Help on class Actor in module domain.Actor:

class Actor(domain.IDObject.IDObject)

| Base class for all objects havin a unique ID within the application

|

| Method resolution order:

| Actor

| domain.IDObject.IDObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, objectID, firstName, lastName, birthday, nationality, residence, height, weight, email, phone)

| Constructor method for building Actor object

| Parameters:

| firstName - Actor's first name (string)

| lastName - Actor's last name (string)

| birthday - Actor's birthday (string)

| nationality - Actor's nationality (string)

| residence - Actor's city of residence (string)

| height - Actor's height in meters (string)

| weigth - Actor's weight in kg (integer number)

| email - Actor's email adderss (string)

| phone - Actor's phone number (string)

|

| \_\_str\_\_(self)

| Prints the object in a given format

| Returns a string containing info about the object

|

| getBirthday(self)

|

| getEmail(self)

|

| getFirstName(self)

|

| getHeight(self)

|

| getLastName(self)

|

| getNationality(self)

|

| getPhone(self)

|

| getResidence(self)

|

| getWeight(self)

|

| setBirthday(self, newBirthday)

|

| setEmail(self, newEmail)

|

| setFirstName(self, newFirstName)

|

| setHeight(self, newHeight)

|

| setLastName(self, newLastName)

|

| setNationality(self, newNationality)

|

| setPhone(self, newPhone)

|

| setResidence(self, newResidence)

|

| setWeight(self, newWeight)

|

| ----------------------------------------------------------------------

| Methods inherited from domain.IDObject.IDObject:

|

| getID(self)

| Returns the object ID

|

| ----------------------------------------------------------------------

| Data descriptors inherited from domain.IDObject.IDObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class ActorValidator in module domain.Actor:

class ActorValidator(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self)

| Validator class for actor, using regex

|

| birthdayValidator(self, birthday)

|

| emailValidator(self, email)

|

| firstNameValidator(self, firstName)

|

| heightValidator(self, height)

|

| idValidator(self, id)

|

| lastNameValidator(self, lastName)

|

| nationalityValidator(self, nationality)

|

| phoneValidator(self, phone)

|

| residenceValidator(self, residence)

|

| validate(self, actor)

|

| weightValidator(self, weight)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class Movie in module domain.Movie:

class Movie(domain.IDObject.IDObject)

| Base class for all objects havin a unique ID within the application

|

| Method resolution order:

| Movie

| domain.IDObject.IDObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, objectID, name, year, website)

| Constructor method for building Actor object

| Parameters:

| name - Movie's name (string)

| year - Movie's apparition year (integer)

| website - Movie website (string)

|

| \_\_str\_\_(self)

| Prints the object in a given format

| Returns a string containing info about the object

|

| getName(self)

|

| getWebsite(self)

|

| getYear(self)

|

| setName(self, newName)

|

| setWebsite(self, newWebsite)

|

| setYear(self, newYear)

|

| ----------------------------------------------------------------------

| Methods inherited from domain.IDObject.IDObject:

|

| getID(self)

| Returns the object ID

|

| ----------------------------------------------------------------------

| Data descriptors inherited from domain.IDObject.IDObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class MovieValidator in module domain.Movie:

class MovieValidator(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self)

| Validator class for movie, using regex

|

| idValidator(self, id)

|

| nameValidator(self, name)

|

| validate(self, movie)

|

| websiteValidator(self, website)

|

| yearValidator(self, year)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class IDObject in module domain.IDObject:

class IDObject(builtins.object)

| Base class for all objects havin a unique ID within the application

|

| Methods defined here:

|

| \_\_init\_\_(self, objectID)

| Constructor method for building IDObject

| objectID - the unique ID of the object in the application

|

| getID(self)

| Returns the object ID

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class ValidatorException in module domain.ValidatorException:

class ValidatorException(builtins.Exception)

| Common base class for all non-exit exceptions.

|

| Method resolution order:

| ValidatorException

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, messageList=['Validation error!'])

