
pyscience Documentation

Release 0.1.0.dev2

Manuel Alcaraz

Feb 19, 2019

Contents:

1	Pyscience installation	3
2	User guide	5
2.1	Working as a calculator	5
2.2	Working with algebra	6
2.3	Working with chemical elements	7
3	License	9
4	Indices and tables	11

Pyscience is a easy python library and command-line application to work with mathematical operations and other science related programming. Also you can manage CSV data with the datam module. See API documentation for more information.

Pyscience requires Python 3.7+ to work.

CHAPTER 1

Pyscience installation

Pyscience needs a recent version of python (3.7). You can download it [here](#). It won't work in older versions. Pyscience is available in pip. To install, type with superuser privileges:

```
pip3 install pyscience
```

If you want to install the latest development version, use the github repo:

```
GitHub repository is not yet available
```


Pyscience's interpreter is very easy to use. Start it with:

```
pyscience
```

2.1 Working as a calculator

Pyscience uses python `eval` function to evaluate expressions after expand it. You can use pyscience as a normal calculator:

```
> 2 + 2
4
> 3 * (2 + 3)
15
> 3 (3+4)
7
```

2.1.1 Addition

To add two numbers, use the `+` operator:

```
> 2 + 3
5
```

2.1.2 Subtraction

To subtract two number, use the `-` operator:

```
> 2 - 3
-1
```

2.1.3 Multiplication

To multiply two numbers, use the `*` operator:

```
> 2 * 3
6
```

2.1.4 Division

To divide one number by other, use the `/` operator:

```
> 8 / 2
4
```

2.1.5 Powers

You can create powers using the `**` operator:

```
> 2 ** 4
16
```

or using `^` numbers:

```
> 2^4
16
```

2.1.6 Fractions

The `F (Fraction)` class provides a way to create and operate with fractions. Numerator and denominator are divided using a coma. For example:

```
> F(2,3) + F(3,4)
F(17,12)
```

You can use the same operators for fractions.

2.2 Working with algebra

Pyscience can operate with Monomials, Variables and Polynomials. Some examples of what can you do:

```
> 2x + 3x
5x
> 3x * 6y
18xy
> 2x / (2x)
1
```

Note: In the last example, you can think why I have put parenthesis for the second Monomial. If you don't do it, you will divide $2x$ by 2 and, AFTER, you will multiply the result by x . In this case, the final result is x^2

2.3 Working with chemical elements

Pyscience can show you basic information about chemical elements. You can do it with the `CE` function:

```
> CE('H')  
...
```

If you want to set a specific mass for the element, indicate that between brackets:

```
> CE('Si(32)') # Set mass to 32  
...
```

Also, you can work with elements which have charge:

```
> CE('Si2+')  
...
```

If you know the atomic number of a element but not the symbol, you can get the element by its atomic number:

```
> CE(20)  
# Calcium (Ca)
```


CHAPTER 3

License

pyscience - python science programming

Copyright (c) 2019 Manuel Alcaraz Zambrano

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 4

Indices and tables

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