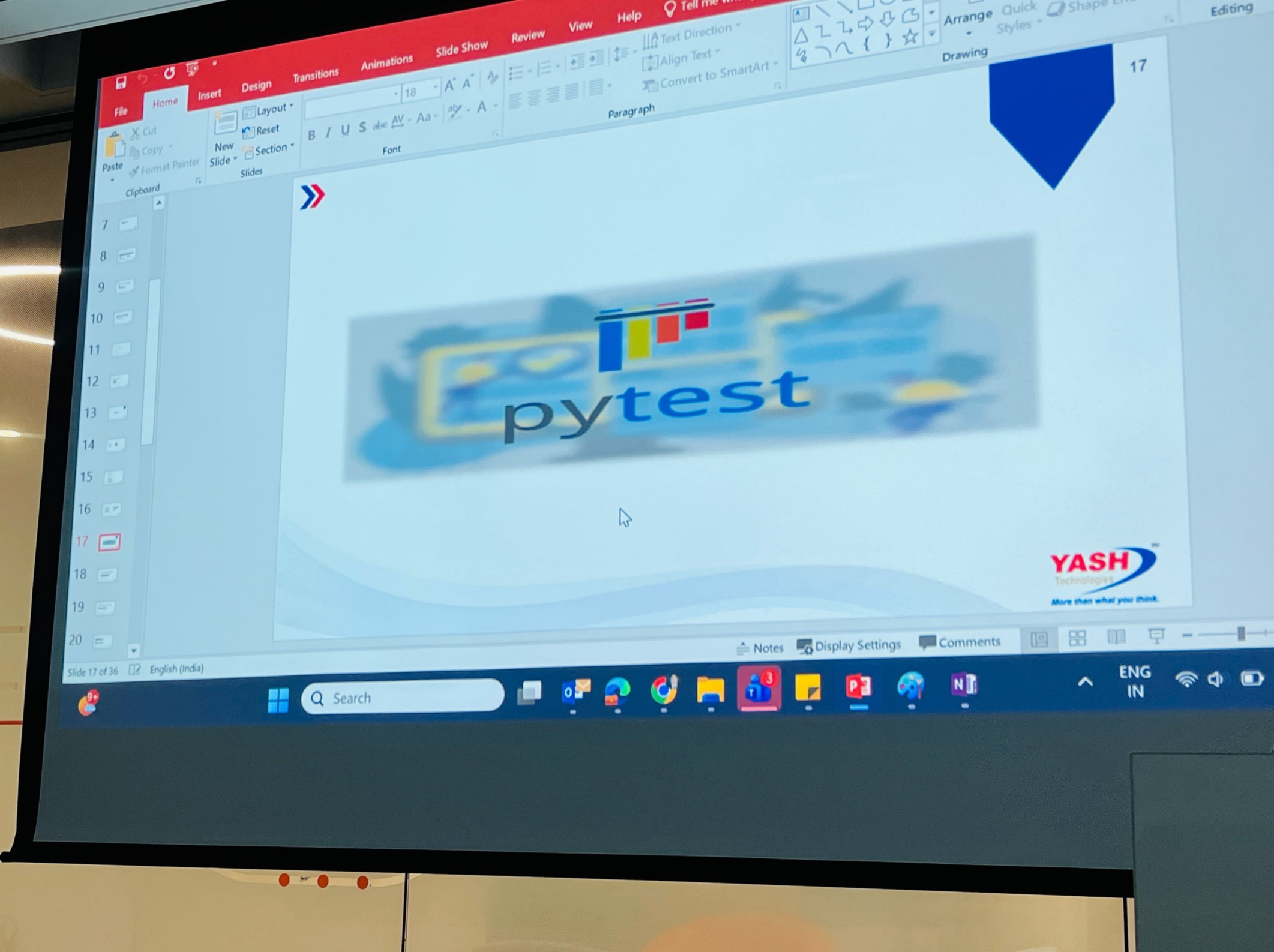

pytest
17
YASH
Technologies
Move then what you think.
Slide 17 of 36
English (India)
Notes
Display Settings
Comments
Search
ENG
IN

Python Testing With Pytest

Pytest – a powerful, modern, and easy-to-use testing framework that supports simple test functions, advanced features like fixtures, parameterization, mocking, and rich reporting.

Among these, **pytest** is the most popular choice today because it allows developers to:

- Write clean, readable test cases with minimal boilerplate code.
- Automatically discover tests without extra configuration.
- Reuse setup code efficiently using fixtures.
- Run tests with detailed output and integrate easily with CI/CD tools.

In short, **pytest** makes testing in Python faster, simpler, and more productive—helping teams maintain reliable and maintainable codebases.

YASH Technologies
More than what you think.

1-Pytest - PowerPoint

Tell me what you want to do

File Home Insert Design Transitions Animations Slide Show Review View Help

Cut Copy Format Painter Paste Clipboard

New Slide Layout Reset Section Slides

Font Paragraph Drawing Arrange Quick Styles Shape Fill Shape Outline Shape Effects

Text Direction Align Text Convert to SmartArt

19

pytest Test Discovery Conventions

Pytest automatically discovers and runs tests based on naming conventions and file structure.
Here's how it works:

Default Test Discovery Behaviour

- If no arguments are passed, pytest searches for tests in:
- The directories listed in **testpaths** (if configured in pytest.ini), or
- The **current working directory** by default.
- Pytest looks for files that match these naming patterns:
- test_*.py or *_test.py

YASH Technologies
More than what you think.

Notes Display Settings Comments

Slide 19 of 36 English (India)

Search

ENG IN

File Discovery Rules

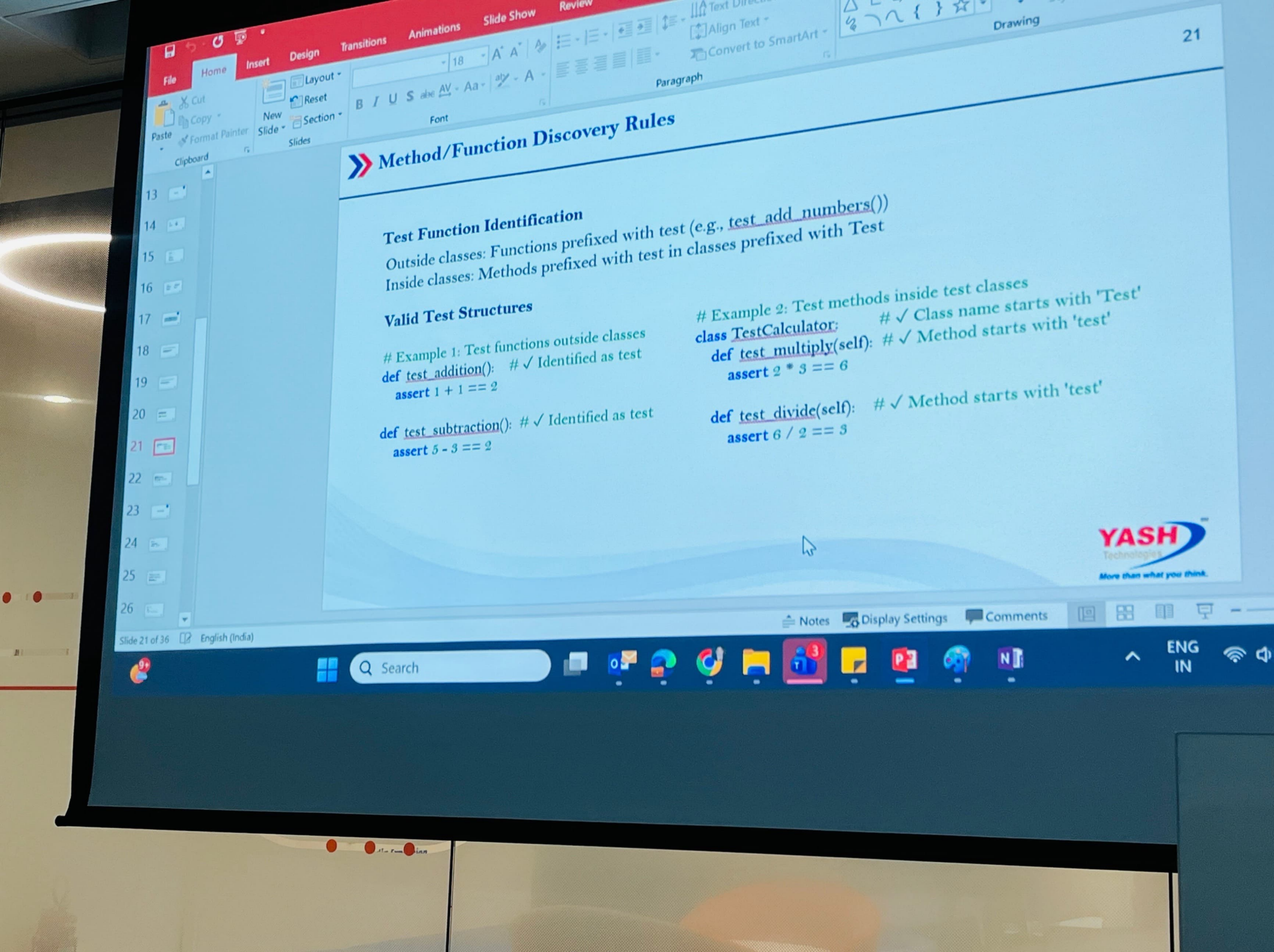
Test File Naming Pattern

pytest automatically identifies test files that match:

Starting with: test_*.py (e.g., test_calculator.py)
Ending with: *_test.py (e.g., calculator_test.py)

Examples of Valid Test Files

```
test_calculator.py    # ✓ Matches - starts with 'test_'
calculator_test.py    # ✓ Matches - ends with '_test'
test_api.py           # ✓ Matches - starts with 'test_'
```


Method/Function Discovery Rules
Test Function Identification
Outside classes: Functions prefixed with test (e.g., test_add_numbers())
Inside classes: Methods prefixed with test in classes prefixed with Test
Valid Test Structures
Example 1: Test functions outside classes
def test_addition(): # ✓ Identified as test
assert 1 + 1 == 2
def test_subtraction(): # ✓ Identified as test
assert 5 - 3 == 2
Example 2: Test methods inside test classes
class TestCalculator: # ✓ Class name starts with 'Test'
def test_multiply(self): # ✓ Method starts with 'test'
assert 2 * 3 == 6
def test_divide(self): # ✓ Method starts with 'test'
assert 6 / 2 == 3
YASH Technologies
More than what you think.
21
Slide 21 of 36
English (India)
Notes
Display Settings
Comments
Search
ENG IN

Important Restrictions

Class Requirement

Test classes must not have an __init__ method.
Class names must start with Test (e.g., TestUser, TestAPI).

Method Name Enforcement

- **Only methods starting with "test"** are recognized as tests.
- **Non-test-prefixed methods** are ignored, even if explicitly requested to run.

Examples of Ignored Code

```
def helper_function():  # ✗ Ignored - doesn't start with 'test'
    return 42

class TestMath:
    def setup_data(self):    # ✗ Ignored - doesn't start with 'test'
        self.data = [1, 2, 3]

    def test_calculation(self):  # ✓ Identified as test
        assert 2 + 2 == 4
```

YASH Technologies
More than what you think


1-Pytest - PowerPoint
Tell me what you want to do
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Paste
Cut
Copy
Format Painter
Clipboard
New Slide
Layout
Reset
Section
Slides
Font
Paragraph
Text Direction
Align Text
Convert to SmartArt
Drawing
Arrange
Quick Styles
Shape Fill
Shape Outline
Shape Effects
Find
Replace
Select
Editing
23
Running Tests In Pytest
YASH
Technologies
More than what you think.
Slide 23 of 36
English (India)
Notes
Display Settings
Comments
Search
ENG
IN

Running Tests In Pytest

Running All Tests

> pytest

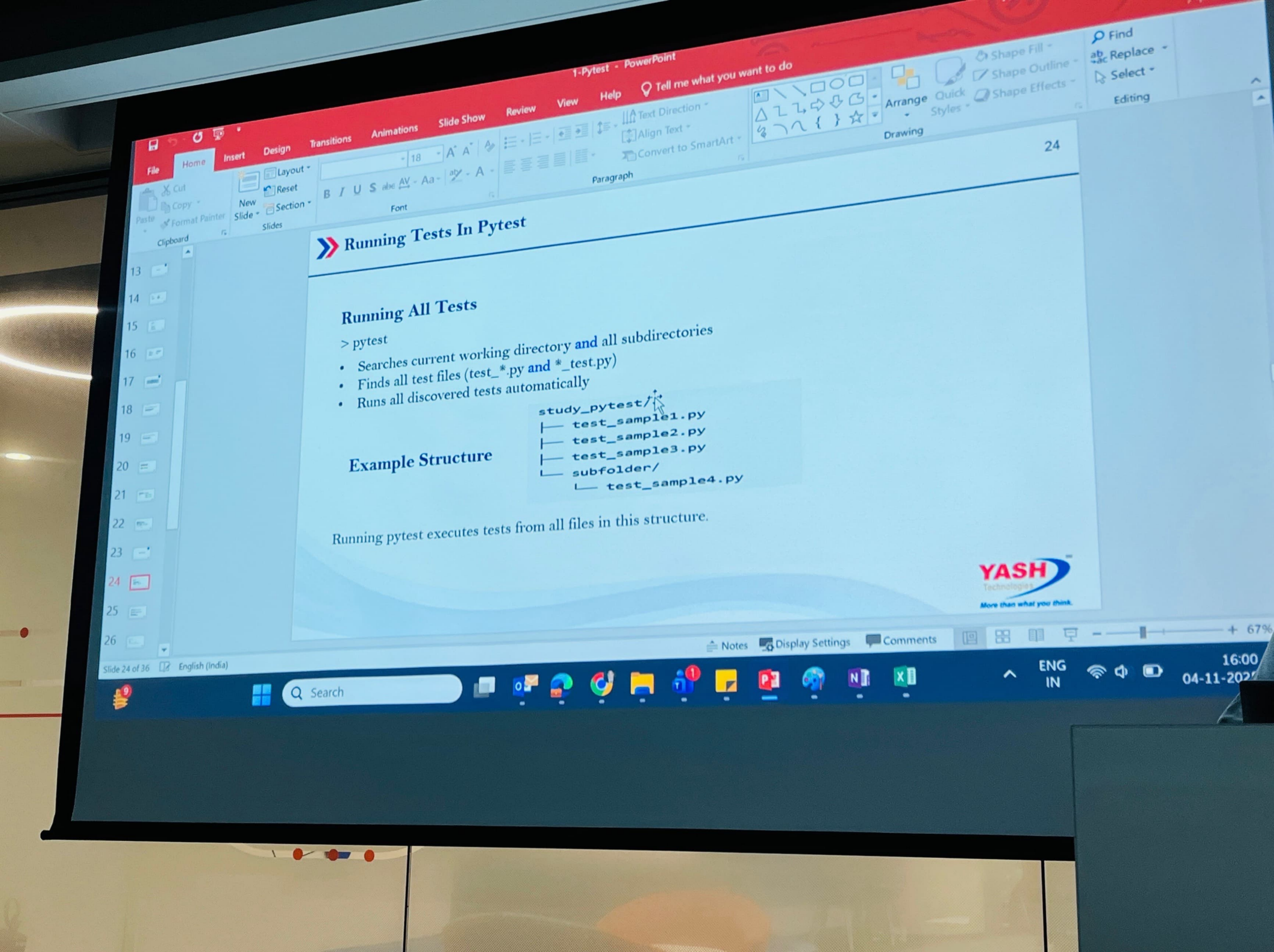
- Searches current working directory and all subdirectories
- Finds all test files (test_*.py and *_test.py)
- Runs all discovered tests automatically

Example Structure

```
study_pytest/
├── test_sample1.py
├── test_sample2.py
├── test_sample3.py
└── subfolder/
    └── test_sample4.py
```

Running pytest executes tests from all files in this structure.

YASH Technologies
More than what you think.


1-Pytest - PowerPoint
Tell me what you want to do
Running Tests In Pytest
24
Running All Tests
> pytest
Searches current working directory and all subdirectories
Finds all test files (test_*.py and *_test.py)
Runs all discovered tests automatically
Example Structure
study_pytest/
├── test_sample1.py
├── test_sample2.py
├── test_sample3.py
└── subfolder/
└── test_sample4.py
Running pytest executes tests from all files in this structure.
YASH
Technologies
More than what you think.
Slide 24 of 36
English (India)
Notes
Display Settings
Comments
67%
Search
ENG
IN
16:00

1-Pytest - PowerPoint

Tell me what you want to do

Share

25

Running Specific Tests

1. Run Specific Test File

```
> pytest test_sample1.py    # Runs all tests from a single specified file.
```

2. Run Specific Test Function

```
> pytest test_sample1.py::test_min
```

Runs only the test_min function from test_sample1.py using the :: separator.

3. Run Using Markers

```
>pytest -m smoke
```

Groups tests using custom markers
Runs only tests marked with @pytest.mark.smoke

YASH Technologies
More than what you think.

Slide 25 of 36 English (India)

Notes Display Settings Comments

67%

Search

ENG IN

16:00
04-11-202

1-Pytest - PowerPoint

File Home Insert Design Transitions Animations Slide Show Review View Help Tell me what you want to do

Cut Copy Format Painter Paste Clipboard

New Slide Layout Reset Section Slides

Font Paragraph Text Direction Align Text Convert to SmartArt

Arrange Quick Styles Shape Fill Shape Outline Shape Effects Drawing

Find Replace Select Editing

26

Run Using Keyword Expressions

> pytest -k "expression"

- Runs tests that match the expression in their names.
- Flexible filtering based on test function/**class** names.

Examples:

```
pytest -k "test_min"     # Runs tests containing "test_min"
pytest -k "min or max"   # Runs tests with "min" OR "max"
pytest -k "not slow"     # Excludes tests with "slow"
pytest -k "login and smoke" # Runs tests with both "login" AND "smoke"
```

YASH Technologies
More than what you think.

