Video Number:	01_01
Video Title:	Introduction to the standard Rust library
Estimated Length:	3 min (400-500 words = 2.5-3.5 minutes) = 150 words

Video learning goal: At the end of this video, the student will learn how to do …(your learning objective).

What will you cover? Add 3 to 5 brief bullet points of what you will cover in this video. This will eventually become the video description, which has the potential to boost search and viewership by 40% because of effective SEO. This also helps you create an outline before scripting.

- Introduction to the standard Rust library
- Point 2
- Point 3
- Point 4

Use the **table below** to write your dialogue and show what will be on screen while you are talking.

Begin with a good **hook** to get the member's attention and engage them from the start. Perhaps tell a brief story or anecdote. This will vary by subject matter and course type. Do your best if the content is tech heavy.

Next let them know the **learning goal** for the movie in a few sentences. Explain what you'll show them how to do. Or, what they'll be able to do after watching.

Share your content. Refer back to the bullet points you wrote above. Try to break up your script so each paragraph is no longer than 1 **to 3 sentences**. If there is a shift to what is seen or you move on to a new topic, then you can start a new paragraph.

If there is a main point that we want the audience to remember, then bold it.

Make sure you have a conclusion. Consider reminding the member of your key points during the movie. Or give them a call to action. Do not tell them what is coming up. Movies are meant to be standalones as well as part of the overall course.

	Script text or talking points	Visuals / Actions on Screen
A	Welcome to 'Rust Web Frameworks: Building with std Library.' In this first module, we'll explore the power of Rust's Standard Library.	Show welcome screen, With rust logo

B	What is the Standard Library? The Standard Library is Rust's toolbox of essentials. It offers robust tools for data, I/O, and more. This library is available in rust by default. Hence we don’t need to add it in the dependencies in cargo.toml or install it separately.	Show rust std library doc (https://doc.rust-lang.org/std)
C	In this chapter, we'll explore the web-related functionalities and tools available within the Standard Library. Our goal is to show you how to use these features effectively for building your web applications.	Show rust std library doc (https://doc.rust-lang.org/std)
D	We'll start by introducing you to the core components of the Standard Library that are commonly used in web development. You'll get to know data types, collections, and error handling	
E	**Data Types and Collections**: Rust's `std` includes a rich set of data types, such as strings, vectors, arrays, and hash maps **Error Handling**: It offers the `Result` and `Option` types, which are crucial for handling errors gracefully in web applications **Concurrency**: It has modules for managing concurrency, such as `std::thread` and `std::sync`. **Cryptography**: Rust's `std::crypto` module provides features for working with cryptographic functions, which are essential for securing web applications **Random Number Generation**: The `std::rand` module allows generating random numbers	
F	By the end of this module, you'll be know exactly how to create web applications with the Rust Standard Library	
G		
H		
I		
J		

K		