**How does python handle large numbers greater than 16 bytes?**

Python has a built-in int data type that can handle arbitrarily large integers. This means that Python can handle numbers greater than 16 bytes, as long as the amount of memory required to store the number is available.

In Python 3.x, integers are represented as variable-length arrays of digits in base 2^30. This means that each digit can hold up to 30 bits, and the number of digits required to represent a number depends on its magnitude. This allows Python to represent integers of practically any size, limited only by the amount of available memory.

Python also provides a set of built-in functions and operators that can be used to perform arithmetic operations on large numbers. For example, the +, -, \*, and / operators can be used to perform addition, subtraction, multiplication, and division on large integers.

In addition, Python provides a number of libraries for working with large numbers, including the decimal module for arbitrary-precision decimal arithmetic and the gmpy2 module for high-speed arbitrary-precision integer and floating-point arithmetic. These libraries can be useful when working with extremely large numbers or when high performance is required