13 14 15 16 17 18 19 20 21 22 23 24 25 26

Slide 26 of 36 English (India)

Notes Display Settings Comments

Search

ENG IN

04-1

Controlling Test Execution

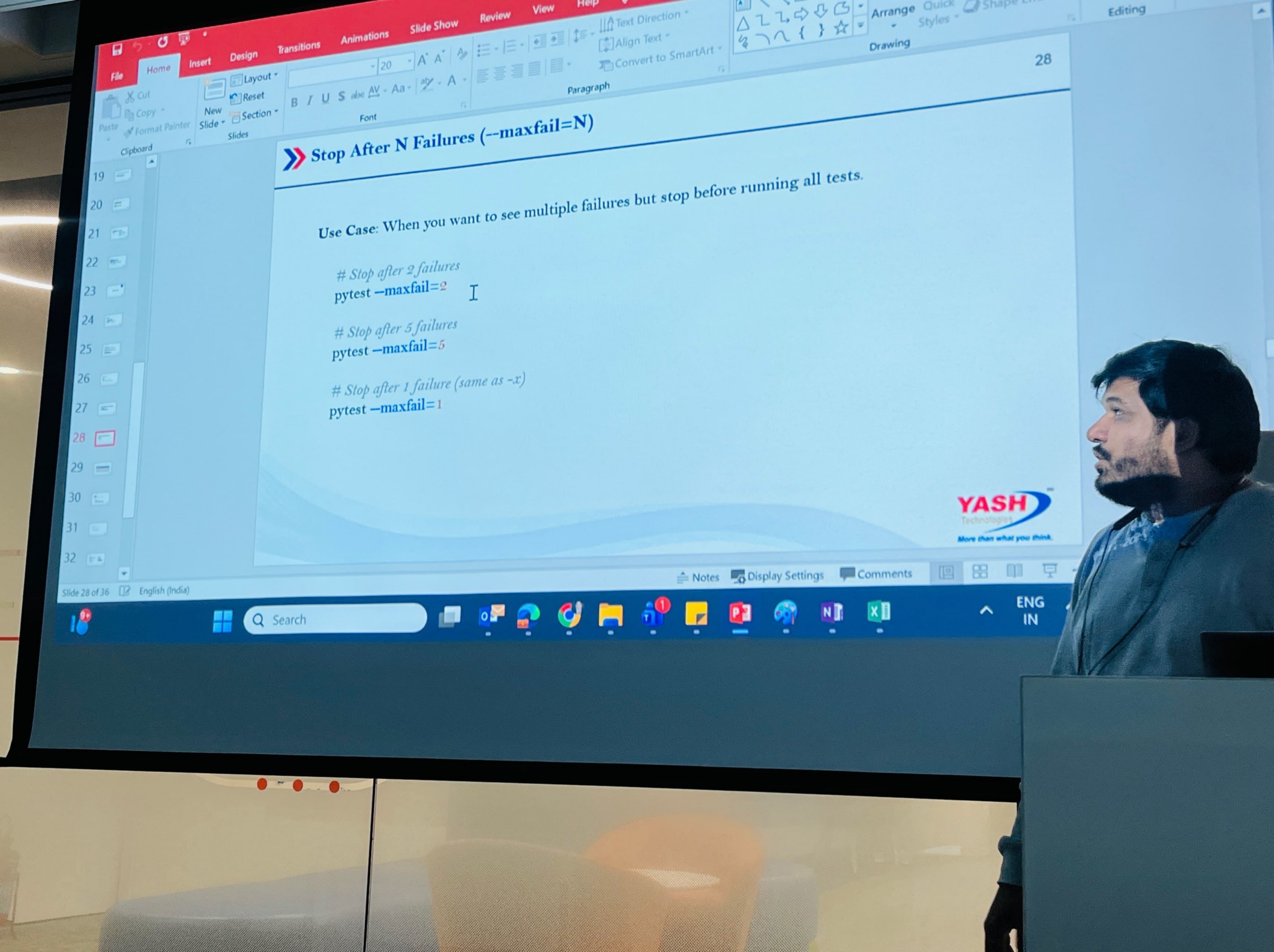
1. **Stop After First Failure (-x or –exitfirst)**

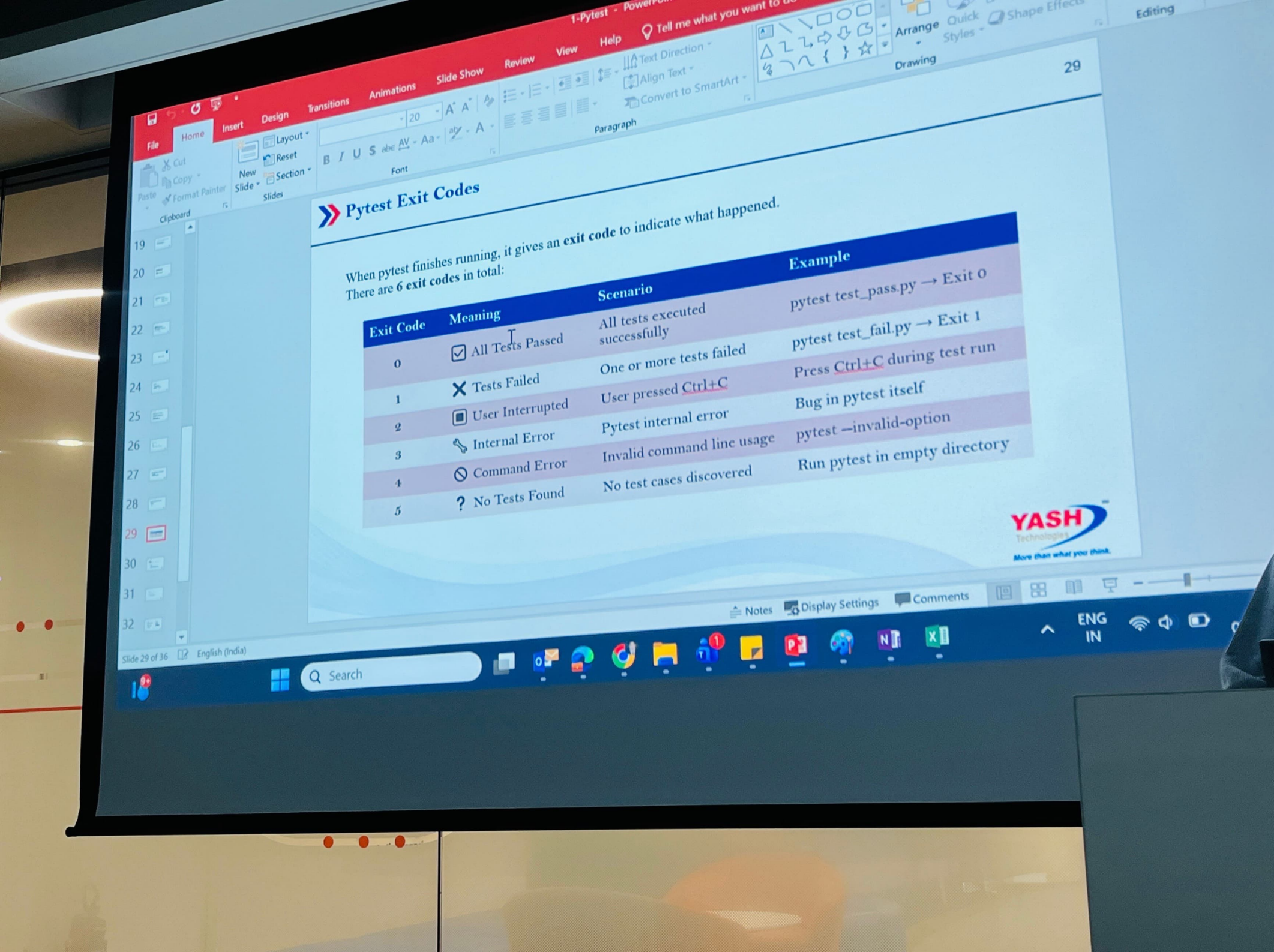
Use Case: When you want to stop immediately at the first test failure to fix it right away.

Run with stop after first failure:

```
•pytest -x test_sample.py
  or
pytest –exitfirst
```

Note: Pytest will stop immediately after test_case2 fails.


Stop After N Failures (--maxfail=N)
Use Case: When you want to see multiple failures but stop before running all tests.
Stop after 2 failures
pytest --maxfail=2
Stop after 5 failures
pytest --maxfail=5
Stop after 1 failure (same as -x)
pytest --maxfail=1
28
YASH Technologies
More than what you think.
Slide 28 of 36
English (India)
Notes
Display Settings
Comments
Search
ENG
IN


1-Pytest - PowerPoint
Tell me what you want to do
File Home Insert Design Transitions Animations Slide Show Review View Help
Cut Copy Format Painter Paste
Clipboard
New Slide Layout Reset Section
Slides
Font
Paragraph
Text Direction Align Text Convert to SmartArt
Arrange Quick Styles Shape Effects
Drawing
Editing
29
Pytest Exit Codes
When pytest finishes running, it gives an exit code to indicate what happened.
There are 6 exit codes in total:
Exit Code | Meaning | Scenario | Example
0 | All Tests Passed | All tests executed successfully | pytest test_pass.py → Exit 0
1 | Tests Failed | One or more tests failed | pytest test_fail.py → Exit 1
2 | User Interrupted | User pressed Ctrl+C | Press Ctrl+C during test run
3 | Internal Error | Pytest internal error | Bug in pytest itself
4 | Command Error | Invalid command line usage | pytest --invalid-option
5 | No Tests Found | No test cases discovered | Run pytest in empty directory
YASH Technologies
More than what you think.
Notes Display Settings Comments
Slide 29 of 36 English (India)
Search
ENG IN

Installation & Verification

Installing Pytest

- pip install pytest

To test weather pytest is installed or not use the following command.