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_str\_\_(self)

| Return str(self).

|

| getMessage(self)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| helper for pickle

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

Help on class CastRepository in module repository.CastRepo:

class CastRepository(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self)

| Constructor for repository class

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| addActor(self, idMovie, idActor)

| searches in the repo for the movie having idMovie(parameter)

| when it finds the movie, it goes to the list associated with the movie, containing the actors

| appends in the list of actors, idActor(parameter)

|

| addMovie(self, idMovie)

| adds to the repository the id of the movie(parameter)

| then, adds the list of actors corresponding to the movie

|

| checkID(self, idMovie, idActor)

| searches if an idActor(parameter) has already been added for a idMovie(parameter)

| return 1 if so, 0 otherwise

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| lengthList(self, idMovie)

| returns the length of the actors' list for movie havinf idMovie(parameter)

|

| removeActor(self, idActor)

| searches in every the list of actor ids the idActor(parameter)

| when it finds the idActor in a list it removes it

|

| removeActorFromMovie(self, idMovie, idActor)

| searches in movie list the ids the idActor(parameter)

| when it finds the idActor in a list it removes it

|

| removeMovie(self, idMovie)

| searches for the idMovie(parameter) in the repo

| when it finds it, it removes it, together with the list of actors associated to it

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class FileCastRepository in module repository.CastRepo:

class FileCastRepository(CastRepository)

| Method resolution order:

| FileCastRepository

| CastRepository

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, filename)

| Constructor for repository with file class

| filename - name of the file containing data

| loadFromFile - method - loads data into the application

|

| addActor(self, idMovie, idActor)

| Add a movie to the repository

| movie - Movie to be added

|

| addMovie(self, idMovie)

| Add a movie to the repository

| movie - Movie to be added

|

| removeActor(self, idActor)

| remove movie by id

| id - movie id

|

| removeActorFrommovie(self, idMovie, idActor)

| remove actor by id from movie list

| isMovie - movie id

| idActor - actor id

|

| removeMovie(self, idMovie)

| remove movie by id

| id - movie id

|

| ----------------------------------------------------------------------

| Methods inherited from CastRepository:

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| checkID(self, idMovie, idActor)

| searches if an idActor(parameter) has already been added for a idMovie(parameter)

| return 1 if so, 0 otherwise

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| lengthList(self, idMovie)

| returns the length of the actors' list for movie havinf idMovie(parameter)

|

| removeActorFromMovie(self, idMovie, idActor)

| searches in movie list the ids the idActor(parameter)

| when it finds the idActor in a list it removes it

|

| ----------------------------------------------------------------------

| Data descriptors inherited from CastRepository:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class MovieRepository in module repository.MovieRepo:

class MovieRepository(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self)

| Constructor for repository class

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| add(self, movie)

| Add a movie to the repository

| movie - Movie to be added

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| removeID(self, id)

| remove movie by id

| id - movie id

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class FileMovieRepository in module repository.MovieRepo:

class FileMovieRepository(MovieRepository)

| Method resolution order:

| FileMovieRepository

| MovieRepository

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, filename)

| Constructor for repository with file class

| filename - name of the file containing data

| loadFromFile - method - loads data into the application

|

| add(self, movie)

| Add a movie to the repository

| movie - Movie to be added

|

| removeID(self, id)

| remove movie by id

| id - movie id

|

| ----------------------------------------------------------------------

| Methods inherited from MovieRepository:

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| ----------------------------------------------------------------------

| Data descriptors inherited from MovieRepository:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class ActorRepository in module repository.ActorRepo:

class ActorRepository(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self)

| Constructor for repository class

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| add(self, actor)

| Add an actor to the repository

| actor - Actor to be added

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| removeID(self, id)

| remove actor by id

| id - Actor's id

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class FileActorRepository in module repository.ActorRepo:

class FileActorRepository(ActorRepository)

| Method resolution order:

| FileActorRepository

| ActorRepository

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, filename)

| Constructor for repository with file class

| filename - name of the file containing data

| loadFromFile - method - loads data into application

|

| add(self, actor)

| Add an actor to the repository

| actor - Actor to be added

|

| removeID(self, id)

| remove actor by id

| id - Actor's id

|

| ----------------------------------------------------------------------

| Methods inherited from ActorRepository:

|

| \_\_len\_\_(self)

| Overriding the len() built-in function

|

| getAll(self)

