le cnam

Main Title

Class subtitle

07/09/2025 - 08/02/2026

Tom Planche

2026-2027

Class name

Table des matières

Chapitre 1

Main title

I - Maths

For my maths class, I made these things:

I.I - #definition

Definition 1.1. (Linearity):
We say that φ is linear (homomorphism) if:

$$\varphi(\lambda_1 X_1 + \lambda_2 X_2 + ... + \lambda_n X_n) = \lambda_1 \varphi(X_1) + \lambda_2 \varphi(X_2) + ... + \lambda_n \varphi(X_n) \qquad (1.1.1.1)$$

I.II - #example

Example 1.1. (Example title): Basic text.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

$$\varphi(0,0,0) = (0,0) = 0_{\mathbb{R}^2}$$
$$\varphi(\alpha X_1 + \beta X_2) \overset{?}{=} \alpha\varphi(X_1) + \beta\varphi(X_2) \qquad (1.1.2.2)$$

I.III - #theorem

I.III.1 - With title

Theorem 1.1. (Stokes' Theorem):
Let M be an oriented differential manifold with boundary of dimension n, and ω a $(n-1)$-form differential form with compact support on M of class C_1.
Then, we have:

$$\int_M d\omega = \int_{\{\partial M\}} i^* \omega \qquad (1.1.3.3)$$

where d denotes the exterior derivative, ∂M the boundary of M, equipped with the induced orientation,
and $i^*\omega = \omega \mid_{\{\partial M\}}$ the restriction of ω to ∂M.

I.III.2 - Without `title`

Theorem 1.2.
Let E be a finite-dimensional vector space, F a vector subspace of E, and $B = (X_1, X_2, ..., X_n)$ a basis of F.
Then, there exists a basis $\left(X_1, X_2, ..., X_n, X_{\{n+1\}}, ..., X_m\right)$ of E such that $(X_1, X_2, ..., X_n)$ is a basis of F.

I.IV - Custom styling

Definition 1.2. (**Styled Definition**): A **group** is a set G equipped with a binary operation $\cdot$ satisfying closure, associativity, identity, and invertibility.

Theorem 1.3. (Styled Theorem):
For any right triangle with sides a, b, and hypotenuse c:

$$a^2 + b^2 = c^2 \tag{1.1.4.4}$$

I.V - `ar`

For vectors, I use `ar(X)` and it gives $\vec{X}$.

II - Subtitle

II.I - Subsubtitle

Custom Block

Styled Block

This block uses custom title and body styling.

> Custom Blockquote

`Basic inline raw text`

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

```
src/string_utils.rs
/// Extension traits and utilities for string manipulation
```

```rust
///
/// 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(" ")
    }
}

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(), "");
    }
}
```