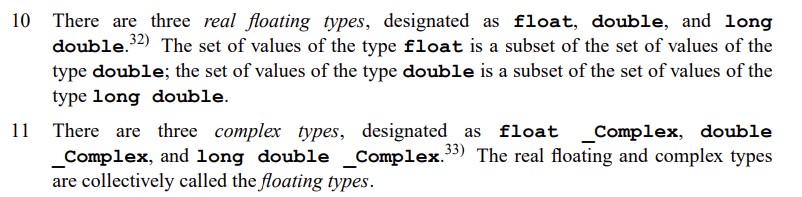
**Know Your System and Your Integration Tools Real Floating Types!**

ISO/IEC 9899:TC3 Committee Draft — September 7, 2007, WG14/N1256

**Shaimaa Soltan**

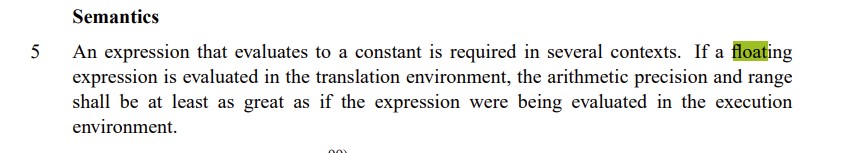
1. By Real Floating point we mean three data types (float, double, long double)



1- real floating are results for an operation that must be represented inside a specified range of a datatype in an execution environment.

2- we need to differentiate between on floating point at rest environment, where you ingest a saved floating pint number from, and the floating-point at execution environment, where you do calculations on floating data types in your integration pipeline and calculations at the execution environment.

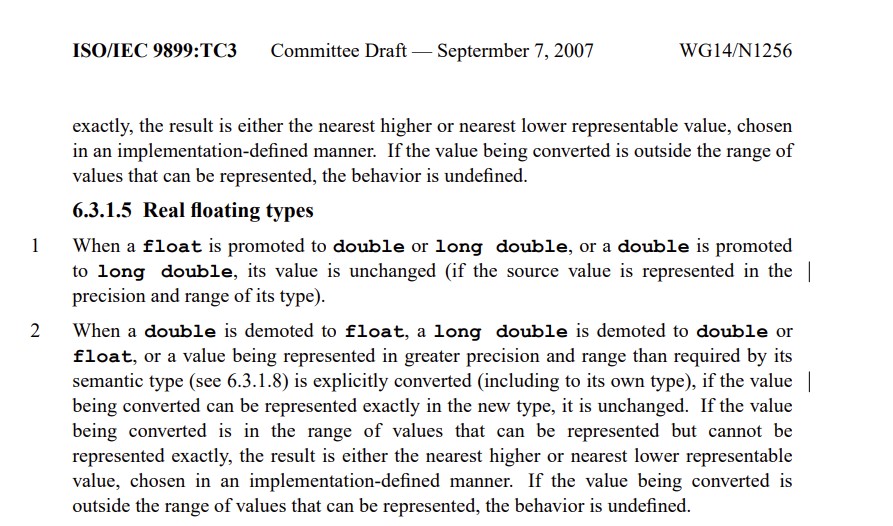
And by execution environment, we mean the used machine capacity or the used tools or the programming language used, and how they define data ranges for a real floating type.



3- floating points at execution environment have four scenarios

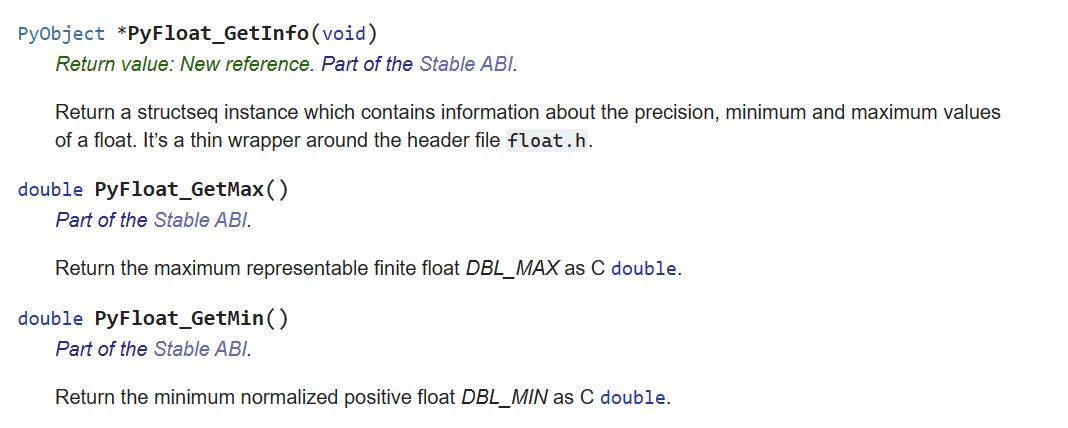
1. Can be represented without any approximation (exact Value) in the defined range for the tool and the machine. (For example, Like when casting float into double or long double)
2. Rounded Up towards + ∞: if the calculation results (demoted) cannot be represented in real floating type range (like casting from double to float or from long double into double)
3. Rounded down towards -∞: if the calculation results (demoted) cannot be represented in real floating type range (like casting from double to float or from long double into double)
4. Undefined behavior: if the value overfloat the allowed data type range.

