

Intro to Programming with Python

Python Documentation and
Stackoverflow

What/Where is the Python Documentation

1. Go here: <https://docs.python.org/3.6/>

The screenshot shows the Python 3.6.2rc2 documentation page. At the top, the breadcrumb navigation shows 'Python » 3.6.2rc2 » Documentation', with '3.6.2rc2' circled in purple. A search bar and links for 'modules' and 'index' are on the right. The left sidebar contains links for 'Download', 'Docs for other versions', and 'Other resources'. The main content area is titled 'Python 3.6.2rc2 documentation' and includes a welcome message and a list of 'Parts of the documentation'. The 'Tutorial' link is circled in purple. Other links include 'What's new in Python 3.6?', 'Library Reference', 'Language Reference', 'Python Setup and Usage', 'Python HOWTOs', 'Installing Python Modules', 'Distributing Python Modules', 'Extending and Embedding', 'Python/C API', and 'FAQs'.

Python » 3.6.2rc2 » Documentation » Quick search Go | modules | index

Download
Download these documents

Docs for other versions
Python 2.7 (stable)
Python 3.5 (stable)
Python 3.7 (in development)
Old versions

Other resources
PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Python 3.6.2rc2 documentation

Welcome! This is the documentation for Python 3.6.2rc2.

Parts of the documentation:

- [What's new in Python 3.6?](#)
or all "What's new" documents since 2.0
- [Tutorial](#)
start here
- [Library Reference](#)
keep this under your pillow
- [Language Reference](#)
describes syntax and language elements
- [Python Setup and Usage](#)
how to use Python on different platforms
- [Python HOWTOs](#)
in-depth documents on specific topics
- [Installing Python Modules](#)
installing from the Python Package Index & other sources
- [Distributing Python Modules](#)
publishing modules for installation by others
- [Extending and Embedding](#)
tutorial for C/C++ programmers
- [Python/C API](#)
reference for C/C++ programmers
- [FAQs](#)
frequently asked questions (with answers!)

Python Documentation

The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see [The Python Standard Library](#). [The Python Language Reference](#) gives a more formal definition of the language. To write extensions in C or C++, read [Extending and Embedding the Python Interpreter](#) and [Python/C API Reference Manual](#). There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [The Python Standard Library](#).

The [Glossary](#) is also worth going through.

- [1. Whetting Your Appetite](#)
- [2. Using the Python Interpreter](#)
 - [2.1. Invoking the Interpreter](#)
 - [2.1.1. Argument Passing](#)
 - [2.1.2. Interactive Mode](#)
 - [2.2. The Interpreter and Its Environment](#)
 - [2.2.1. Source Code Encoding](#)
- [3. An Informal Introduction to Python](#)
 - [3.1. Using Python as a Calculator](#)
 - [3.1.1. Numbers](#)

Python Documentation

3.1.1. Numbers

The interpreter acts as a simple calculator: you can type an expression at it and it will write the value. Expression syntax is straightforward: the operators `+`, `-`, `*` and `/` work just like in most other languages (for example, Pascal or C); parentheses `()` can be used for grouping. For example:

```
>>> 2 + 2
4
>>> 50 - 5*6
20
>>> (50 - 5*6) / 4
5.0
>>> 8 / 5 # division always returns a floating point number
1.6
```

The integer numbers (e.g. 2, 4, 20) have type `int`, the ones with a fractional part (e.g. 5.0, 1.6) have type `float`. We will see more about numeric types later in the tutorial.

Division (`/`) always returns a float. To do [floor division](#) and get an integer result (discarding any fractional result) you can use the `//` operator; to calculate the remainder you can use `%`:

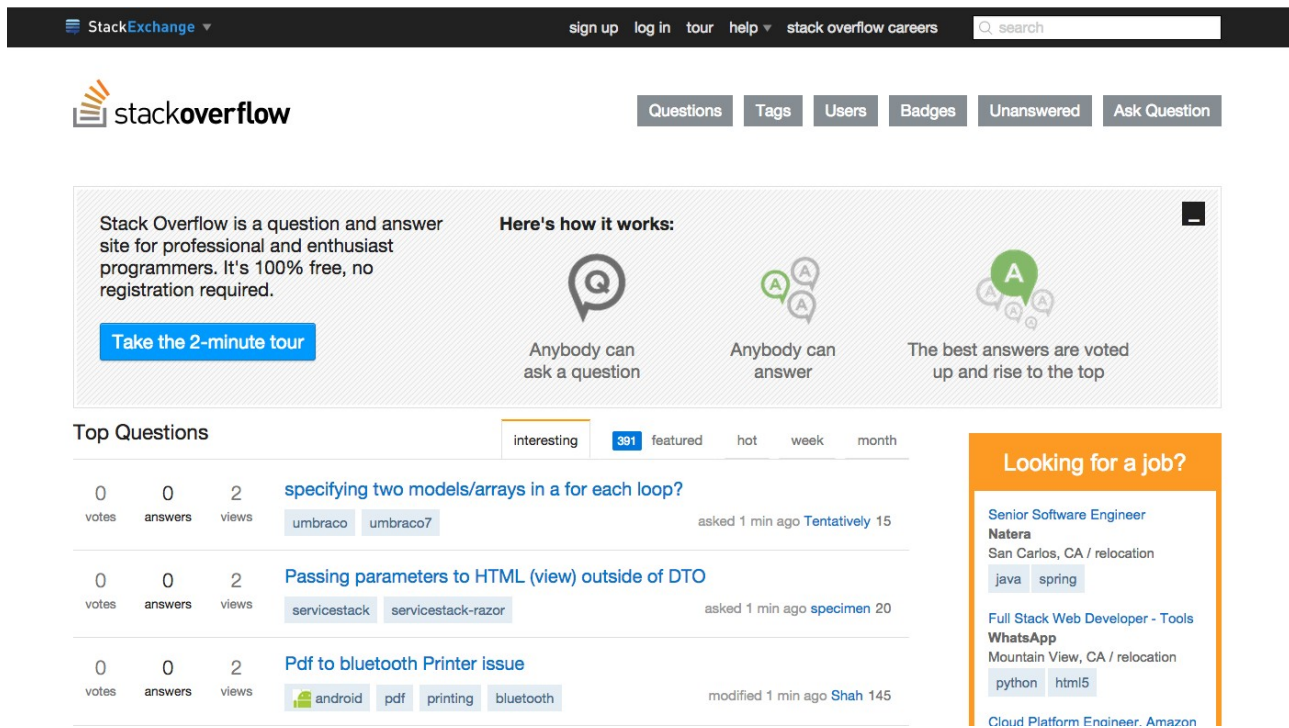
```
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>>
>>> 17 // 3 # floor division discards the fractional part
5
>>> 17 % 3 # the % operator returns the remainder of the division
2
>>> 5 * 3 + 2 # result * divisor + remainder
17
```

