1.  Both the float and decimal types store numerical values in Python.

Use floats when convenience and speed matter. A float gives you an approximation of the number you declare.

Use decimals when precision matters. Decimals can suffer from their own precision issues, but generally, decimals are more precise than floats. The performance difference between float and decimal, with Python 3, is not outlandish, and in my experience, the precision benefits of a decimal outweigh the performance benefits of a float.

2. Both values are same but internal representation at storage is different. Precision differs, Decimal ('1.200') gives internally 1.200 and Decimal ('1.2') gives 1.2.

3. Both values are checked to be equal, they only differ in precision.

4. Floating-point value is converted to Decimal format. Decimal can store float value with absolute precision. But when float value is given as Decimal object, it first has to be converted from floating point value which might already have rounding error.

Hence it is preferable to start a Decimal object with a string.

5. We can do it with use of Decimal().

6. Arithmetic operations like adding, subtracting or multiplying a Decimal object by a floating-point value is generates an error.

To do these operations, the floating point has to be converted to a Decimal.

7. Value of 0.5 will be represented as ½.

9. Both will be reduced to ½.

10. Fraction class and integer type(int) are related in form of a container. It contains two int, one is the numerator and the other is the denominator.