# Q & A -

**Q.1**  How to run a specific method before each test method in a class using fixture ?
**Ans -**

```
@pytest.fixture(autouse=True)
Def specific_method(self):
      ... block of  code ...
Def method_1(self):
      ... block of  code ...
Def method_2(self):
      ... block of  code ...
```

Note - `@pytest.fixture(scope = 'Function' , autouse=True)` - Here `scope = 'Function'` will be default if not added.

---

**Q.2**  How to run a specific method before all the test methods in a class using fixture ?
**Ans -**

```
@pytest.fixture(scope = 'class', autouse=True)
Def specific_method(self):
      ... block of  code ...
Def method_1(self):
      ... block of  code ...
Def method_2(self):
      ... block of  code ...
```

---

**Q.3**  How to run a specific method before all the classes using fixture ?
**Ans -**

Create a conftest.py file and write code -

```
Import pytest

@pytest.fixture()
Def myfixture():
      ... block of  code ...
```

---

**Q.4**  How to use dependency in Pytest ?
Ans -

Q.5    What is a finalizer and how to use it inside a fixture ?
Ans -

```python
@pytest.fixture(scope = 'class', autouse=True)
Def fixture_method(self, request):
      … block of  code …

      Def teardown():
            … block of  code …
      request.addfinalizer(teardown)


Def method_1(self):
      … block of  code …
Def method_2(self):
      … block of  code …
```

---

Q.6    How to call a fixture from a test method ?

```python
@pytest.fixture()
Def fixture_method(self):
      … block of  code …

Def method_1(self,fixture_method):
      … block of  code …
Def method_2(self,fixture_method):
      … block of  code …
```

---

Q.7     How to use a fixture inside another fixture ?

```python
@pytest.fixture()
Def fixture_method_0(self):
      … block of  code …

@pytest.fixture()
Def fixture_method_1(self,fixture_method_0):
      … block of  code …
```

---

Q.8    How to run the same test method for multiple set of test data  OR
       How to do parameterization in the Pytest framework ?

**Ans -**

```
@pytest.mark.parametrize("Astro", ["test_1","test_2","test_3"])
Def test_method(self, Astro):
          … block of  code …
```

---

References -
- https://www.javainuse.com/home
- https://www.lambdatest.com/learning-hub/pytest-interview-questions
- https://readthedocs.org/projects/pytest-dependency/