Round towards ±∞ depends on the defined representation for a real floating in an execution tool or machine or language configuration.

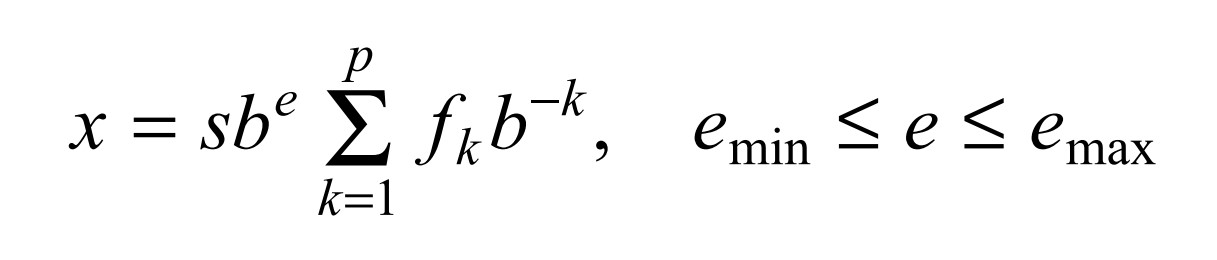


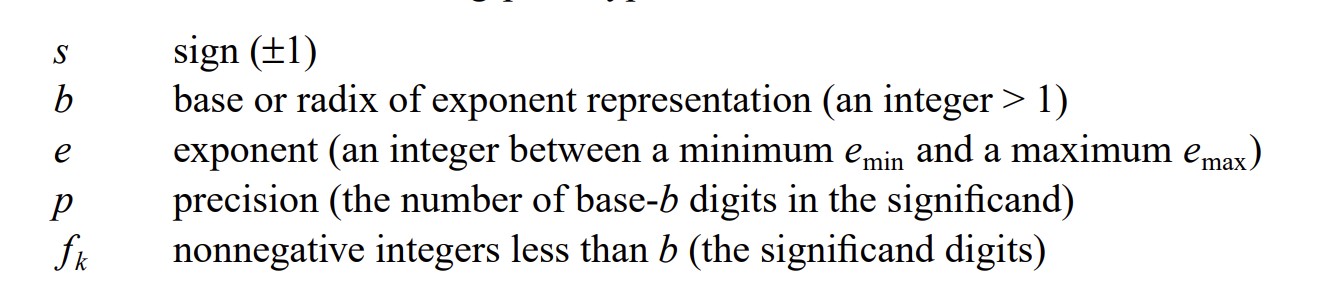
Now we are going to see two languages (Python and .NET) and its real floating type configurations and parameters. (Python uses C++ library <float.h>).

1. Python and <Float.h>



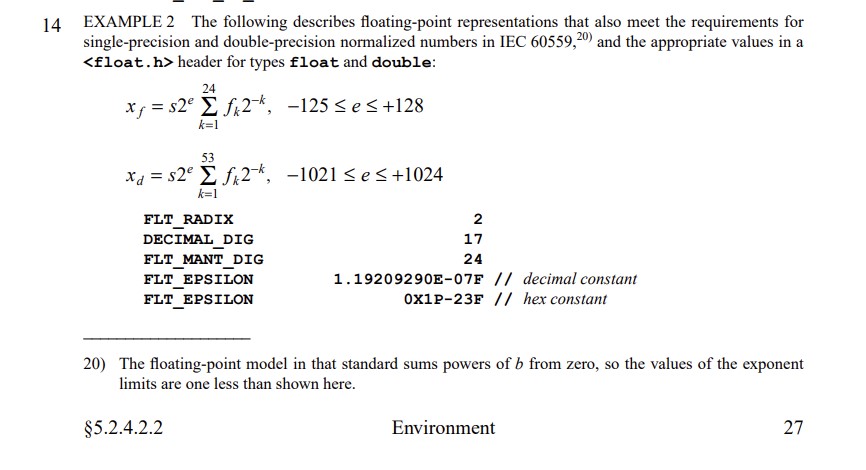
1. <Float.h> include representation and configurations for both double and float types.
2. Floating types can be calculated using a summation series which is called artificial floating-point formula.





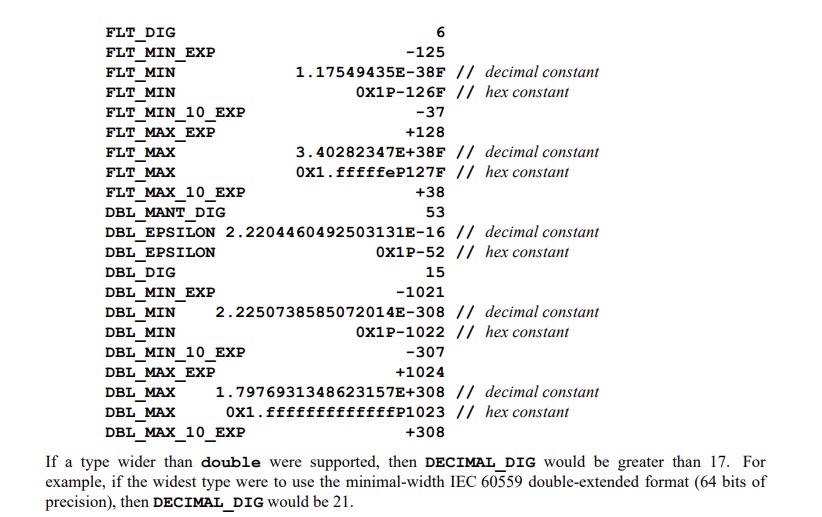
1. In the next image, Xf is the series for float data type and Xd is the series for double data type.

