Assert library comparisons

Category	Junit4	Junit5	Truth	Assertj
Basic	assertEquals(“Frodo”, Frodo.getName()	assertEquals(“Frodo”, Frodo.getName()	*assertThat*(frodo.getName()).isEqualTo("Frodo")	*assertThat*(frodo.getName()).isEqualTo("Frodo")
	assertTrue(condition)	assertTrue(condition)		
	assertFalse(condition)	assertFalse(condition)		
	assertNull(object)	assertNull(object)		
	assertNotNull(object)	assertNotNull(object)	*assertThat*(frodo).isNotEqualTo(sauron)	*assertThat*(frodo).isNotEqualTo(sauron)
Extended data methods	N/A	N/A	assertThat(frodo.getName()).startsWith("Fro")	assertThat(frodo.getName()).startsWith("Fro")
			assertThat(frodo.getName()).endsWith("do")	assertThat(frodo.getName()).endsWith("do")
				assertThat(frodo.getName()).isEqualToIgnoringCase("frodo")
			Not supported: isEqualToIgnoringCase	
Array	assertArrayEquals(expectedArray, actualArray)	assertArrayEquals(expectedArray, actualArray)		

Custom message	assertEquals("Custom message", expected, actual)	assertEquals(expected, actual, "Custom message")
		assertEquals(expected, actual, () -> "Custom message")		

Grouped assertions	N/A	assertAll("Grouped assertion", () -> Assertions.assertEquals(expected1, actual1), () -> Assertions.assertEquals(expected2, actual2))	assertAll("Grouped assertion", () -> Assertions.assertEquals(expected1, actual1), () -> Assertions.assertEquals(expected2, actual2))	assertAll("Grouped assertion", () -> Assertions.assertEquals(expected1, actual1), () -> Assertions.assertEquals(expected2, actual2))
Explicit failing test	fail("This test should fail")	fail("This test should fail")	Not supported	fail("This test should fail")

Exception Testing	@Test(expected = IllegalArgumentException.class)
	public void testExceptionWithExpected() {
	throw new IllegalArgumentException("This is an error message"); }
		assertThrows(ExpectedException.class, () -> {
		// code that should throw the exception }
		)
			Not natively supported	assertThatThrownBy(() -> {
				// Code that is expected to throw the exception
				throw new IllegalArgumentException("This is an error message") })
				.isInstanceOf(IllegalArgumentException.class)
				.hasMessageContaining("This is an error message")

Timeout assertions	@Test(timeout = 2000)
	public void testTimeout() throws InterruptedException {
	Thread.sleep(1500); // a delay
	}
		assertTimeout(Duration.ofSeconds(2), () -> {
		// Code that should complete within 2 seconds
		Thread.sleep(1000);
		});
			Not natively supported	Not natively supported

Fluent assertions	N/A	N/A	Limited fluency:
			
			assertThat(fellowshipOfTheRing).hasSize(2);
			assertThat(fellowshipOfTheRing).contains(frodo);
			assertThat(fellowshipOfTheRing).contains(sam);
			assertThat(fellowshipOfTheRing).doesNotContain(Sauron)
				assertThat(frodo.getName()).startsWith("Fro")
				.endsWith("do")
				.isEqualToIgnoringCase("frodo")

Messaging	assertEquals("check " + frodo.getName() + "'s age", frodo.getAge(), 33);	assertEquals(frodo.getAge(), 33, "check " + frodo.getName() + "'s age");	assertThat(frodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33)	assertThat(frodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33)

Soft Asserts	N/A	N/A	SoftAssertions softly = new SoftAssertions();
			softly.assertThat(frodo.getAge()).isNegative();
			softly.assertThat(frodo.getAge()).isEqualTo(32);
			softly.assertAll();
				SoftAssertions softly = new SoftAssertions();
				softly.assertThat(frodo.getAge()).isNegative();
				softly.assertThat(frodo.getAge()).isEqualTo(32);
				softly.assertAll();