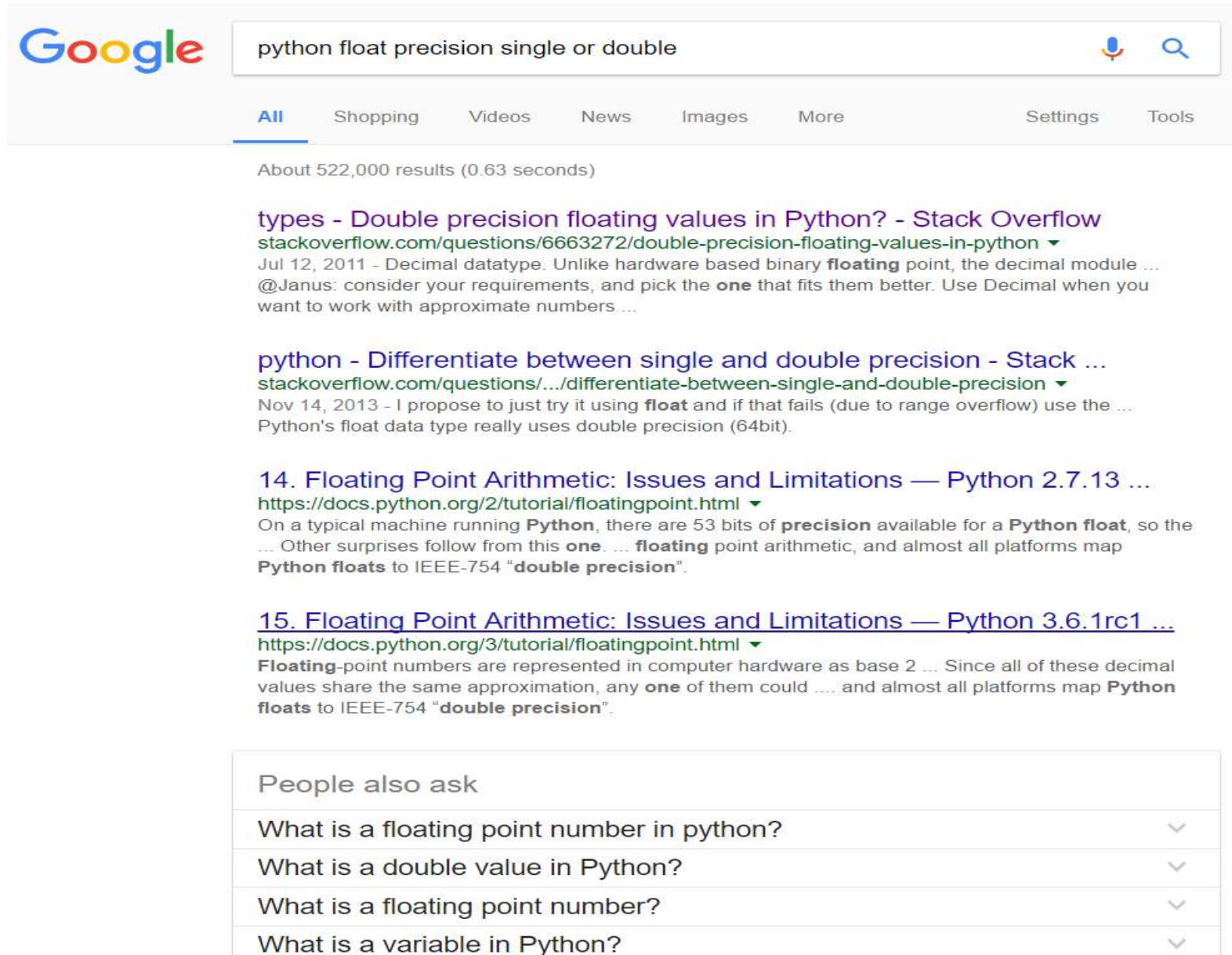


Programmazione 2

Introduzione al corso

Domande lezione precedente 1/2



The image shows a Google search interface. The search bar contains the text "python float precision single or double". Below the search bar, there are navigation tabs for "All", "Shopping", "Videos", "News", "Images", "More", "Settings", and "Tools". The "All" tab is selected. Below the tabs, it says "About 522,000 results (0.63 seconds)". There are four search results listed. The first result is titled "types - Double precision floating values in Python? - Stack Overflow" and includes a link to a Stack Overflow question. The second result is titled "python - Differentiate between single and double precision - Stack ..." and also includes a link to a Stack Overflow question. The third result is titled "14. Floating Point Arithmetic: Issues and Limitations — Python 2.7.13 ..." and includes a link to a Python documentation page. The fourth result is titled "15. Floating Point Arithmetic: Issues and Limitations — Python 3.6.1rc1 ..." and includes a link to a Python documentation page. At the bottom of the search results, there is a section titled "People also ask" with four questions listed, each with a downward arrow indicating it can be expanded.

