

---

```
1 # Demonstrates defining a function with a return value
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 main()
```

```
1 # Demonstrates defining a function with a return value
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

```
1  from calculator1 import square
2
3
4  def main():
5      test_square()
6
7
8  def test_square():
9      if square(2) != 4:
10         print("2 squared was not 4")
11         if square(3) != 9:
12             print("3 squared was not 9")
13
14
15  if __name__ == "__main__":
16     main()
```

```
1 # Demonstrates defining a function with a return value
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

```
1 from calculator2 import square
2
3
4 def main():
5     test_square()
6
7
8 def test_square():
9     assert square(2) == 4
10    assert square(3) == 9
11
12
13 if __name__ == "__main__":
14    main()
```

```
1 # Demonstrates defining a function with a return value
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

```
1  from calculator3 import square
2
3
4  def main():
5      test_square()
6
7
8  def test_square():
9      try:
10         assert square(2) == 4
11     except AssertionError:
12         print("2 squared was not 4")
13     try:
14         assert square(3) == 9
15     except AssertionError:
16         print("3 squared was not 9")
17
18
19  if __name__ == "__main__":
20     main()
```

---

```
1 # Demonstrates defining a function with a return value
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```



```
1 from calculator4 import square
2
3
4 def main():
5     test_square()
6
7
8 def test_square():
9     try:
10        assert square(2) == 4
11    except AssertionError:
12        print("2 squared was not 4")
13    try:
14        assert square(3) == 9
15    except AssertionError:
16        print("3 squared was not 9")
17    try:
18        assert square(-2) == 4
19    except AssertionError:
20        print("-2 squared was not 4")
21    try:
22        assert square(-3) == 9
23    except AssertionError:
24        print("-3 squared was not 9")
25    try:
26        assert square(0) == 0
27    except AssertionError:
28        print("0 squared was not 0")
29
30
31 if __name__ == "__main__":
32     main()
```

```
1 # Tests a function with one function via pytest
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

---

```
1 from calculator5 import square
2
3
4 def test_square():
5     assert square(2) == 4
6     assert square(3) == 9
7     assert square(-2) == 4
8     assert square(-3) == 9
9     assert square(0) == 0
```

---

```
1 # Tests a function with multiple functions via pytest
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

```
1  from calculator6 import square
2
3
4  def test_positive():
5      assert square(1) == 1
6      assert square(2) == 4
7      assert square(3) == 9
8
9
10 def test_negative():
11     assert square(-1) == 1
12     assert square(-2) == 4
13     assert square(-3) == 9
14
15
16 def test_zero():
17     assert square(0) == 0
```

```
1 # Tests a function with multiple functions via pytest
2
3
4 def main():
5     x = int(input("What's x? "))
6     print("x squared is", square(x))
7
8
9 def square(n):
10    return n * n
11
12
13 if __name__ == "__main__":
14    main()
```

```
1 import pytest
2
3 from calculator import square
4
5
6 def test_positive():
7     assert square(2) == 4
8     assert square(3) == 9
9
10
11 def test_negative():
12     assert square(-2) == 4
13     assert square(-3) == 9
14
15
16 def test_zero():
17     assert square(0) == 0
18
19
20 def test_str():
21     with pytest.raises(TypeError):
22         square("cat")
```

---

```
1 # Function to be tested
2
3
4 def main():
5     name = input("What's your name? ")
6     hello(name)
7
8
9 def hello(to="world"):
10    print("hello,", to)
11
12
13 if __name__ == "__main__":
14    main()
```



---

```
1 # Has function return a str instead
2
3
4 def main():
5     name = input("What's your name? ")
6     print(hello(name))
7
8
9 def hello(to="world"):
10     return f"hello, {to}"
11
12
13 if __name__ == "__main__":
14     main()
```

---

```
1 from hello1 import hello
2
3
4 def test_default():
5     assert hello() == "hello, world"
6
7
8 def test_argument():
9     assert hello("David") == "hello, David"
```

---

```
1 from hello1 import hello
2
3
4 def test_default():
5     assert hello() == "hello, world"
6
7
8 def test_argument():
9     for name in ["Hermione", "Harry", "Ron"]:
10        assert hello(name) == f"hello, {name}"
```

---

```
1 from hello1 import hello
2
3
4 def test_default():
5     assert hello() == "hello, world"
6
7
8 def test_argument():
9     assert hello("David") == "hello, David"
```