Number of terms in the double series = 53 is bigger than the number of terms in the float series = 24, which gives more accurate approximation for the actual result.



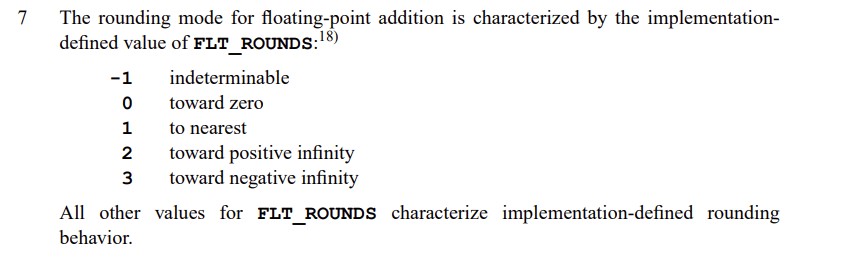
1. <float.h> file have some properties that related to this formula parameters. Float properties are prefixed by FLT\_\* and double properties are prefixed by DBL\_\*.

As it says in the notation in the following image the double-extended (64 bits of precision) the DECIMAL\_DIG = 21.



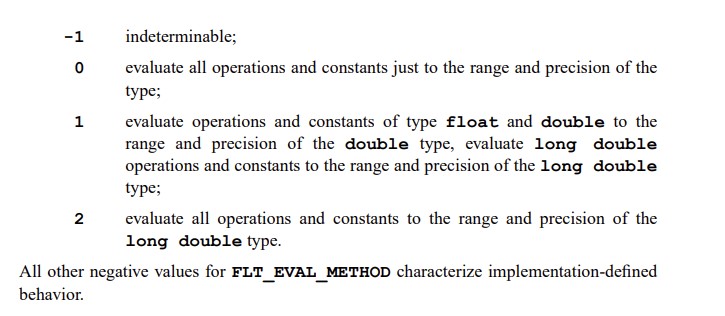


1. <float.h> file have a Rounding mode if the result (promoted, demoted or overfloated) data type range.

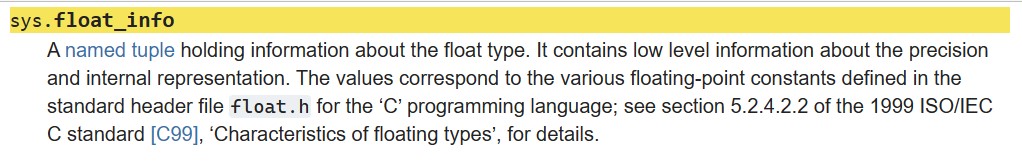


1. <float.h> file have an Evaluation mode. Because we have three types in real floating type, we have three different ranges, therefore we need to know how to move values between types using this Evaluation mode.

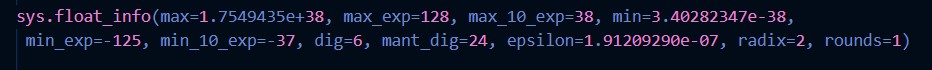
It can be configured to stick to each data type range or use the max range long double or float and double to double range and only long double to double range.