| Return all repository data

| Returns the live list of the repository

|

| ----------------------------------------------------------------------

| Data descriptors inherited from ActorRepository:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class CastController in module controller.CastController:

class CastController(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self, repo, repoActor)

| Constructor for cast

|

| addActor(self, idMovie, idActor)

| adds an actor to movie cast

| parameters: idMovie - movie id, idActor - actor id

|

| addMovie(self, idMovie)

| Adds a movie id

| parameters: idMovie - movie id

|

| checkAddActor(self, idMovie, idActor)

| checks if actor id is a valid id (there actually is an actor with that id)

| if yes, it adds it, otherwise, no

| parameters: idMovie - movie id, idActor - actor id

|

| checkAddToCast(self, idMovie, idActor)

| Checks if id actor already exists in movie list

| if it does not, it adds it

| parameters: idMovie - movie id, idActor - actor id

|

| checkID(self, idMovie, idActor)

| Performn the repository check if idActor already exists in id movie

| parameters: idMovie - movie id, idActor - actor id

|

| getAll(self)

| Gets all items in the repository

|

| lengthList(self, idMovie)

| Returns the length of a given movie cast list

| parameters: idMovie - movie id

|

| removeActor(self, idActor)

| parameters: idActor - actor id

| removes actor by id

|

| removeActorFromMovie(self, idMovie, idActor)

| removes actor from movie

| parameters: idMovie - movie id, idActor - actor id

|

| removeMovie(self, idMovie)

| Removes a movie by id

| parameters: idMovie - movie id

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class MovieController in module controller.MovieController:

class MovieController(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self, repo, repoCast)

| initialize controller

|

| add(self, movie)

| Adds movie

| movie - movie to be added

|

| addMovie(self, id, name, year, website)

| checks if movie is valid and creates it if so

| parameters: movie characteristics: id, name, year, website

| if error: prints message

|

| checkID(self, id)

| ckecks if id exists

| id - the given id

| return 0 if id exists, 1 otherwise

|

| getAll(self)

| Gets all items in the repository

|

| removeID(self, id)

| removes movie by id

| id - movie id

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class ActorController in module controller.ActorController:

class ActorController(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self, repo)

| initialize controller

|

| add(self, actor)

| adds an actor to the repository

| actor - actor to be added

|

| addActor(self, id, firstName, lastName, birthday, nationality, residence, height, weigth, email, phone)

| checks if actor is valid and creates it if so

| parameters: actor characteristics: id, firstName, lastName, birthday, nationality, residence, height, weigth, email, phone

| if error: prints message

|

| checkID(self, id)

| ckecks if id exists

| id - the given id

| return 0 if id exists, 1 otherwise

|

| getAll(self)

| Gets all items in the repository

|

| removeID(self, id)

| removes actor by id

| id - actor's id

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

Help on class ControllerException in module controller.MovieController:

class ControllerException(builtins.Exception)

| Exception class for controller errors

|

| Method resolution order:

| ControllerException

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, message)

| Constructor for controller exception class

| message - A string representing the exception message

|

| \_\_str\_\_(self)

| Return str(self).

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| helper for pickle

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

Help on class RepositoryTest in module test.CastTest:

class RepositoryTest(unittest.case.TestCase)

| Unit test case for repository and controller

|

| Method resolution order:

| RepositoryTest

| unittest.case.TestCase

| builtins.object

|

| Methods defined here:

|

| setUp(self)

| Hook method for setting up the test fixture before exercising it.

|

| testCast(self)

| tests addMovie, addActor, removeMovie, removeActor, getAll, len for current repository

|

| ----------------------------------------------------------------------

| Methods inherited from unittest.case.TestCase:

|

| \_\_call\_\_(self, \*args, \*\*kwds)

| Call self as a function.

