1. In python 2.7 the return type of a division (/) operation depends on its *operands*. If both operands are of type *int*, floor division is performed and an *int* is returned. If either operand is a *float*,*classic division*is performed, and a *float* is returned. The // operator is also provided for doing floor division no matter what the operands are.

In python 3 division (/) always returns a *float*. To do *floor division* and get an integer result (discarding any fractional result) you need to use // operator.

1. In python 2.7 there is no need for the use of parenthesis while in python 3 we need to use parenthesis.

For example: print “Hello” in python 2.7

And print(‘Hello’) in python 3

1. (In python 2.7) Return the *floating-point value* number rounded to n digits after the decimal point. If n digits are omitted, it defaults to zero. The result is a floating-point number. Values are rounded to the closest multiple of 10 to the power minus n digits; if two multiples are equally close, rounding is done away from 0 (so, for example, round (0.5) is 1.0 and round (-0.5) is -1.0).

(In python 3) Return *number* rounded to n digits precision after the decimal point. If n digits is omitted or is None, it returns the nearest integer to its input.