1. You can check your python environment float configuration by calling sys.float\_info

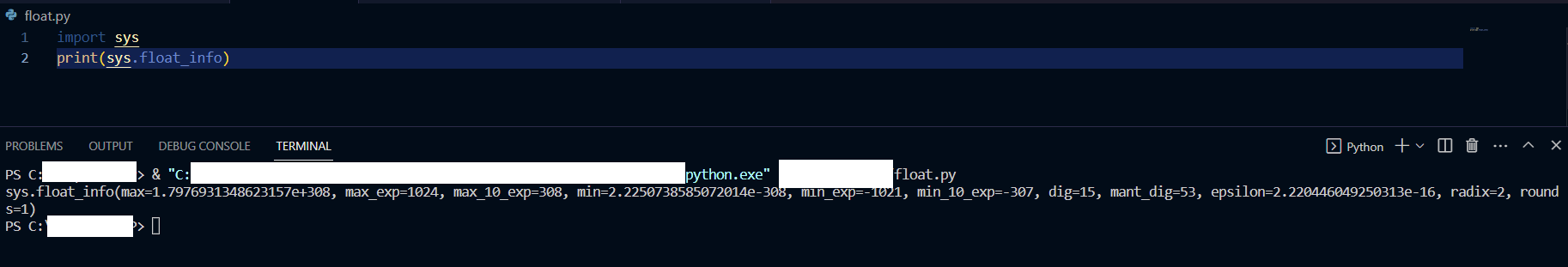


For float

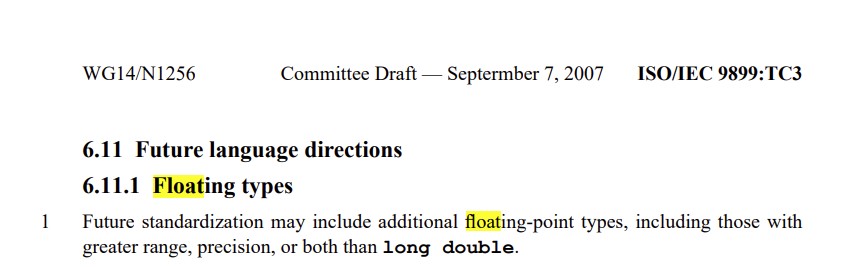


For double

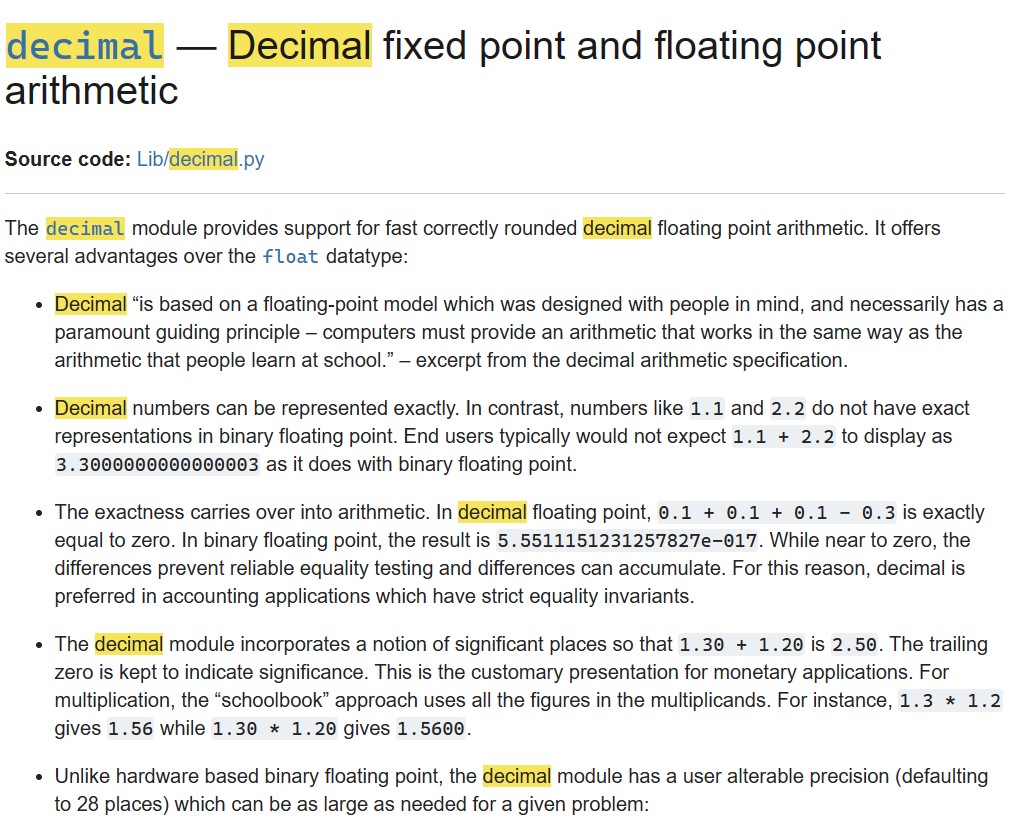




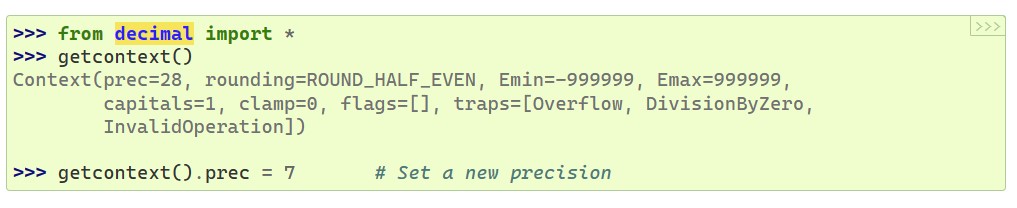
8- A New real floating type may be introduced in the future for C <float.h>.

