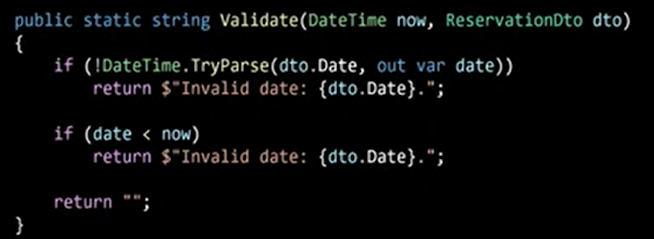
**What to LOG and when - Functional programming approach**

<https://www.youtube.com/watch?v=Ak1hGQuGBhY&ab_channel=NDCConferences>

Impure functions - functions that have side effects : writing to a DB, changing global state, sending emails etc etc

Pure functions - functions that given the same imput, returns the same output

This function validates whether the reservation is valid or not, meanining you can;t make a reservaiton in the past



Let's say I want a reservation for 2022-2-6.

Today is 2022-2-4 -> Validated returns that it's valid

Today is 2022-2-5 -> Validated returns that it's valid

Today is 2022-2-6 -> Validated returns that it's valid

Today is 2022-2-7 -> Validated returns that it's invalid

If the application would break in 2022-2-4 it would be hard for us to reproduce the bug in 2022-2-8 because given the same input we get a different output and that's an impure function

Function with logging embedded - looks bad



Code with logging hidden



We shouldn't log methods that are pure, because we can always reproduce the output with the same input

When dealing wiht a database, a user might get an error when trying to add an entity, at the time you try to reproduce the bug, some of the rows have been deleted, new added, others updated and therefore you can't really reproduce the exact state the user has been when he got the error, in this case we need to log the returned data so we can later troubleshoot it easier.

Minute 42:30 - What and How

What - > impure functions, functions that might have different outputs given the same input when trying to reproduce errors

How - >

Datetime -> wrap the datime in a interface

Repositories -> For every Repository Interface, create two implementations : one implementing the data access, another one for logging, using decator pattern



Inner is the data acess repository

Once a bug occurs , you logged all the data necessary to reproduce -> create a unit test and write code untill it passes