**Python Asserts**

The Python unittest library's TestCase class provides a comprehensive set of assertion methods to verify conditions in unit tests. These assertion methods help ensure that code behaves as expected by checking for equality, truthiness, exceptions, membership, and other conditions. Here is a detailed description of all commonly used assert functions in the unittest library:

* assertEqual(a, b, msg=None): Checks that a == b.
* assertNotEqual(a, b, msg=None): Checks that a != b.
* assertTrue(x, msg=None): Checks that bool(x) is True.
* assertFalse(x, msg=None): Checks that bool(x) is False.
* assertIs(a, b, msg=None): Checks that a is b (same object).
* assertIsNot(a, b, msg=None): Checks that a is not b.
* assertIsNone(x, msg=None): Checks that x is None.
* assertIsNotNone(x, msg=None): Checks that x is not None.
* assertIn(a, b, msg=None): Checks that a is in b (membership).
* assertNotIn(a, b, msg=None): Checks that a is not in b.
* assertIsInstance(a, b, msg=None): Checks that isinstance(a, b) is True.
* assertNotIsInstance(a, b, msg=None): Checks that isinstance(a, b) is False.
* assertRaises(exception, callable, \*args, \*\*kwargs): Checks that calling callable with given arguments raises the specified exception.
* assertRaisesRegex(exception, regex, callable, \*args, \*\*kwargs): Checks that calling callable raises exception and the message matches regex.
* assertAlmostEqual(a, b, places=7, msg=None, delta=None): Checks that a and b are approximately equal to the given decimal places or delta.
* assertNotAlmostEqual(a, b, places=7, msg=None, delta=None): Checks that a and b are not approximately equal.
* assertGreater(a, b, msg=None): Checks that a > b.
* assertGreaterEqual(a, b, msg=None): Checks that a >= b.
* assertLess(a, b, msg=None): Checks that a < b.
* assertLessEqual(a, b, msg=None): Checks that a <= b.
* assertRegex(s, regex, msg=None): Checks that regex.search(s) is True.
* assertNotRegex(s, regex, msg=None): Checks that regex.search(s) is False.
* assertCountEqual(a, b, msg=None): Checks that a and b have the same elements in the same number, regardless of order.
* assertMultiLineEqual(a, b, msg=None): Checks that two multi-line strings are equal.
* assertSequenceEqual(a, b, msg=None): Checks that two sequences are equal.
* assertListEqual(a, b, msg=None): Checks that two lists are equal.
* assertTupleEqual(a, b, msg=None): Checks that two tuples are equal.
* assertSetEqual(a, b, msg=None): Checks that two sets are equal.
* assertDictEqual(a, b, msg=None): Checks that two dictionaries are equal.
* assertDictContainsSubset(subset, dictionary, msg=None): Checks that all key/value pairs in subset exist in dictionary.

Each of these assert methods takes optional message arguments which can be used to provide custom failure messages. These assertions form the core of writing effective unit tests in Python using the unittest framework, allowing explicit and expressive verification of expected results in test cases.

⁂