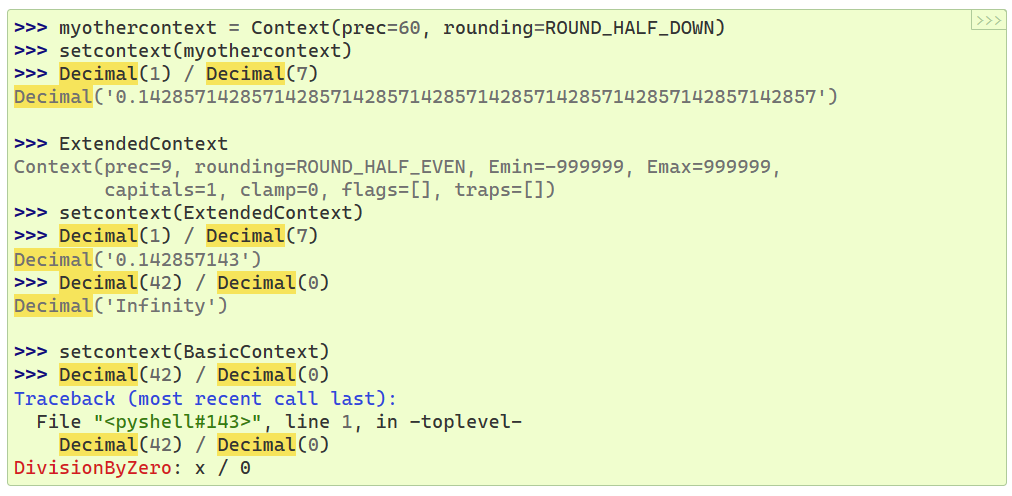


9-python have one module that can be imported to have more accurate real floating type with higher precision (from Decimal import \*).

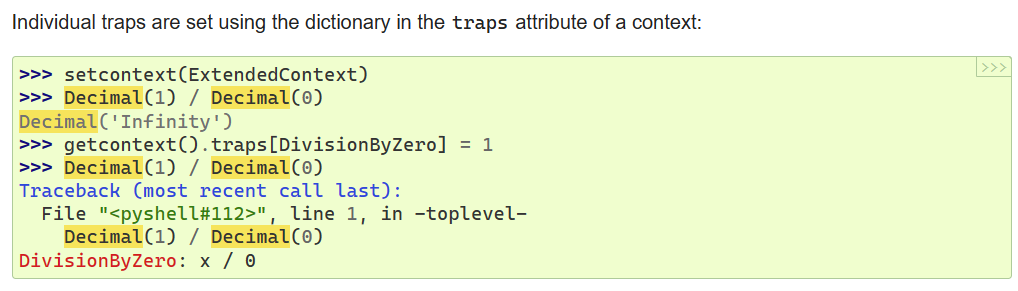


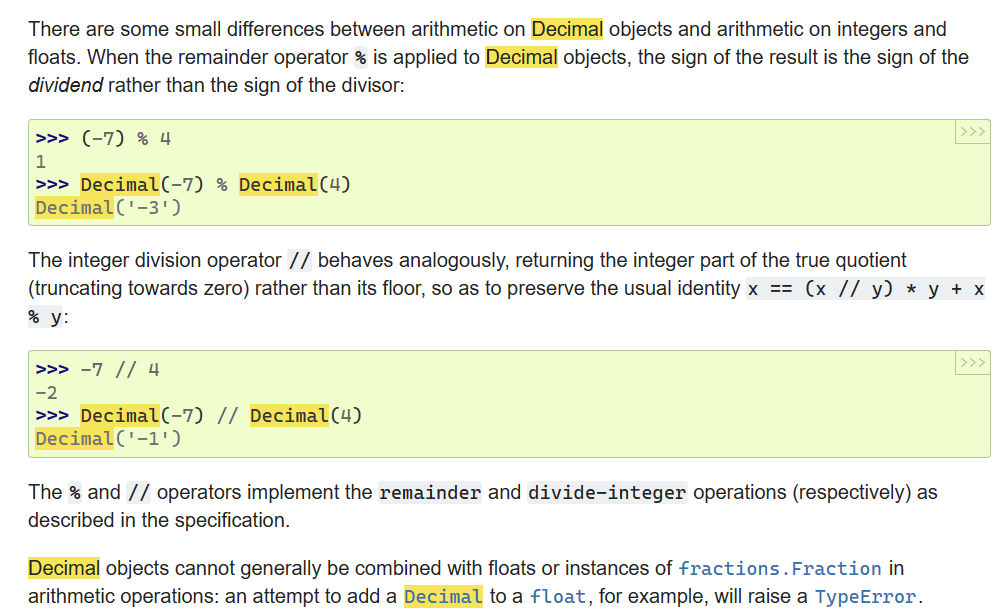
Decimal Type in python can be customized to certain precision by setting the execution context precision to specific precision value.





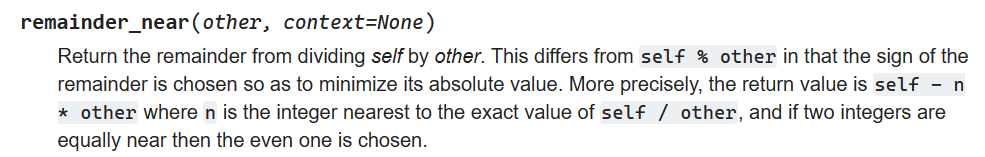
Also, can build customized error handler to avoid float to decimal compare errors or floating type overflow errors.





