

```
import unittest

class TestStrings(unittest.TestCase):
    def test_strings(self):
        """
        A basic introduction to strings
        """
        str_dir = ['__add__', '__class__', '__contains__', '__delattr__',
            '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
            '__getslice__', '__gt__', '__hash__', '__init__', '__le__', '__len__', '__lt__', '__mod__', '__mul__',
            '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__',
            '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '_formatter_field_name_split',
            '_formatter_parser', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format',
            'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower',
            'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
            'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper',
            'zfill']

        # Create the variable ``hol a`` and assign 'hello world'
        *****
        hol a = 'hello world'
        self.assertEqual(hol a, """"hello world""")
        self.assertIsInstance(hol a, str)

        # Create a string, ``hol a2`` that equals ``hol a`` multiplied by 2
        *****
        hol a2 = hol a * 2
        self.assertEqual(hol a2, 'hello worldhello world')

        # Triple quoted strings (with ''' or """) allow embedded of
        # both single and double quotes.
        # Create a triple quoted string
        # ``darwin_quote2`` that has the following content:
        # Darwin said, "There is grandeur in this view of things."
        *****
        darwin_quote2 = 'Darwin said, "There is grandeur in this view of things."'
        self.assertEqual(darwin_quote2, 'Darwin said, "There is grandeur in this
view of things."')

        # Assign the method names of a string to a variable ``string_methods``
        # use ``dir()`` to list them
        *****
        string_methods = dir(str)
        self.assertEqual(string_methods, str_dir)

        # Create a variable where_is_gra that has holds the index of the
        # substring "gra" in the string darwin_quote2. (Find a string method to
        # figure it out)
        *****
        where_is_gra = darwin_quote2.find("gra")
        self.assertEqual(where_is_gra, 23)

if __name__ == '__main__':
    unittest.main()
```