

```
# test_presentation_assistant.py
```

```
import pytest
```

```
from presentation_assistant import (
```

```
    PresentationAssistant,
```

```
    SlideNotFoundError,
```

```
)
```

```
@pytest.fixture
```

```
def assistant():
```

```
    slides = [
```

```
        "Welcome to our presentation!",
```

```
        "Here is the agenda for today.",
```

```
        "Let's dive into the first topic.",
```

```
        "Thank you for attending.",
```

```
    ]
```

```
    return PresentationAssistant(slides)
```

```
def test_init():
```

```
    slides = ["Slide 1", "Slide 2"]
```

```
    pa = PresentationAssistant(slides)
```

```
    assert pa.slides == slides
```

```
    assert pa.current_slide == 0
```

```
def test_next_slide(assistant):  
    assistant.next_slide()  
  
    assert assistant.current_slide == 1  
  
    assistant.next_slide()  
  
    assert assistant.current_slide == 2
```

```
def test_previous_slide(assistant):  
    assistant.current_slide = 2  
  
    assistant.previous_slide()  
  
    assert assistant.current_slide == 1  
  
    assistant.previous_slide()  
  
    assert assistant.current_slide == 0
```

```
def test_next_slide_at_end(assistant):  
    assistant.current_slide = len(assistant.slides) - 1  
  
    with pytest.raises(SlideNotFoundError):  
        assistant.next_slide()
```

```
def test_previous_slide_at_start(assistant):  
    with pytest.raises(SlideNotFoundError):  
        assistant.previous_slide()
```

```
def test_go_to_slide(assistant):
```

```
    assistant.go_to_slide(2)
```

```
    assert assistant.current_slide == 2
```

```
def test_go_to_slide_out_of_range(assistant):
```

```
    with pytest.raises(SlideNotFoundError):
```

```
        assistant.go_to_slide(len(assistant.slides))
```

```
def test_go_to_slide_negative(assistant):
```

```
    with pytest.raises(SlideNotFoundError):
```

```
        assistant.go_to_slide(-1)
```

```
def test_current_slide_content(assistant):
```

```
    content = assistant.current_slide_content()
```

```
    assert content == assistant.slides[0]
```

```
    assistant.next_slide()
```

```
    content = assistant.current_slide_content()
```

```
    assert content == assistant.slides[1]
```

```
def test_show_slide(
```

```
    assistant, capsys
```

): # capsys is a pytest fixture to capture stdout and stderr

```
assistant.show_slide()
```

```
captured = capsys.readouterr()
```

```
assert captured.out.strip() == assistant.slides[0]
```

```
assistant.next_slide()
```

```
assistant.show_slide()
```

```
captured = capsys.readouterr()
```

```
assert captured.out.strip() == assistant.slides[1]
```