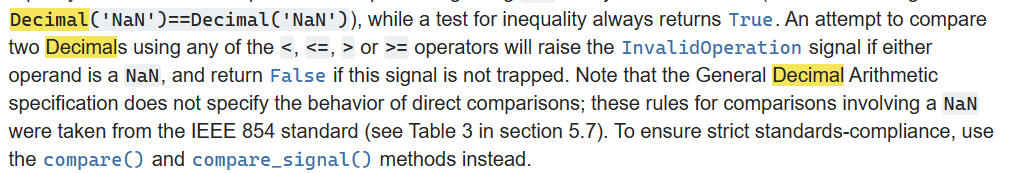
In python you need to check on the updates between versions to overcome some of the older version’s limitations or use better functions or handlers

This is an example for reminder % operator.

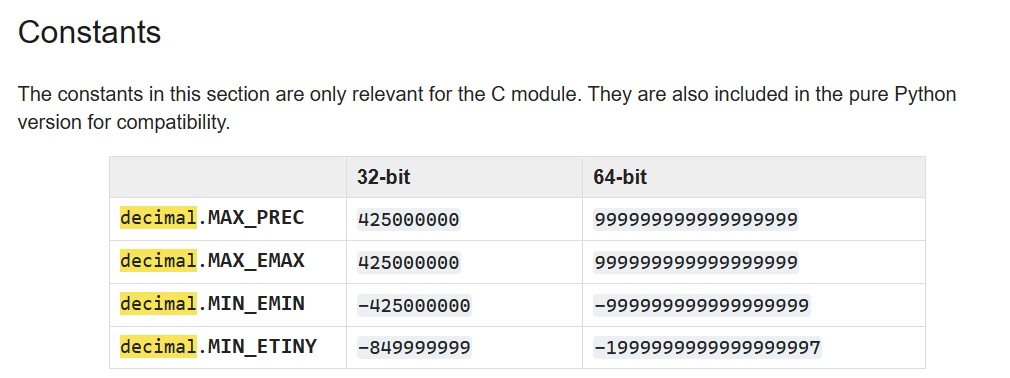


= 

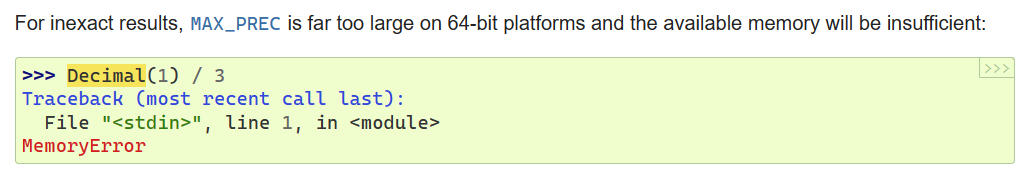
Compare Decimal values in python



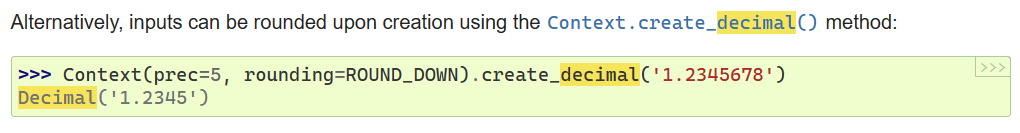
In python Decimal Be aware of the Maximum Precision and your memory size and you machine (32bit Vs 64 bit).