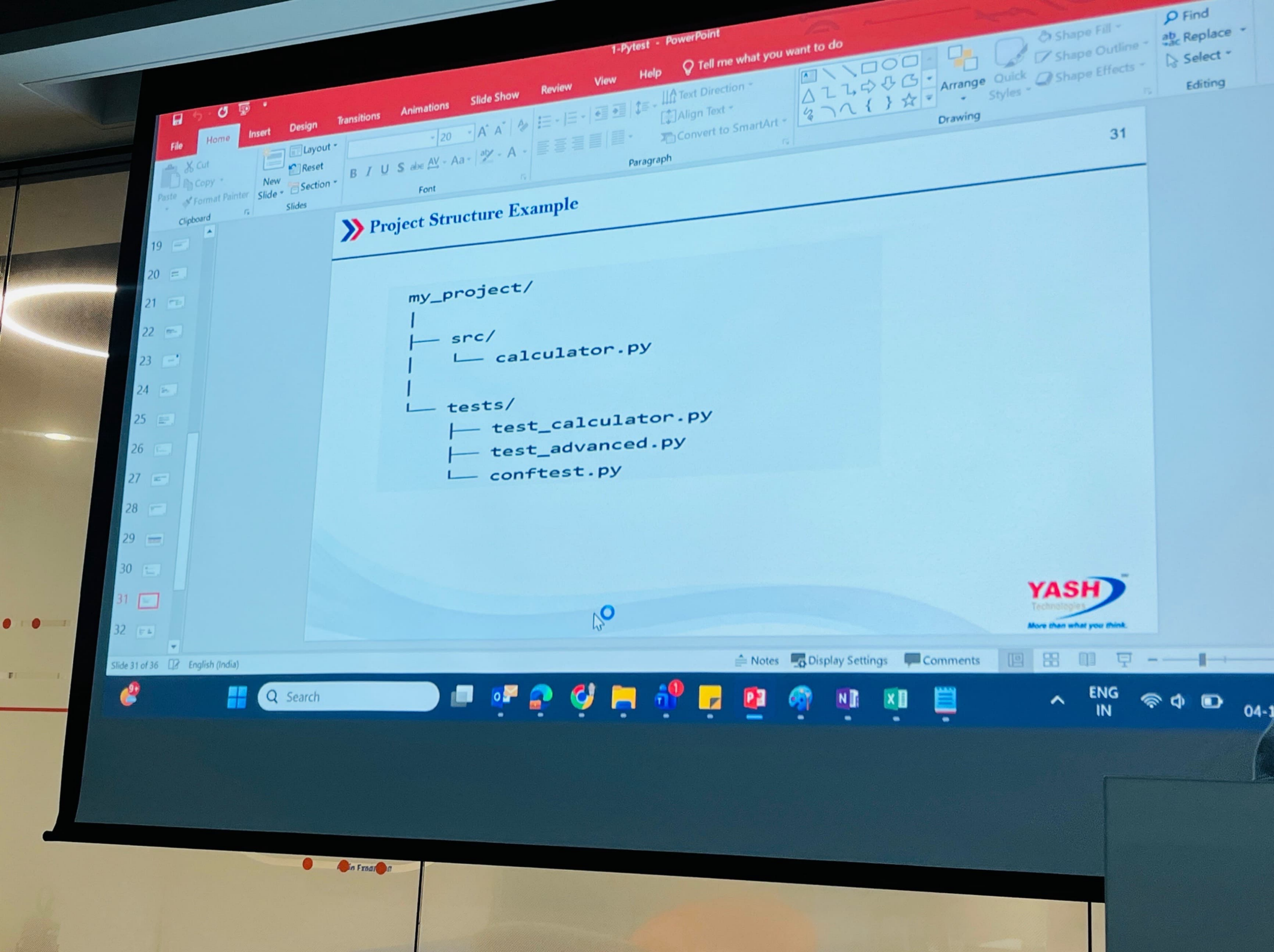
- pytest -h

Ex1:

```
import pytest
def test_method1():
    num=3
    num2=4
    assert num+1 == num2, "Test Passed"
    assert num == num2, "Test Failed"
```

Run: pytest basic.py or pytest if file name is test_basic.py

YASH Technologies
More than what you think.


1-Pytest - PowerPoint
Tell me what you want to do
Project Structure Example
31
my_project/
├── src/
│ └── calculator.py
└── tests/
├── test_calculator.py
├── test_advanced.py
└── conftest.py
YASH
Technologies
Slide 31 of 36
English (India)
Notes
Display Settings
Comments
Search
ENG
IN

Pytest Ex-2

```
import pytest
def test_case1():
    num1=5
    num2=2
    assert num2+3 == num1, "First test failed"
    assert num1 == num2, "Test failed, num1 is not eq to num2"

def test_case2():
    lang= "python"
    assert lang.upper() == 'PYTHON'

def test_case3():
    list_val=[1,2,3,4,5]
    assert len(list_val)==5

def test_case4():
    assert True

def test_Case5():
    assert False
```

```
>pytest basic.py
```

YASH Technologies
More than what you think

Pytest Ex-3:

basic.py

```
def max(values):
    _max = values[0]
    for val in values:
        if val > _max:
            _max = val
    return _max

def min(values):

    _min = values[0]
    for val in values:
        if val < _min:
            _min = val
    return _min
```

min_max_test.py

```
import basic
def test_min():
    values = (2, 3, 1, 4, 6)
    val = basic.min(values)
    assert val == 1

def test_max():
    values = (2, 3, 1, 4, 6)
    val = basic.max(values)
    assert val == 6
```

Run: pytest min_max_test.py

YASH Technologies
Move than what you think.

Pytest Ex- File:

Note- For file we have to name start with test.

Ex: test_demo1.py test_Dem2.py

test_demo1.py

```
import pytest
def test_case1():
    num1=5
    num2=2
    assert num2+3 == num1, "First test failed"
    assert num1 == num2, "Test failed, num1 is not eq to num2"

def test_case2():
    lang= "python"
    assert lang.upper() == 'PYTHON'
```

test_demo2.py

```
def test_case3():
    list_val=[1,2,3,4,5]
    assert len(list_val)==5

def test_case4():
    assert True

def test_Case5():
    assert False
```

Run: py.test or pytest

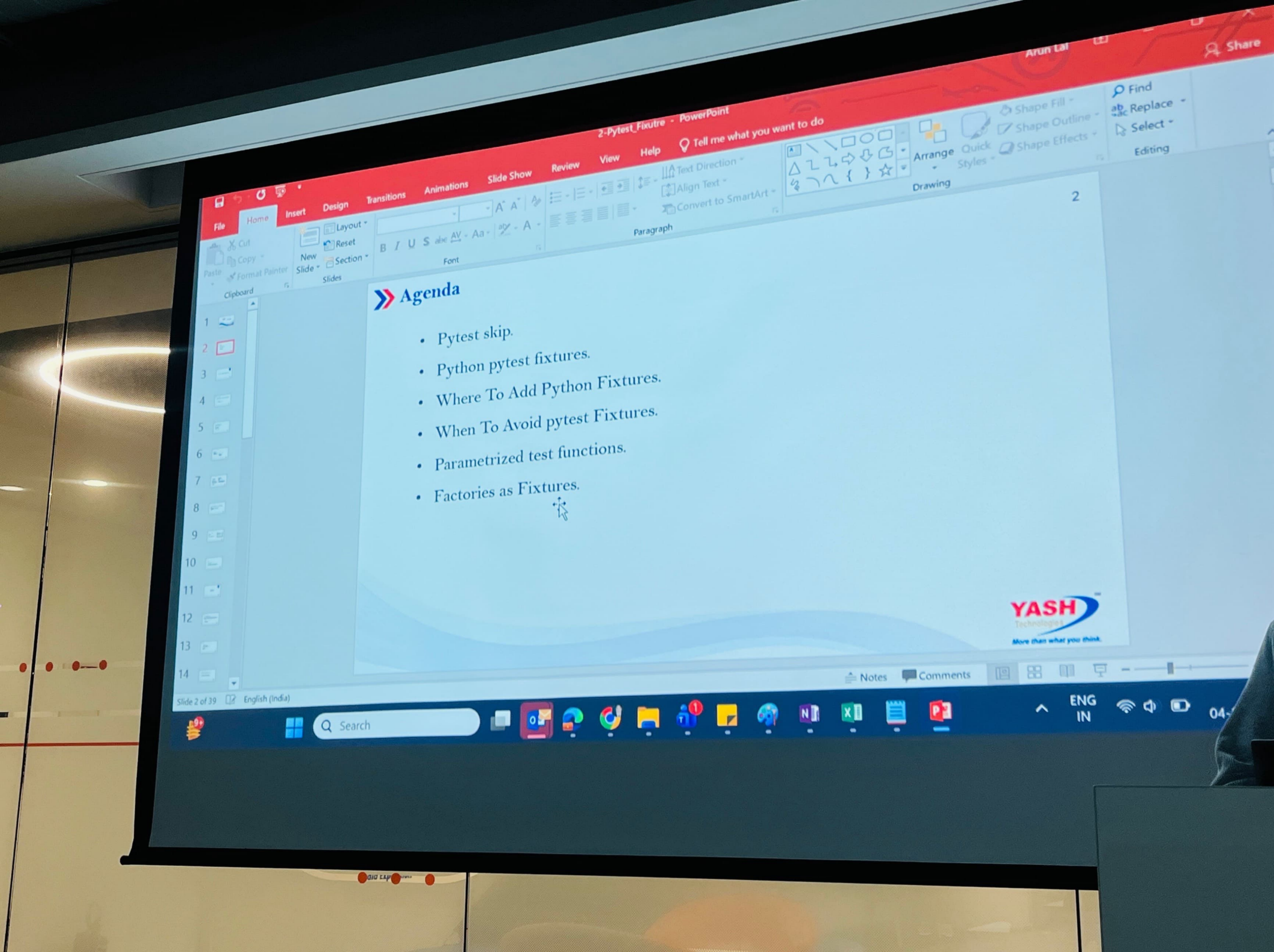
Run a particular file:

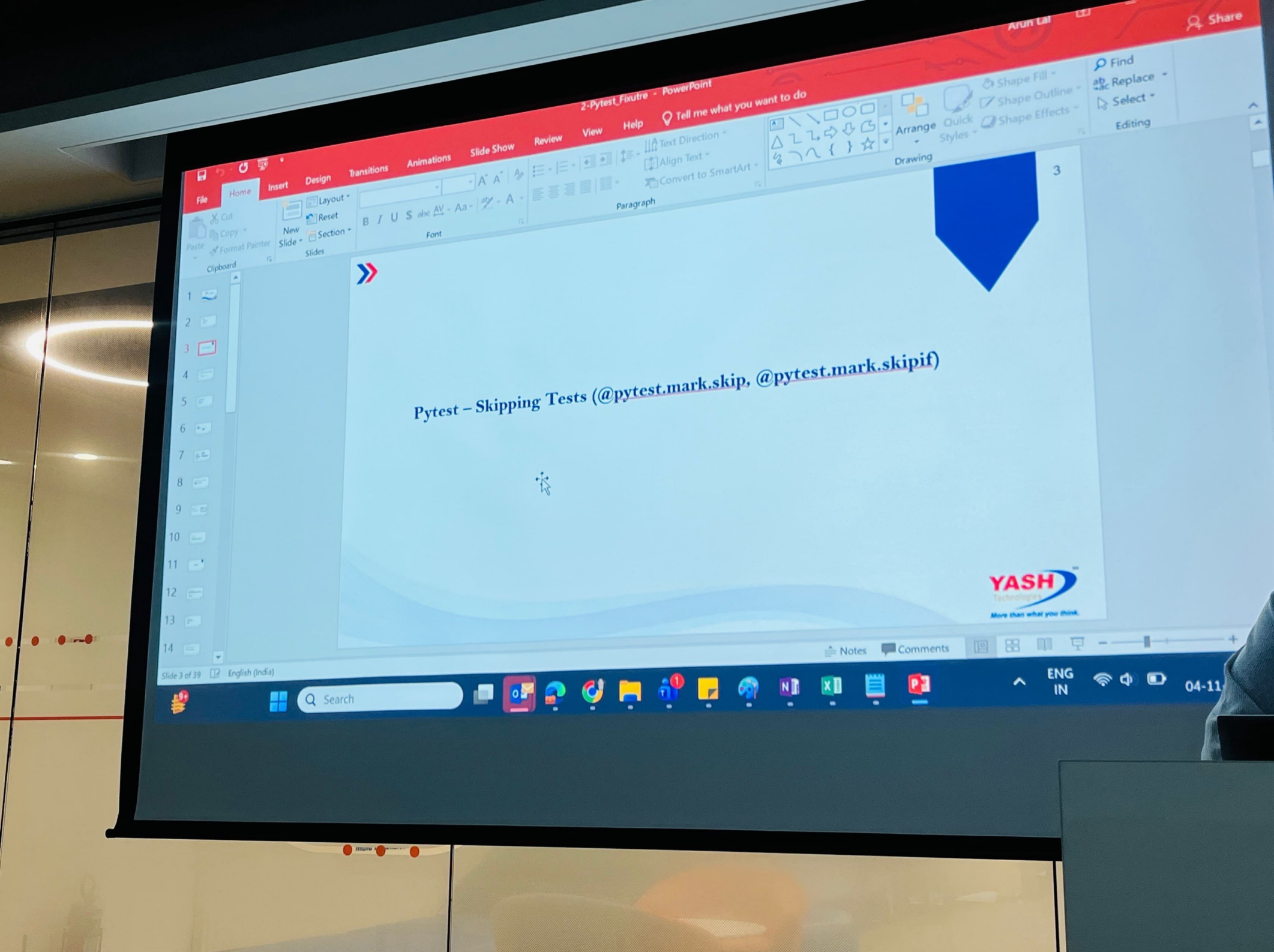
- py.test test_demo1.py

Run from a particular folder file:

- py.test auth/test_demo2.py

YASH Technologies
More than what you think.


2-Pytest_Fixutre - PowerPoint
Tell me what you want to do
Share
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Paste
Cut
Copy
Format Painter
Clipboard
New Slide
Layout
Reset
Section
Slides
Font
Paragraph
Text Direction
Align Text
Convert to SmartArt
Arrange
Quick Styles
Shape Fill
Shape Outline
Shape Effects
Drawing
Find
Replace
Select
Editing
2
Agenda
Pytest skip.
Python pytest fixtures.
Where To Add Python Fixtures.
When To Avoid pytest Fixtures.
Parametrized test functions.
Factories as Fixtures.
YASH
Technologies
Move than what you think.
Notes
Comments
Slide 2 of 39
English (India)
Search
ENG
IN


2-Pytest_Fixutre - PowerPoint
Tell me what you want to do
Share
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Clipboard
Slides
Font
Paragraph
Drawing
Editing
Find
Replace
Select
Arrange
Quick Styles
Shape Fill
Shape Outline
Shape Effects
Text Direction
Align Text
Convert to SmartArt
3
Pytest – Skipping Tests (@pytest.mark.skip, @pytest.mark.skipif)
YASH
More than what you think.
Notes
Comments
Slide 3 of 39
English (India)
Search
ENG
IN

Pytest – Skipping Tests

With the skip decorator, we can skip the specified tests. There are multiple reasons for skipping test; for instance, a database/online service is not available at the moment, or we skip Linux specific tests on Windows.

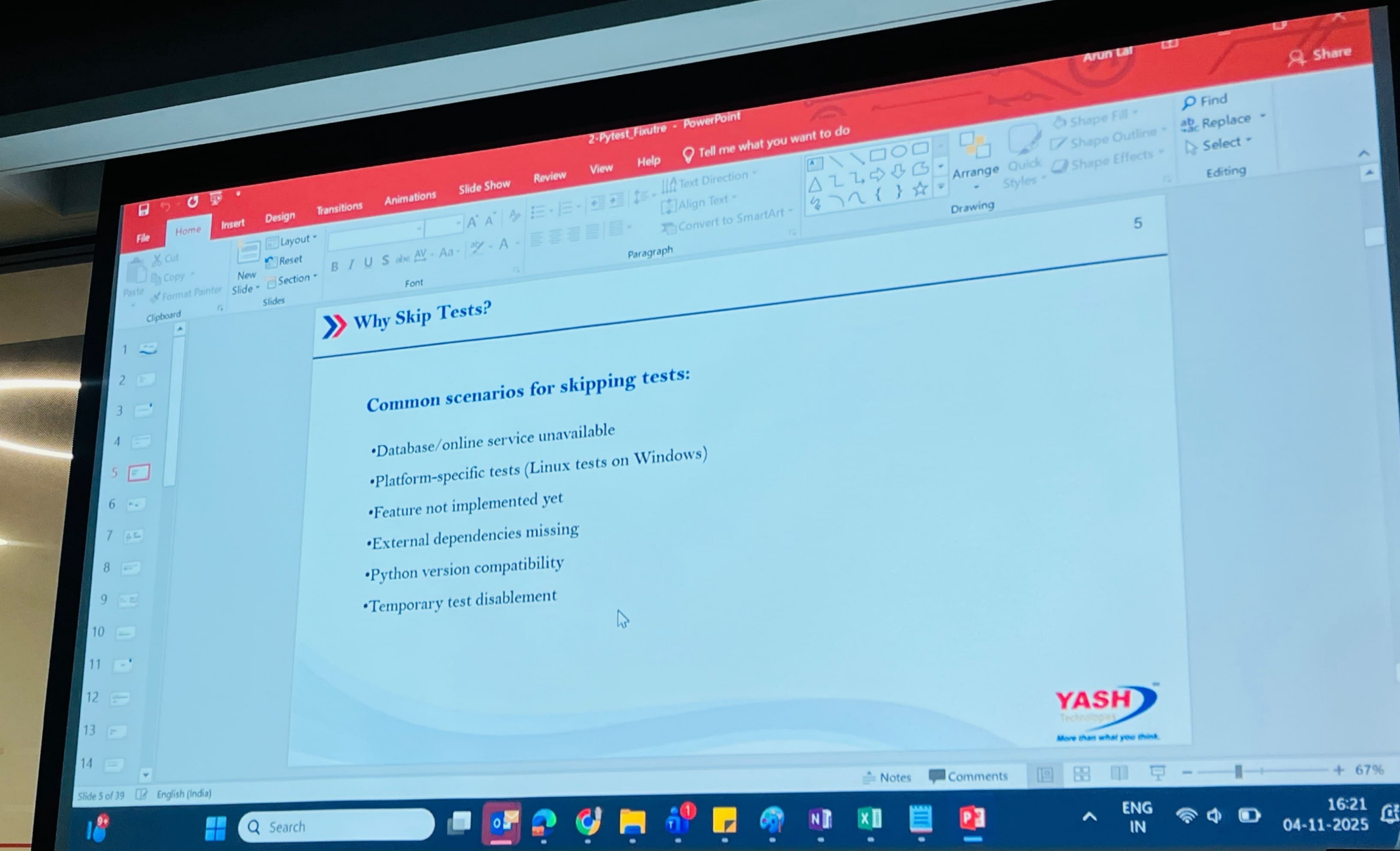
Skip: This marker is used to skip the particular test method by applying the pytest decorator