With Python, it is possible to use the `**` operator to calculate powers [1]:

```
>>> 5 ** 2 # 5 squared
25
>>> 2 ** 7 # 2 to the power of 7
128
```

Stackoverflow

- Go to: <http://stackoverflow.com/>



The screenshot shows the Stack Overflow homepage. At the top is a dark navigation bar with the StackExchange logo, links for sign up, log in, tour, help, and stack overflow careers, and a search bar. Below this is the Stack Overflow logo and a row of buttons: Questions, Tags, Users, Badges, Unanswered, and Ask Question. The main content area features a 'Here's how it works' section with three icons and text: 'Anybody can ask a question', 'Anybody can answer', and 'The best answers are voted up and rise to the top'. Below this is a 'Top Questions' section with a filter bar (interesting, 391 featured, hot, week, month). Three questions are listed: 'specifying two models/arrays in a for each loop?' (0 votes, 0 answers, 2 views, tags: umbraco, umbraco7), 'Passing parameters to HTML (view) outside of DTO' (0 votes, 0 answers, 2 views, tags: servicestack, servicestack-razor), and 'Pdf to bluetooth Printer issue' (0 votes, 0 answers, 2 views, tags: android, pdf, printing, bluetooth). On the right side, there is a 'Looking for a job?' section with job listings for 'Senior Software Engineer' at Natera and 'Full Stack Web Developer - Tools' at WhatsApp.

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour

Here's how it works:

- Anybody can ask a question
- Anybody can answer
- The best answers are voted up and rise to the top

Top Questions

interesting 391 featured hot week month

votes	answers	views	question	tags	status
0	0	2	specifying two models/arrays in a for each loop?	umbraco umbraco7	asked 1 min ago Tentatively 15
0	0	2	Passing parameters to HTML (view) outside of DTO	servicestack servicestack-razor	asked 1 min ago specimen 20
0	0	2	Pdf to bluetooth Printer issue	android pdf printing bluetooth	modified 1 min ago Shah 145

Looking for a job?

Senior Software Engineer
Natera
San Carlos, CA / relocation
java spring

Full Stack Web Developer - Tools
WhatsApp
Mountain View, CA / relocation
python html5

Cloud Platform Engineer. Amazon

Stackoverflow

StackExchange ▾

sign up log in tour help ▾ stack overflow careers

python lambda function



Questions

Tags

Users

Badges

Unanswered

Ask Question

Search

python lambda function

search

[Advanced Search Tips](#)

1,785 results

relevance

newest

votes

active

2

votes

5

answers

Q: Python: Lambda function

I'm relatively new to **Lambda** functions and esp this one got me quite confused. I have a set of words that have an average length 3.4 and list of words = ['hello', 'my', 'name', 'is', 'lisa'] I want ... to compare the length of a word to average length and print out only the words higher than average length average = 3.4 words = ['hello', 'my', 'name', 'is', 'lisa'] print(filter(**lambda** x: len(words ...

python

asked Nov 6 '14 by [pirelius](#)

20

votes

5

answers

Q: Equivalent for Python's lambda functions in Java?

Can someone please tell me if there is an equivalent for Python's **lambda** functions in Java? ...

java

python

function

lambda

asked May 30 '09 by [tresnja](#)

11

votes

Q: Python lambda function

What is happening here? reduce(**lambda** x,y: x+y, [x for x in range(1,1000) if x % 3 == 0 or x % 5 == 0]) I understand how x is iterating through all of the numbers from 1 to 999 and taking out those that are divisible by 3 or 5, but the **lambda** function part is confusing me.

results found containing
python lambda function



Looking for a job?

UCI Division of Continuing Education

Success

SUCCESS!