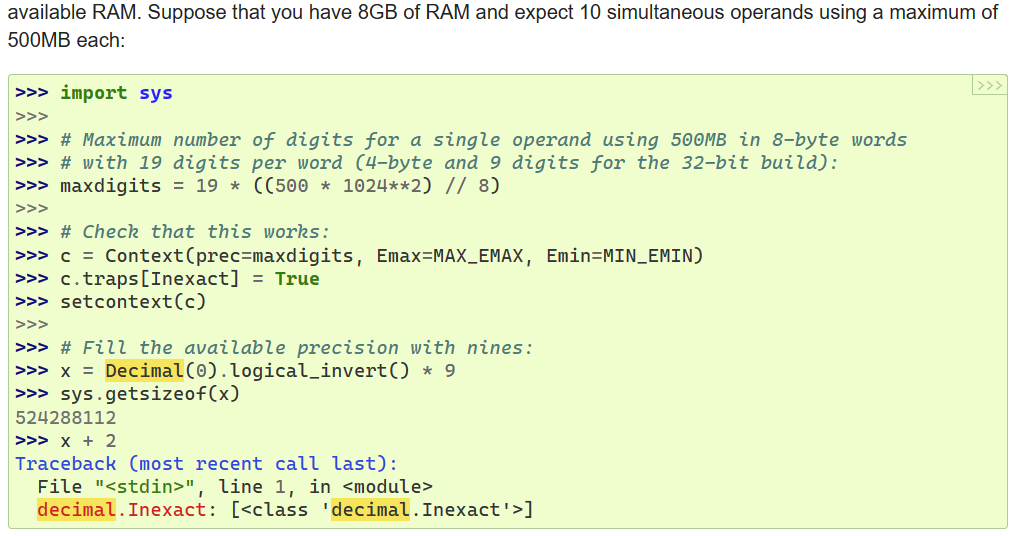


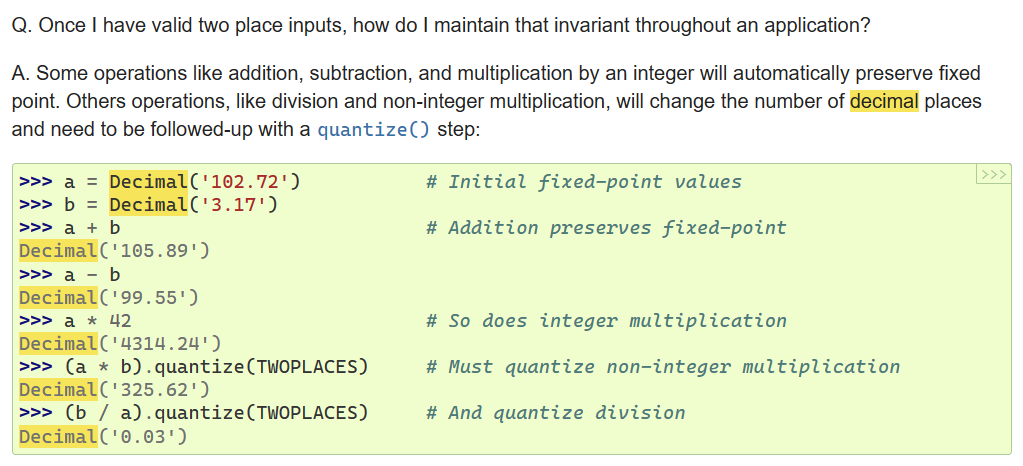




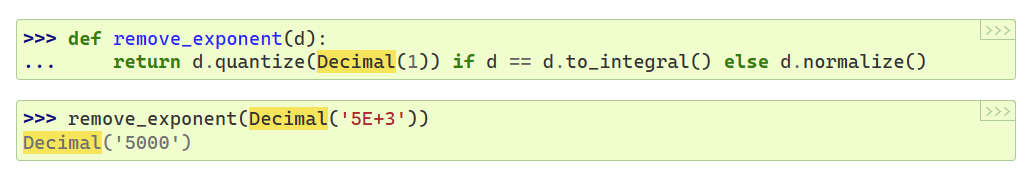
How to choose a precision that fit into your memory?

In the following code You can change 19 until you can fit into your memory size with the best precision.

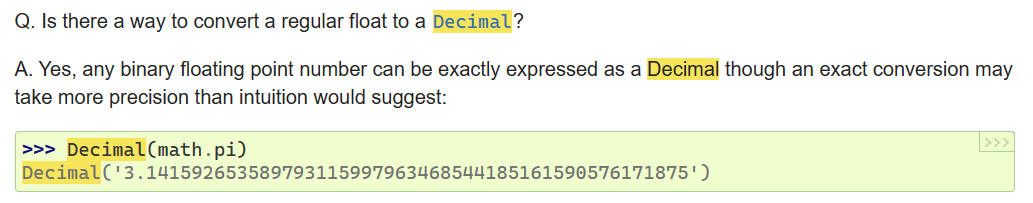


Use quantize function to format Decimal in python 

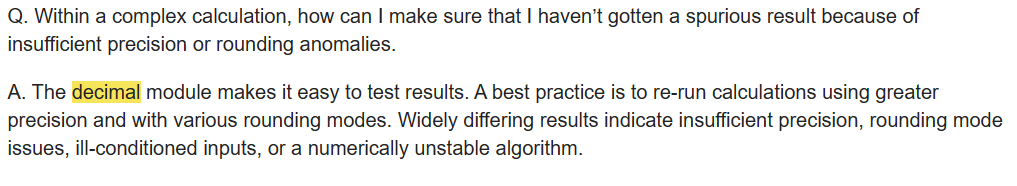
Remove scientific notation in python.

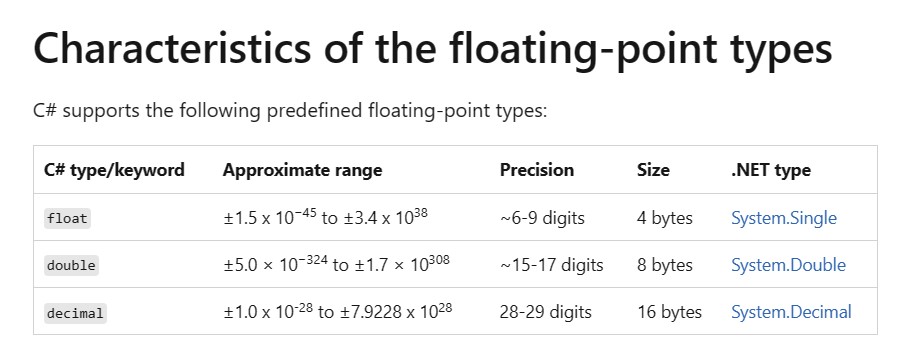


Be aware of your floating type value and the conversion Decimal precision in order to get the exact float value.



Recommendation for Decimal values QA.

