Main Title

Class subtitle

04/09/2024 - 20/11/2024

Tom Planche
2024-2025
Class name

I - Main title

I. I - Subtitle

I. I .I - Subsubtitle

Custom Block

> Custom Blockquote

`Basic inline raw text`

This code block uses `#code()` macro.

Rust

```rust
// src/string_utils.rs
/// Extension traits and utilities for string manipulation
///
/// This module provides additional functionality for working with strings,
/// including title case conversion and other string transformations.
use std::string::String;

/// Trait that adds title case functionality to String and &str types
pub trait TitleCase {
    /// Converts the string to title case where each word starts with an uppercase letter
    /// and the rest are lowercase
    ///
    fn to_title_case(&self) -> String;
}

impl TitleCase for str {
    fn to_title_case(&self) -> String {
        self.split(|c: char| c.is_whitespace() || c == '_' || c == '-')
            .filter(|s| !s.is_empty())
            .map(|word| {
                // If the word is all uppercase and longer than 1 character, preserve it
                if word.chars().all(|c| c.is_uppercase()) && word.len() > 1 {
                    word.to_string()
                } else {
                    let mut chars = word.chars();
                    match chars.next() {
                        None => String::new(),
                        Some(first) => {
                            let first_upper = first.to_uppercase().collect::<String>();
                            let rest_lower = chars.as_str().to_lowercase();
                            format!("{}{}", first_upper, rest_lower)
                        }
                    }
                }
            })
            .collect::<Vec<String>>()
            .join(" ")
    }
}

```

```rust
impl TitleCase for String {
    fn to_title_case(&self) -> String {
        self.as_str().to_title_case()
    }
}

#[cfg(test)]
mod tests {
    use super::*;

    #[test]
    fn test_title_case_str() {
        assert_eq!("hello world".to_title_case(), "Hello World");
        assert_eq!("HASH_TABLE".to_title_case(), "HASH TABLE");
        assert_eq!("dynamic-programming".to_title_case(), "Dynamic Programming");
        assert_eq!("BFS".to_title_case(), "BFS");
        assert_eq!("two-sum".to_title_case(), "Two Sum");
        assert_eq!("binary_search_tree".to_title_case(), "Binary Search Tree");
        assert_eq!("   spaced   words   ".to_title_case(), "Spaced Words");
        assert_eq!("".to_title_case(), "");
    }
}
```