary mathematical operator function.

`mathparse.mathparse.is_word` (*word*, *language*)

Return true if the word is a math word for the specified language.

`mathparse.mathparse.parse` (*string*, *language=None*)

Return a solution to the equation in the input string.

`mathparse.mathparse.replace_word_tokens` (*string*, *language*)

Given a string and an ISO 639-2 language code, return the string with the words replaced with an operational equivalent.

`mathparse.mathparse.to_postfix` (*tokens*)

Convert a list of evaluable tokens to postfix format.

`mathparse.mathparse.tokenize` (*string*, *language=None*, *escape='\_\_\_'*)

Given a string, return a list of math symbol tokens



---

### The mathwords utility module

---

**exception** `mathparse.mathwords.InvalidLanguageCodeException`

Exception to be raised when a language code is given that is not a part of the ISO 639-2 standard.

`mathparse.mathwords.word_groups_for_language` (*language\_code*)

Return the math word groups for a language code. The `language_code` should be an ISO 639-2 language code. [https://www.loc.gov/standards/iso639-2/php/code\\_list.php](https://www.loc.gov/standards/iso639-2/php/code_list.php)

`mathparse.mathwords.words_for_language` (*language\_code*)

Return the math words for a language code. The `language_code` should be an ISO 639-2 language code. [https://www.loc.gov/standards/iso639-2/php/code\\_list.php](https://www.loc.gov/standards/iso639-2/php/code_list.php)



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
  - `modindex`
  - `search`
- 

A special thanks to [Griffin Cox](#) for the design of the Mathparse logo.



**m**

`mathparse`, ??

`mathparse.mathparse`, 3

`mathparse.mathwords`, 5



## E

`evaluate_postfix()` (in module `mathparse.mathparse`), 3  
`extract_expression()` (in module `mathparse.mathparse`), 3  
`words_for_language()` (in module `mathparse.mathwords`), 5

## F

`find_word_groups()` (in module `mathparse.mathparse`), 3

## I

`InvalidLanguageCodeException`, 5  
`is_binary()` (in module `mathparse.mathparse`), 3  
`is_constant()` (in module `mathparse.mathparse`), 3  
`is_float()` (in module `mathparse.mathparse`), 3  
`is_int()` (in module `mathparse.mathparse`), 3  
`is_symbol()` (in module `mathparse.mathparse`), 3  
`is_unary()` (in module `mathparse.mathparse`), 3  
`is_word()` (in module `mathparse.mathparse`), 3

## M

`mathparse` (module), 1  
`mathparse.mathparse` (module), 3  
`mathparse.mathwords` (module), 5

## P

`parse()` (in module `mathparse.mathparse`), 3  
`PostfixTokenEvaluationException`, 3

## R

`replace_word_tokens()` (in module `mathparse.mathparse`), 3

## T

`to_postfix()` (in module `mathparse.mathparse`), 4  
`tokenize()` (in module `mathparse.mathparse`), 4

## W

`word_groups_for_language()` (in module `mathparse.mathwords`), 5