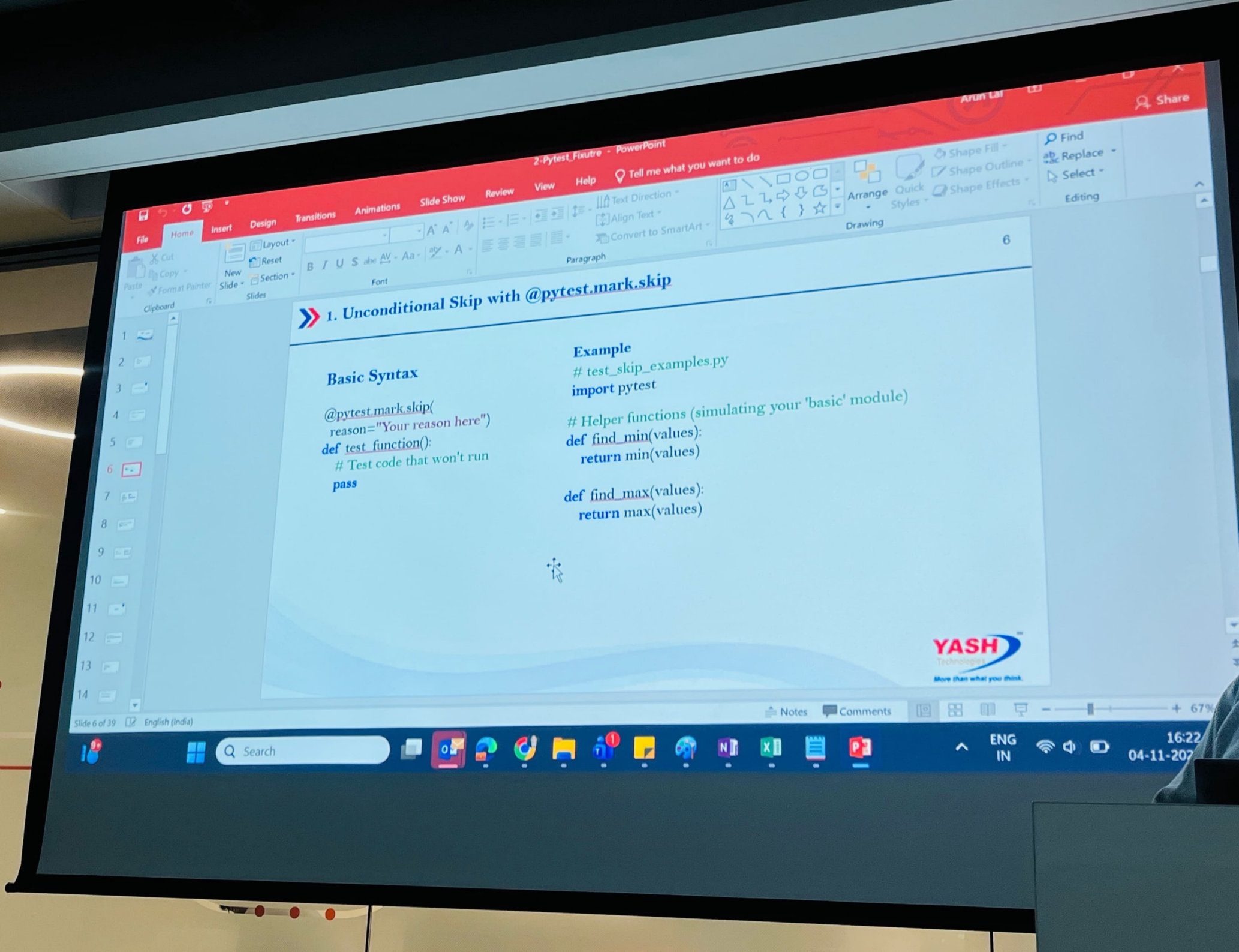
Pytest provides two main decorators to skip tests:

@pytest.mark.skip - Unconditionally skip a test

@pytest.mark.skipif - Conditionally skip a test based on a condition

YASH Technologies
More than what you think.


2-Pytest_Fixutre - PowerPoint
Tell me what you want to do
Share
Why Skip Tests?
5
Common scenarios for skipping tests:
•Database/online service unavailable
•Platform-specific tests (Linux tests on Windows)
•Feature not implemented yet
•External dependencies missing
•Python version compatibility
•Temporary test disablement
YASH
Slide 5 of 39
English (India)
Notes
Comments
67%
Search
ENG
IN
16:21
04-11-2025


2-Pytest_Fixutre - PowerPoint
Tell me what you want to do
Share
6
1. Unconditional Skip with @pytest.mark.skip
Basic Syntax
@pytest.mark.skip(
reason="Your reason here")
def test_function():
# Test code that won't run
pass
Example
# test_skip_examples.py
import pytest
# Helper functions (simulating your 'basic' module)
def find_min(values):
return min(values)
def find_max(values):
return max(values)
YASH
Technologies
Notes
Comments
Slide 6 of 39
English (India)
Search
ENG
IN
16:22
04-11-202

1. Unconditional Skip with @pytest.mark.skip

```
def test_max_function():
    values = (2, 3, 1, 4, 6)
    val = find_max(values)
    assert val == 6

# Skip without reason
@pytest.mark.skip
def test_always_skipped():
    assert False  # This would fail if it ran

# Unconditional skip - test will NEVER run
@pytest.mark.skip(reason="Temporarily
        disabled for maintenance")
def test_min_function():
    values = (2, 3, 1, 4, 6)
    val = find_min(values)
    assert val == 1
```

Running the Tests

```
> pytest test_skip_examples.py -v
```

Output:

- test_skip_examples.py::test_max_function PASSED
- test_skip_examples.py::test_always_skipped SKIPPED
- test_skip_examples.py::test_min_function SKIPPED (Temporarily disabled for maintenance)

YASH Technologies
More than what you think.

2. Conditional Skip with @pytest.mark.skipif

If you wish to skip something conditionally then you can use skipif instead. Here is an example of marking a test function to be skipped when run on an interpreter earlier than Python3.10:

Basic Syntax

```
@pytest.mark.skipif(condition, reason="Your reason here")
def test_function():
    # Test code that runs only if condition is False
    pass
```

YASH Technologies
More than what you think.

2. Conditional Skip with @pytest.mark.skipif

Example : Skip a test if Python version < 3.8

```
import pytest
@pytest.mark.skip(reason="This test is temporarily disabled")
def test_example_skip():
    assert True

@pytest.mark.skipif(sys.version_info < (3, 8),
                    reason="Requires Python 3.8 or higher")
def test_example_skipif():
    assert 2 + 2 == 4
```