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, methodName='runTest')

| Create an instance of the class that will use the named test

| method when executed. Raises a ValueError if the instance does

| not have a method with the specified name.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_(self)

| Return str(self).

|

| addCleanup(self, function, \*args, \*\*kwargs)

| Add a function, with arguments, to be called when the test is

| completed. Functions added are called on a LIFO basis and are

| called after tearDown on test failure or success.

|

| Cleanup items are called even if setUp fails (unlike tearDown).

|

| addTypeEqualityFunc(self, typeobj, function)

| Add a type specific assertEqual style function to compare a type.

|

| This method is for use by TestCase subclasses that need to register

| their own type equality functions to provide nicer error messages.

|

| Args:

| typeobj: The data type to call this function on when both values

| are of the same type in assertEqual().

| function: The callable taking two arguments and an optional

| msg= argument that raises self.failureException with a

| useful error message when the two arguments are not equal.

|

| assertAlmostEqual(self, first, second, places=None, msg=None, delta=None)

| Fail if the two objects are unequal as determined by their

| difference rounded to the given number of decimal places

| (default 7) and comparing to zero, or by comparing that the

| between the two objects is more than the given delta.

|

| Note that decimal places (from zero) are usually not the same

| as significant digits (measured from the most signficant digit).

|

| If the two objects compare equal then they will automatically

| compare almost equal.

|

| assertAlmostEquals = deprecated\_func(\*args, \*\*kwargs)

|

| assertCountEqual(self, first, second, msg=None)

| An unordered sequence comparison asserting that the same elements,

| regardless of order. If the same element occurs more than once,

| it verifies that the elements occur the same number of times.

|

| self.assertEqual(Counter(list(first)),

| Counter(list(second)))

|

| Example:

| - [0, 1, 1] and [1, 0, 1] compare equal.

| - [0, 0, 1] and [0, 1] compare unequal.

|

| assertDictContainsSubset(self, subset, dictionary, msg=None)

| Checks whether dictionary is a superset of subset.

|

| assertDictEqual(self, d1, d2, msg=None)

|

| assertEqual(self, first, second, msg=None)

| Fail if the two objects are unequal as determined by the '=='

| operator.

|

| assertEquals = deprecated\_func(\*args, \*\*kwargs)

|

| assertFalse(self, expr, msg=None)

| Check that the expression is false.

|

| assertGreater(self, a, b, msg=None)

| Just like self.assertTrue(a > b), but with a nicer default message.

|

| assertGreaterEqual(self, a, b, msg=None)

| Just like self.assertTrue(a >= b), but with a nicer default message.

|

| assertIn(self, member, container, msg=None)

| Just like self.assertTrue(a in b), but with a nicer default message.

|

| assertIs(self, expr1, expr2, msg=None)

| Just like self.assertTrue(a is b), but with a nicer default message.

|

| assertIsInstance(self, obj, cls, msg=None)

| Same as self.assertTrue(isinstance(obj, cls)), with a nicer

| default message.

|

| assertIsNone(self, obj, msg=None)

| Same as self.assertTrue(obj is None), with a nicer default message.

|

| assertIsNot(self, expr1, expr2, msg=None)

| Just like self.assertTrue(a is not b), but with a nicer default message.

|

| assertIsNotNone(self, obj, msg=None)

| Included for symmetry with assertIsNone.

|

| assertLess(self, a, b, msg=None)

| Just like self.assertTrue(a < b), but with a nicer default message.

|

| assertLessEqual(self, a, b, msg=None)

| Just like self.assertTrue(a <= b), but with a nicer default message.

|

| assertListEqual(self, list1, list2, msg=None)

| A list-specific equality assertion.

|

| Args:

| list1: The first list to compare.

| list2: The second list to compare.

| msg: Optional message to use on failure instead of a list of

| differences.

|

| assertLogs(self, logger=None, level=None)

| Fail unless a log message of level \*level\* or higher is emitted

| on \*logger\_name\* or its children. If omitted, \*level\* defaults to

| INFO and \*logger\* defaults to the root logger.

|

| This method must be used as a context manager, and will yield

| a recording object with two attributes: `output` and `records`.

| At the end of the context manager, the `output` attribute will

| be a list of the matching formatted log messages and the

| `records` attribute will be a list of the corresponding LogRecord

| objects.

|

| Example::

|

| with self.assertLogs('foo', level='INFO') as cm:

| logging.getLogger('foo').info('first message')

| logging.getLogger('foo.bar').error('second message')

| self.assertEqual(cm.output, ['INFO:foo:first message',

| 'ERROR:foo.bar:second message'])

|

| assertMultiLineEqual(self, first, second, msg=None)

| Assert that two multi-line strings are equal.