Google

python float precision single or double

All Shopping Videos News Images More Settings Tools

About 522,000 results (0.63 seconds)

[types - Double precision floating values in Python? - Stack Overflow](#)
stackoverflow.com/questions/6663272/double-precision-floating-values-in-python ▾
Jul 12, 2011 - Decimal datatype. Unlike hardware based binary **floating** point, the decimal module ...
@Janus: consider your requirements, and pick the **one** that fits them better. Use Decimal when you want to work with approximate numbers ...

[python - Differentiate between single and double precision - Stack ...](#)
stackoverflow.com/questions/.../differentiate-between-single-and-double-precision ▾
Nov 14, 2013 - I propose to just try it using **float** and if that fails (due to range overflow) use the ...
Python's float data type really uses double precision (64bit).

[14. Floating Point Arithmetic: Issues and Limitations — Python 2.7.13 ...](#)
<https://docs.python.org/2/tutorial/floatingpoint.html> ▾
On a typical machine running **Python**, there are 53 bits of **precision** available for a **Python float**, so the ...
Other surprises follow from this **one**. ... **floating** point arithmetic, and almost all platforms map **Python floats** to IEEE-754 "**double precision**".

[15. Floating Point Arithmetic: Issues and Limitations — Python 3.6.1rc1 ...](#)
<https://docs.python.org/3/tutorial/floatingpoint.html> ▾
Floating-point numbers are represented in computer hardware as base 2 ... Since all of these decimal values share the same approximation, any **one** of them could ... and almost all platforms map **Python floats** to IEEE-754 "**double precision**".

People also ask

- What is a floating point number in python? ▾
- What is a double value in Python? ▾
- What is a floating point number? ▾
- What is a variable in Python? ▾

▲ Are there data types with better precision than float?

32

python types



share edit



11

add a comment

asked Jul 12 '11 at 11:09



kravemir

4,081 ● 10 ● 39 ● 72

5 Answers

active

oldest

votes



Decimal datatype

27

- Unlike hardware based binary floating point, the decimal module has a user alterable precision (defaulting to 28 places) which can be as large as needed for a given problem.



If you are pressed by performance issues, have a look at [GMPY](#)



share edit

answered Jul 12 '11 at 11:11



Jacob

24.5k ● 3 ● 49 ● 64

If I were asking the original question, then @larsmans's would be the answer (even though formally it's off-topic). – [Piotr Findeisen](#) Feb 6 '16 at 14:25

add a comment



Tired of recruiter spam?
Want jobs tailored to your needs?



Get started



Python's built-in `float` type has double precision (it's a C `double` in CPython, a Java `double` in Jython). If you need more precision, get [NumPy](#) and use its `numpy.float128`.

54



share edit

edited Jul 12 '11 at 11:27

answered Jul 12 '11 at 11:11



Fred Foo

238k ● 40 ● 463 ● 640



For some applications you can use `Fraction` instead of floating-point numbers.

11



```
>>> from fractions import Fraction
>>> Fraction(1, 3**54)
Fraction(1, 58149737003040059690390169)
```

(For other applications, there's `decimal`, as suggested out by the other responses.)

share edit

answered Jul 12 '11 at 11:16



Gareth Rees

44k ● 5 ● 87 ● 123

how do I choose between Decimal and Fraction? Fraction seems better since it can represent continuing fractions which I guess Decimal can't? – [Janus Troelsen](#) Nov 26 '12 at 13:44

- 1 @Janus: consider your requirements, and pick the one that fits them better. Use `Decimal` when you want to work with approximate numbers that have fixed (but configurable) precision. Use `Fraction` when you want to work with exact ratios, and are prepared to put up with their unbounded storage requirements. – [Gareth Rees](#) Nov 26 '12 at 17:10

Does Fraction support all the operations you can do with float? – [danijar](#) Jan 14 '15 at 10:18

add a comment



May be you need Decimal

8



```
>>> from decimal import Decimal
>>> Decimal(2.675)
Decimal('2.6749999999999982236431605997495353221893310546875')
```

Floating Point Arithmetic

share edit

answered Jul 12 '11 at 11:15



jerboa

478 ● 1 ● 7 ● 16

add a comment

Domande lezione precedente 2/2

In python esistono due operatori per effettuare la divisione tra due numeri a e b

1. ``\`` effettua la divisione come una "normale" calcolatrice (a meno della precisione numerica)
2. ``\\`` effettua la divisione "intera", ovvero esegue il «floor» del risultato della divisione

http://desktop.github.com/

The screenshot displays the GitHub Desktop application interface. On the left, a sidebar shows a list of repositories under the heading 'Tutorial', with 'Tutorial' selected. The main area shows the 'master' branch selected, with tabs for 'Changes' and 'History'. The 'History' tab is active, showing a commit by Josh Abernathy with the message 'Added .gitattributes & .gitignore files'. The commit details show two files: '.gitattributes' and '.gitignore', both with green plus signs indicating they were added. The interface also includes a 'Publish' button and a 'Revert' button.