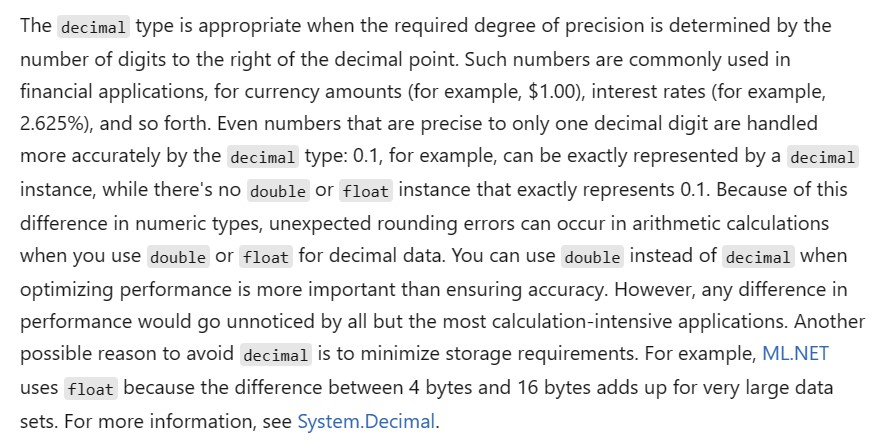


1. .NET floating Type

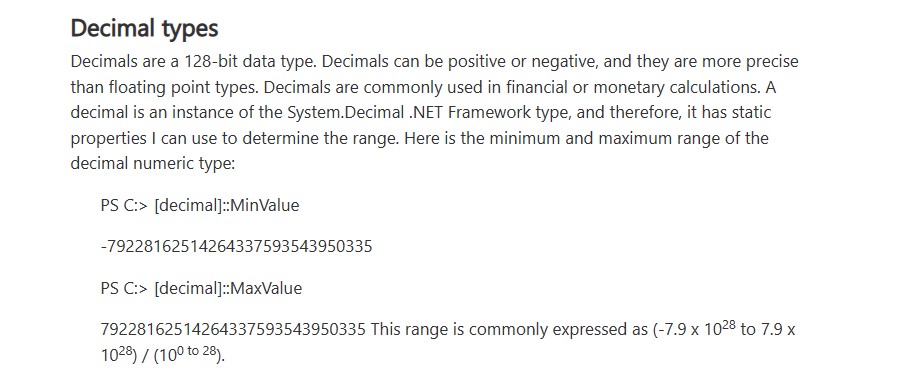
As it shows in the previous image .NET real Floating type ranges are bigger than the real floating point in python. As the precision in .NET use more digits. (~6-9) and (~15-17) instead of (6 and 15) in python.

There are some recommendations for using these real floating types.

It is recommended to use decimal over float and double if accuracy is important and only if your system performance will be affected by a bigger precision. for example, it is recommended for currency even if only need one decimal digit accuracy, because decimal type has exact value 0.1 which is not exact value in float or double.



[Floating-point numeric types - C# reference | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/floating-point-numeric-types)



C) Execution Environment effect on floating types

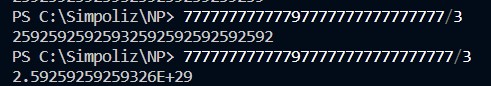
1- .NET uses Decimal with 16 bytes and 29 digits as default, so if the number can be represented by 29 digits no rounding will be done. but if number of digits exceeds 29 digits it will be move to the scientific notation and as double is ~15-17 digits, so result will be rounded to 14 digits after decimal point.

1. in PowerShell (29-digit number divide by 3)

(77777777777797777777777777777/3) = 25925925925932592592592592592

1. in PowerShell (30-digit number divide by 3)

(777777777777977777777777777777/3) = 2.59259259259326E+29



2-python uses double data type (16 digits) as default, and if number exceeds this real floating type range will be rounded to the floating scientific notation.

In python (25-digit number divide by 3) i.e., exceeds python double type range (16-digits)

(7777777777779777777777777/3) = 2.592592592593259E+24

If we cast this scientific notation number to float with Zero decimal places

(7777777777779777777777777/3) = 

Which is different from the result from PowerShell with higher precision after digit number 16 in the result.

3-If we go back to PowerShell with result from python it will never give us exact value 3 again.

=

Even if we cut the number up until 16 digits from python in PowerShell, we never get exact value 3 again.

=

Points to take into consideration in your execution environment:

1. know your calculations results Ranges and not only inputs values data ranges.
2. Know the accuracy level needed in your system calculations.
3. Know the difference in ranges in data types for tools used in your data pipeline.
4. Know the difference between tools and machine used for execution to save floating type to rest. And tools and machine used in testing for re-calculating the floating type for quality assurance purposes.
5. If use less precision floating type, be aware that if testing environment floating type configuration is different from execution environment floating type configuration used to save the floating point at rest, the re-execution can produce less accurate values.
6. Choosing a real Floating type when doing executions on different machines (like Development and production environment) and like (32 bit and 64 bit)
7. Choosing a real Floating type when doing executions on different Tools in your integration pipeline.
8. Choosing a real Floating type when updating tools or machines used in any executions in your integration pipeline.
9. In General, for better accuracy use floating type with bigger precision. Like Decimal in .NET. and Decimal in python.