|

| assertNotAlmostEqual(self, first, second, places=None, msg=None, delta=None)

| Fail if the two objects are equal as determined by their

| difference rounded to the given number of decimal places

| (default 7) and comparing to zero, or by comparing that the

| between the two objects is less than the given delta.

|

| Note that decimal places (from zero) are usually not the same

| as significant digits (measured from the most signficant digit).

|

| Objects that are equal automatically fail.

|

| assertNotAlmostEquals = deprecated\_func(\*args, \*\*kwargs)

|

| assertNotEqual(self, first, second, msg=None)

| Fail if the two objects are equal as determined by the '!='

| operator.

|

| assertNotEquals = deprecated\_func(\*args, \*\*kwargs)

|

| assertNotIn(self, member, container, msg=None)

| Just like self.assertTrue(a not in b), but with a nicer default message.

|

| assertNotIsInstance(self, obj, cls, msg=None)

| Included for symmetry with assertIsInstance.

|

| assertNotRegex(self, text, unexpected\_regex, msg=None)

| Fail the test if the text matches the regular expression.

|

| assertNotRegexpMatches = deprecated\_func(\*args, \*\*kwargs)

|

| assertRaises(self, expected\_exception, \*args, \*\*kwargs)

| Fail unless an exception of class expected\_exception is raised

| by the callable when invoked with specified positional and

| keyword arguments. If a different type of exception is

| raised, it will not be caught, and the test case will be

| deemed to have suffered an error, exactly as for an

| unexpected exception.

|

| If called with the callable and arguments omitted, will return a

| context object used like this::

|

| with self.assertRaises(SomeException):

| do\_something()

|

| An optional keyword argument 'msg' can be provided when assertRaises

| is used as a context object.

|

| The context manager keeps a reference to the exception as

| the 'exception' attribute. This allows you to inspect the

| exception after the assertion::

|

| with self.assertRaises(SomeException) as cm:

| do\_something()

| the\_exception = cm.exception

| self.assertEqual(the\_exception.error\_code, 3)

|

| assertRaisesRegex(self, expected\_exception, expected\_regex, \*args, \*\*kwargs)

| Asserts that the message in a raised exception matches a regex.

|

| Args:

| expected\_exception: Exception class expected to be raised.

| expected\_regex: Regex (re pattern object or string) expected

| to be found in error message.

| args: Function to be called and extra positional args.

| kwargs: Extra kwargs.

| msg: Optional message used in case of failure. Can only be used

| when assertRaisesRegex is used as a context manager.

|

| assertRaisesRegexp = deprecated\_func(\*args, \*\*kwargs)

|

| assertRegex(self, text, expected\_regex, msg=None)

| Fail the test unless the text matches the regular expression.

|

| assertRegexpMatches = deprecated\_func(\*args, \*\*kwargs)

|

| assertSequenceEqual(self, seq1, seq2, msg=None, seq\_type=None)

| An equality assertion for ordered sequences (like lists and tuples).

|

| For the purposes of this function, a valid ordered sequence type is one

| which can be indexed, has a length, and has an equality operator.

|

| Args:

| seq1: The first sequence to compare.

| seq2: The second sequence to compare.

| seq\_type: The expected datatype of the sequences, or None if no

| datatype should be enforced.

| msg: Optional message to use on failure instead of a list of

| differences.

|

| assertSetEqual(self, set1, set2, msg=None)

| A set-specific equality assertion.

|

| Args:

| set1: The first set to compare.

| set2: The second set to compare.

| msg: Optional message to use on failure instead of a list of

| differences.

|

| assertSetEqual uses ducktyping to support different types of sets, and

| is optimized for sets specifically (parameters must support a

| difference method).

|

| assertTrue(self, expr, msg=None)

| Check that the expression is true.

|

| assertTupleEqual(self, tuple1, tuple2, msg=None)

| A tuple-specific equality assertion.

|

| Args:

| tuple1: The first tuple to compare.

| tuple2: The second tuple to compare.