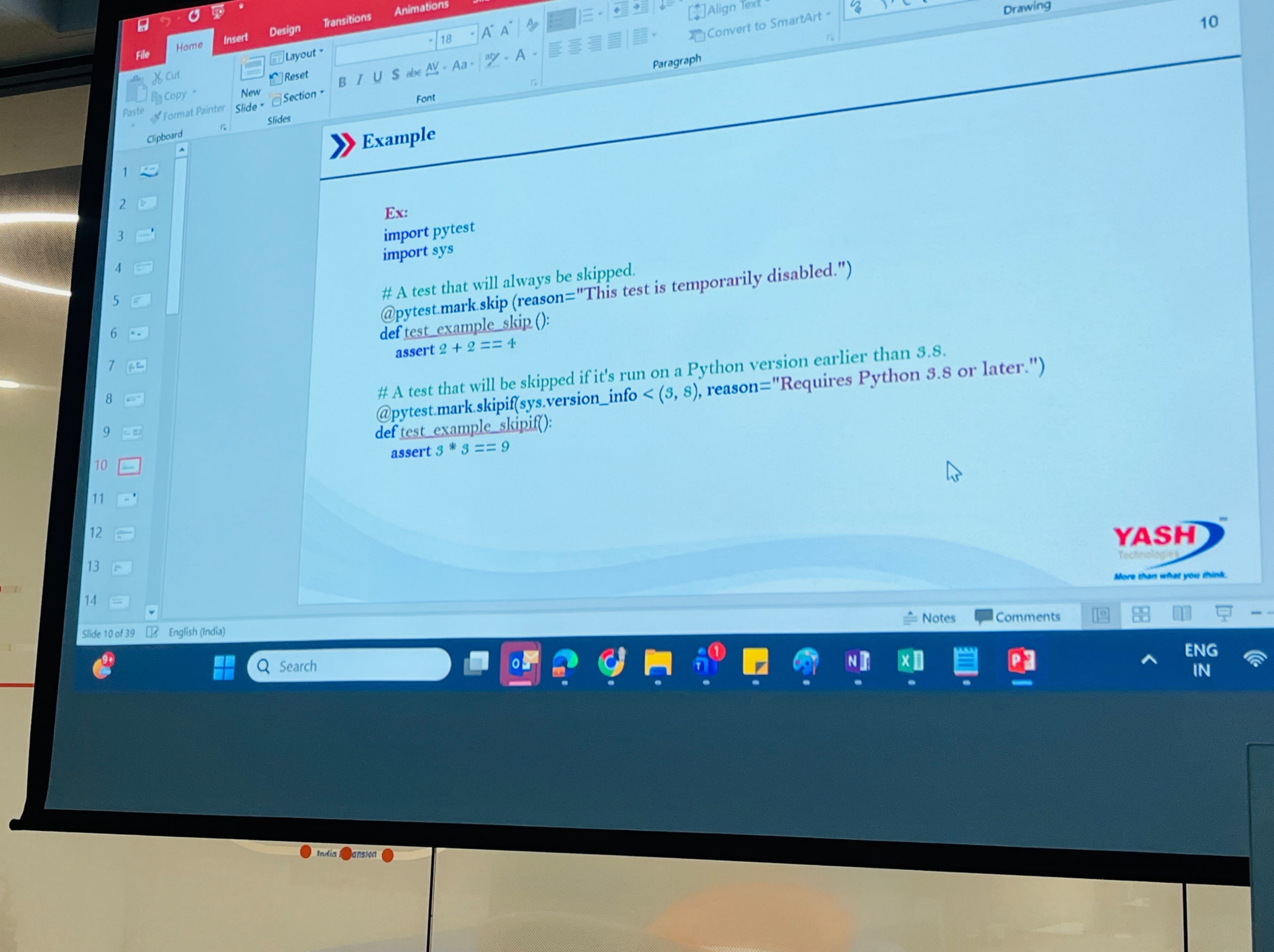
Run: pytest test_skip_condition.py

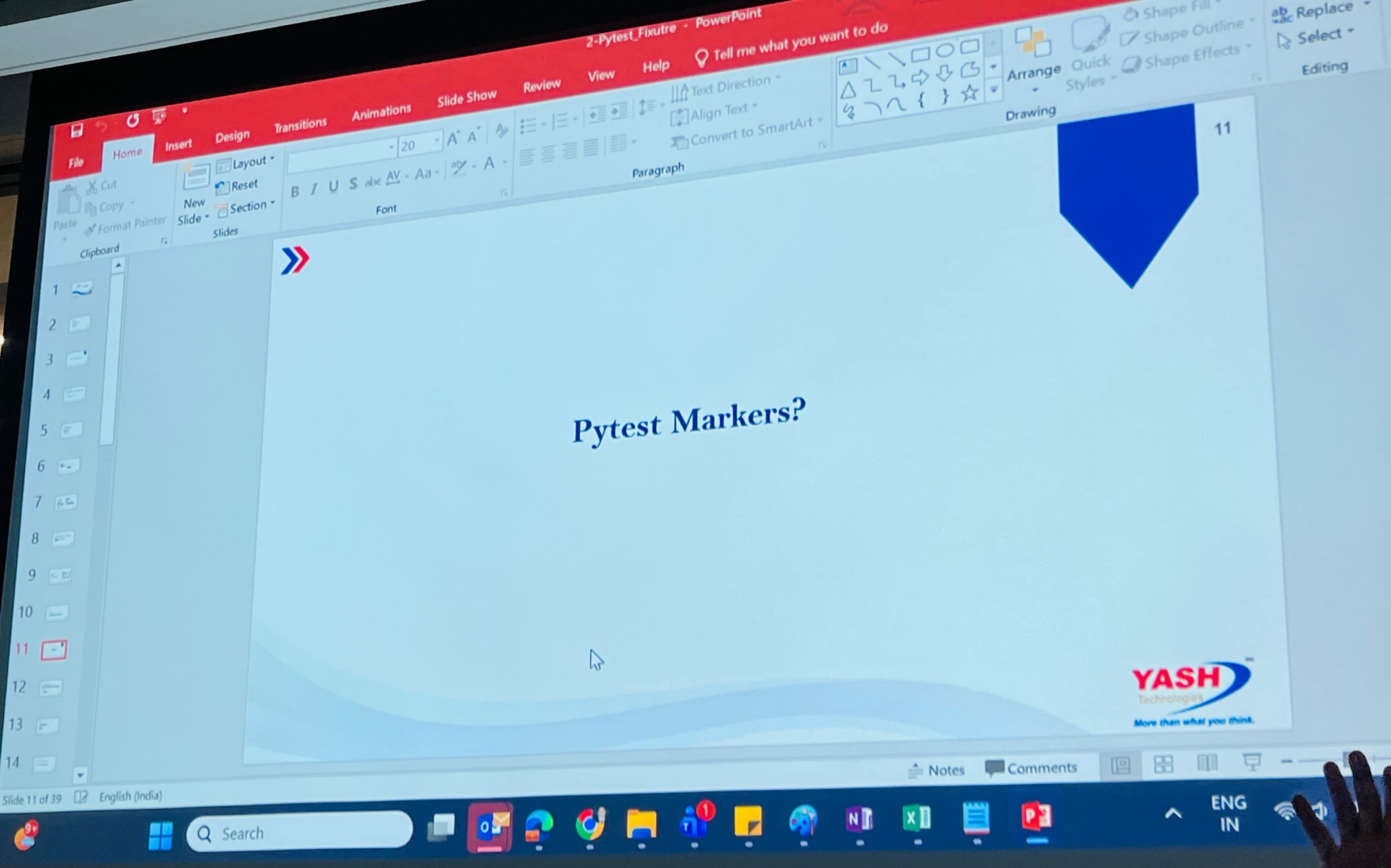
Output (on Python 3.7):
test_skip_condition.py [100%]

========= 2 skipped =====
Output (on Python 3.10+):
test_skip_condition.py s. [100%]

=====1 skipped, 1 passed ======
The second test only runs if Python ≥ 3.8.

YASH Technologies
Move than what you think.


File
Home
Insert
Design
Transitions
Animations
Cut
Copy
Format Painter
Paste
Clipboard
New Slide
Layout
Reset
Section
Slides
Font
Paragraph
Align Text
Convert to SmartArt
Drawing
10
Example
Ex:
import pytest
import sys
# A test that will always be skipped.
@pytest.mark.skip (reason="This test is temporarily disabled.")
def test_example_skip ():
assert 2 + 2 == 4
A test that will be skipped if it's run on a Python version earlier than 3.8.
@pytest.mark.skipif(sys.version_info < (3, 8), reason="Requires Python 3.8 or later.")
def test_example_skipif():
assert 3 * 3 == 9
YASH
Technologies
More than what you think.
Notes
Comments
Slide 10 of 39
English (India)
Search
ENG
IN


2-Pytest_Fixutre - PowerPoint
Tell me what you want to do
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Cut
Copy
Format Painter
Paste
Clipboard
Layout
Reset
Section
New Slide
Slides
Font
Paragraph
Text Direction
Align Text
Convert to SmartArt
Arrange
Quick Styles
Shape Fill
Shape Outline
Shape Effects
Drawing
Replace
Select
Editing
11
Pytest Markers?
YASH Technologies
More than what you think.
Notes
Comments
Slide 11 of 39
English (India)
Search
ENG
IN

Pytest Markers?

We can use markers to organize tests into units.
Pytest markers are a powerful feature that allows you to add metadata or labels to your test functions, making it easier to organize and customize your test suite.

Markers help you categorize and select specific tests to run, especially when dealing with large test suites.

Pytest Markers are labels you can add to test functions to:

- Organize tests into categories/groups.
- Run specific subsets of tests.
- Add metadata to tests.
- Control test behaviour.

YASH Technologies
More than what you think.

Why Use Markers?

When you have **many test cases**, you may not want to run them all at once.

For example:

- Run only fast tests
- Run only database-related tests
- Run tests related to a specific module

Markers let you easily select which group of tests to run.

