

```
[1]: # Let's rewrite the given code in a different structure, while maintaining the original intent
```

```
# First, we will import the necessary library which is numpy in this case
```

```
import numpy as np
```

```
# Define the function to calculate the t-value for a hypothesis test
```

```
def calculate_t_statistic(sample_mean, population_mean, sample_std_dev, sample_size):
```

```
    # Calculate the t-statistic (t-value) using the formula
```

```
    t_statistic = (sample_mean - population_mean) / (sample_std_dev / np.sqrt(sample_size))
```

```
    return t_statistic
```

```
# Define the function to find the critical t-value based on the significance level and degrees of freedom
```

```
def get_critical_t_value(significance_level, degrees_freedom):
```

```
    # Import the t distribution from scipy.stats to use its ppf method
```

```
    from scipy.stats import t
```

```
    # Calculate the critical t-value
```

```
    critical_value = t.ppf(1 - significance_level / 2, degrees_freedom)
```

```
    return critical_value
```

```
# Define the function to perform the hypothesis test
```

```
def hypothesis_test(sample_data, hypothesized_mean):
```

```
    # Calculate sample mean and standard deviation using numpy functions
```

```
    sample_mean = np.mean(sample_data)
```

```
    sample_std_dev = np.std(sample_data, ddof=1)
```

```
    sample_size = len(sample_data)
```

```
    # Calculate the t-value
```

```
    t_value = calculate_t_statistic(sample_mean, hypothesized_mean, sample_std_dev, sample_size)
```

```
    # Degrees of freedom
```

```
    degrees_freedom = sample_size - 1
```

```
    # Determine the critical t-value for a 95% confidence interval
```

```
    critical_t_value = get_critical_t_value(0.05, degrees_freedom)
```

```
    # Check if the absolute t-value is greater than the critical t-value
```

```
    if abs(t_value) > critical_t_value:
```

```
        print("The null hypothesis is rejected.")
```

```
        return False
```

```
    else:
```

```
        print("The null hypothesis is not rejected.")
```

```
        return True
```

```
# Sample data provided
```

```
sample_scores = [92.64, 79.00, 84.79, 97.41, 93.68, 65.23, 84.50, 73.49, 73.97, 79.11]
```

```
national_average = 75
```

```
# Perform the hypothesis test
```

```
hypothesis_test_result = hypothesis_test(sample_scores, national_average)
```

```
hypothesis_test_result
```

```
The null hypothesis is rejected.
```

```
[1]: False
```