Changes History Pull request Publish

stegua mathcoding

Filter repositories

Programmazione2

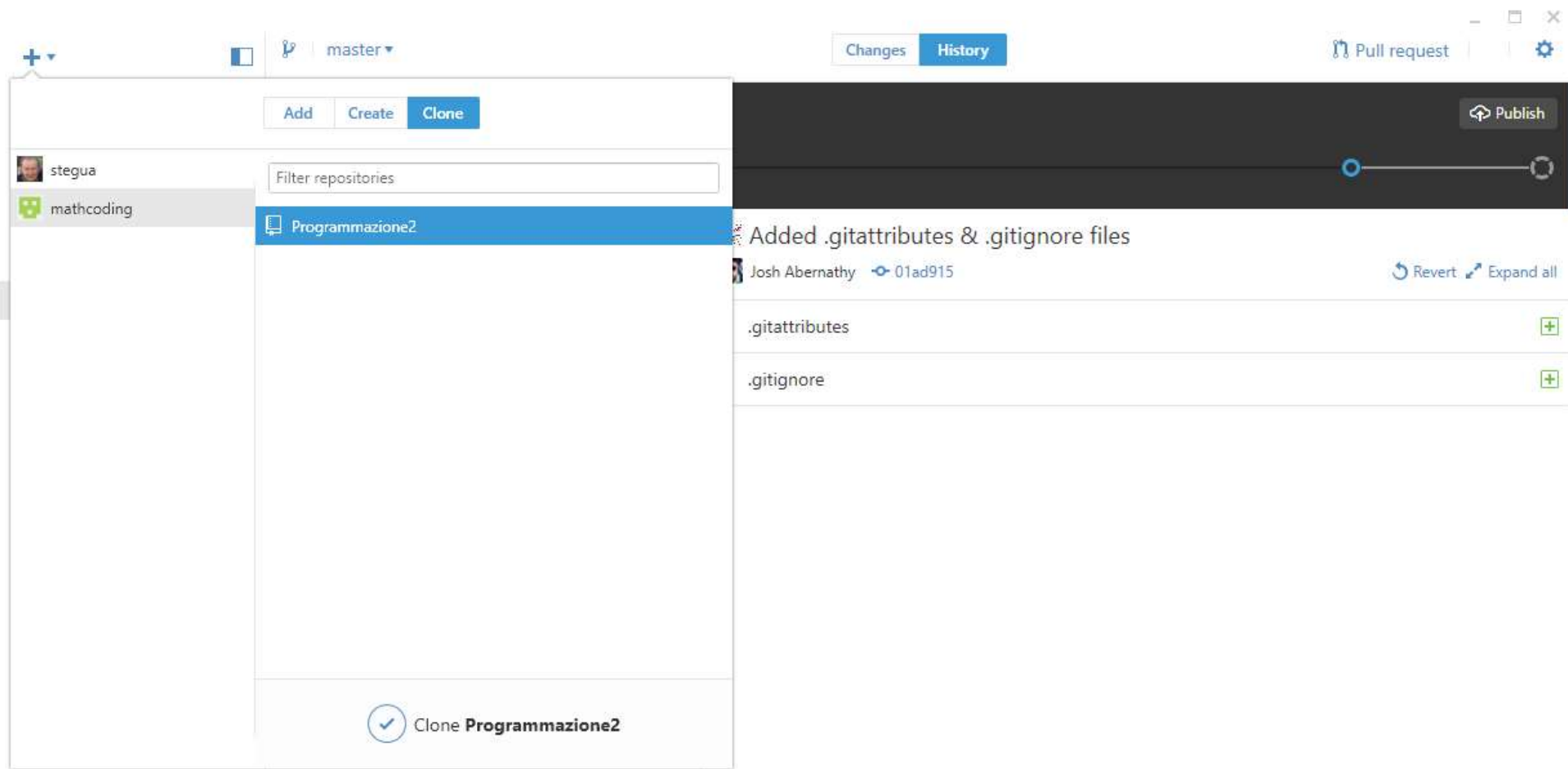
Added .gitattributes & .gitignore files

Josh Abernathy 01ad915 Revert Expand all

.gitattributes +

.gitignore +

Clone Programmazione2





master

Changes

History

Pull request



Filter repositories

GitHub

Programmazione2

stegua.github.com

VehicleRouting

Other

Tutorial

Compare

Sync

master



0 changes

No changes

Summary

Description



Commit to master

No local changes
Would you like to [open this repository](#) in Explorer?