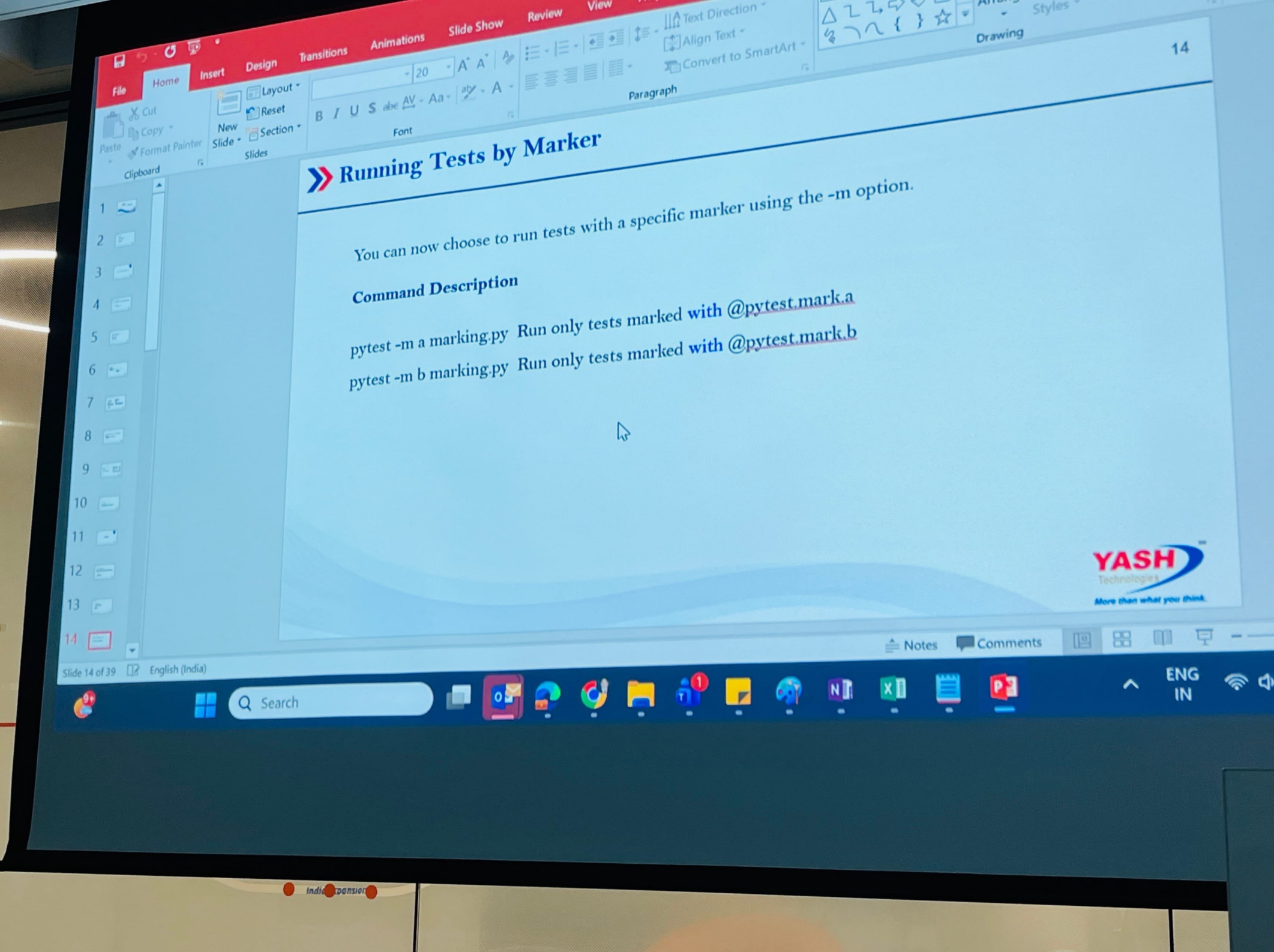
Basic Marker Syntax

```
import pytest
@pytest.mark.marker_name
def test_function():
    assert True
```

YASH Technologies
More than what you think.

13

Slide 13 of 39 English (India)


File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Cut
Copy
Format Painter
Paste
Clipboard
Layout
Reset
Section
New Slide
Slides
Font
Paragraph
Text Direction
Align Text
Convert to SmartArt
Drawing
Styles
14
Running Tests by Marker
You can now choose to run tests with a specific marker using the -m option.
Command Description
pytest -m a marking.py Run only tests marked with @pytest.mark.a
pytest -m b marking.py Run only tests marked with @pytest.mark.b
YASH
Technologies
More than what you think.
Notes
Comments
Slide 14 of 39
English (India)
Search
ENG
IN

Example

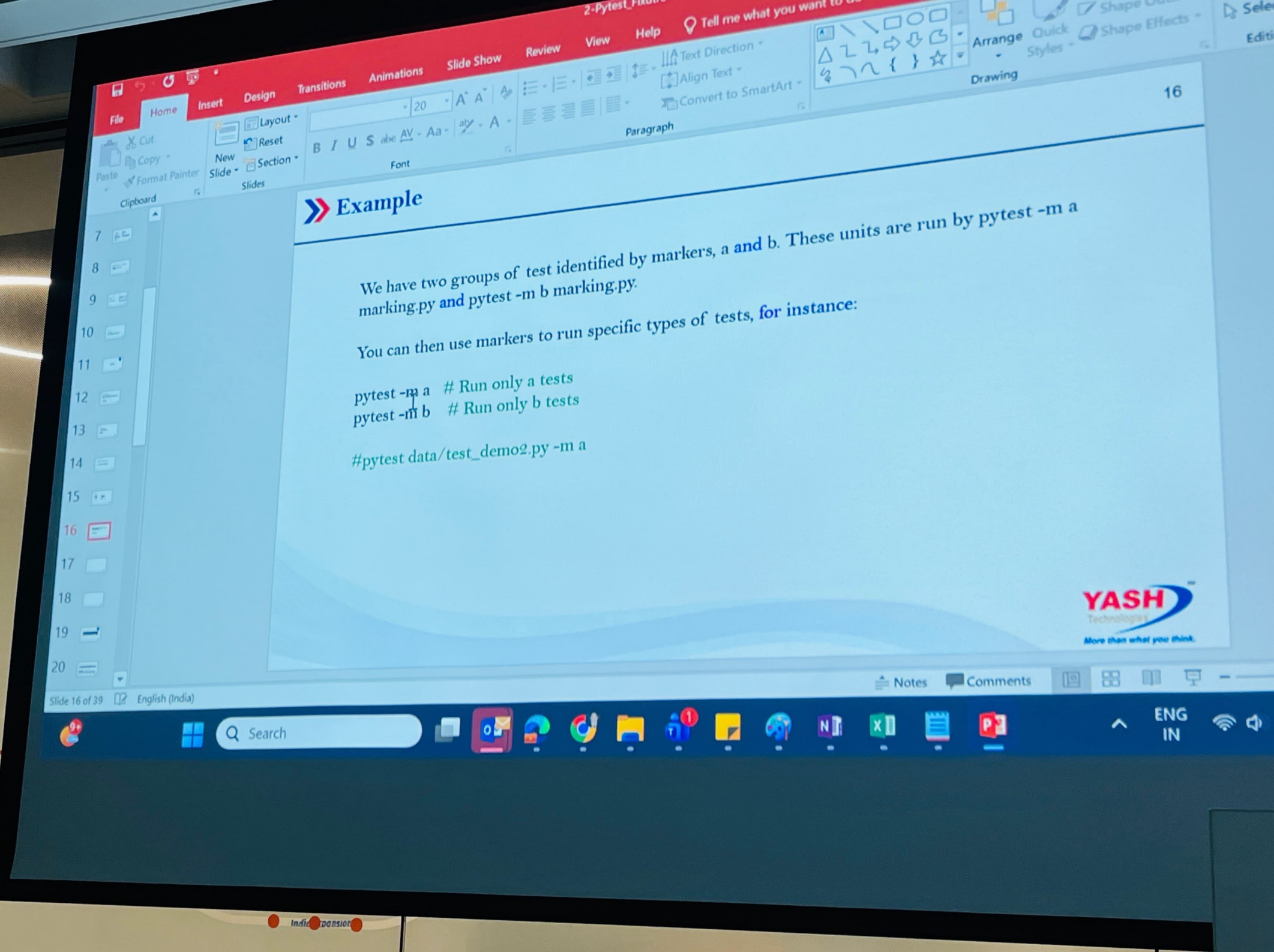
marking.py

```
# pytest -m a marking.py
# pytest -m b marking.py
import pytest
@pytest.mark.a
def test_a1():
    assert (1) == (1)

@pytest.mark.a
def test_a2():
    assert (1, 2) == (1, 2)

@pytest.mark.a
def test_a3():
    assert (1, 2, 3) == (1, 2, 3)
@pytest.mark.b
def test_b1():
    assert "falcon" == "fal" + "con"
@pytest.mark.b
def test_b2():
    assert "falcon" == f'fal{'con'}"
```

YASH Technologies
More than what you think.


2-Pytest_Fixtures
Tell me what you want to do
File
Home
Insert
Design
Transitions
Animations
Slide Show
Review
View
Help
Cut
Copy
Format Painter
Paste
Clipboard
New Slide
Layout
Reset
Section
Slides
Font
Paragraph
Text Direction
Align Text
Convert to SmartArt
Arrange
Quick Styles
Shape Effects
Drawing
16
Example
We have two groups of test identified by markers, a and b. These units are run by pytest -m a marking.py and pytest -m b marking.py.
You can then use markers to run specific types of tests, for instance:
pytest -m a # Run only a tests
pytest -m b # Run only b tests
#pytest data/test_demo2.py -m a
YASH Technologies
More than what you think.
Notes
Comments
Slide 16 of 39
English (India)
Search
ENG
IN