| msg: Optional message to use on failure instead of a list of

| differences.

|

| assertWarns(self, expected\_warning, \*args, \*\*kwargs)

| Fail unless a warning of class warnClass is triggered

| by the callable when invoked with specified positional and

| keyword arguments. If a different type of warning is

| triggered, it will not be handled: depending on the other

| warning filtering rules in effect, it might be silenced, printed

| out, or raised as an exception.

|

| If called with the callable and arguments omitted, will return a

| context object used like this::

|

| with self.assertWarns(SomeWarning):

| do\_something()

|

| An optional keyword argument 'msg' can be provided when assertWarns

| is used as a context object.

|

| The context manager keeps a reference to the first matching

| warning as the 'warning' attribute; similarly, the 'filename'

| and 'lineno' attributes give you information about the line

| of Python code from which the warning was triggered.

| This allows you to inspect the warning after the assertion::

|

| with self.assertWarns(SomeWarning) as cm:

| do\_something()

| the\_warning = cm.warning

| self.assertEqual(the\_warning.some\_attribute, 147)

|

| assertWarnsRegex(self, expected\_warning, expected\_regex, \*args, \*\*kwargs)

| Asserts that the message in a triggered warning matches a regexp.

| Basic functioning is similar to assertWarns() with the addition

| that only warnings whose messages also match the regular expression

| are considered successful matches.

|

| Args:

| expected\_warning: Warning class expected to be triggered.

| expected\_regex: Regex (re pattern object or string) expected

| to be found in error message.

| args: Function to be called and extra positional args.

| kwargs: Extra kwargs.

| msg: Optional message used in case of failure. Can only be used

| when assertWarnsRegex is used as a context manager.

|

| assert\_ = deprecated\_func(\*args, \*\*kwargs)

|

| countTestCases(self)

|

| debug(self)

| Run the test without collecting errors in a TestResult

|

| defaultTestResult(self)

|

| doCleanups(self)

| Execute all cleanup functions. Normally called for you after

| tearDown.

|

| fail(self, msg=None)

| Fail immediately, with the given message.

|

| failIf = deprecated\_func(\*args, \*\*kwargs)

|

| failIfAlmostEqual = deprecated\_func(\*args, \*\*kwargs)

|

| failIfEqual = deprecated\_func(\*args, \*\*kwargs)

|

| failUnless = deprecated\_func(\*args, \*\*kwargs)

|

| failUnlessAlmostEqual = deprecated\_func(\*args, \*\*kwargs)

|

| failUnlessEqual = deprecated\_func(\*args, \*\*kwargs)

|

| failUnlessRaises = deprecated\_func(\*args, \*\*kwargs)

|

| id(self)

|

| run(self, result=None)

|

| shortDescription(self)

| Returns a one-line description of the test, or None if no

| description has been provided.

|

| The default implementation of this method returns the first line of

| the specified test method's docstring.

|

| skipTest(self, reason)

| Skip this test.

|

| subTest(self, msg=None, \*\*params)

| Return a context manager that will return the enclosed block

| of code in a subtest identified by the optional message and

| keyword parameters. A failure in the subtest marks the test

| case as failed but resumes execution at the end of the enclosed

| block, allowing further test code to be executed.

|

| tearDown(self)

| Hook method for deconstructing the test fixture after testing it.

|

| ----------------------------------------------------------------------

| Class methods inherited from unittest.case.TestCase:

|

| setUpClass() from builtins.type

| Hook method for setting up class fixture before running tests in the class.

|

| tearDownClass() from builtins.type

| Hook method for deconstructing the class fixture after running all tests in the class.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from unittest.case.TestCase:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from unittest.case.TestCase:

|

| failureException = <class 'AssertionError'>

| Assertion failed.

|

| longMessage = True

|

| maxDiff = 640

Help on class UI in module ui.Menu:

class UI(builtins.object)

| Methods defined here:

|

| \_\_init\_\_(self, actors, movies, cast)

| Constructor class for ui

| parameters: actor repo, movie repo, cast repo

|

| mainMenu(self)

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| printCast(list, listMovies, listActors)

|

| printList(list)

|

| printMenu()

|

| readActor()

|

| readMovie()

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)