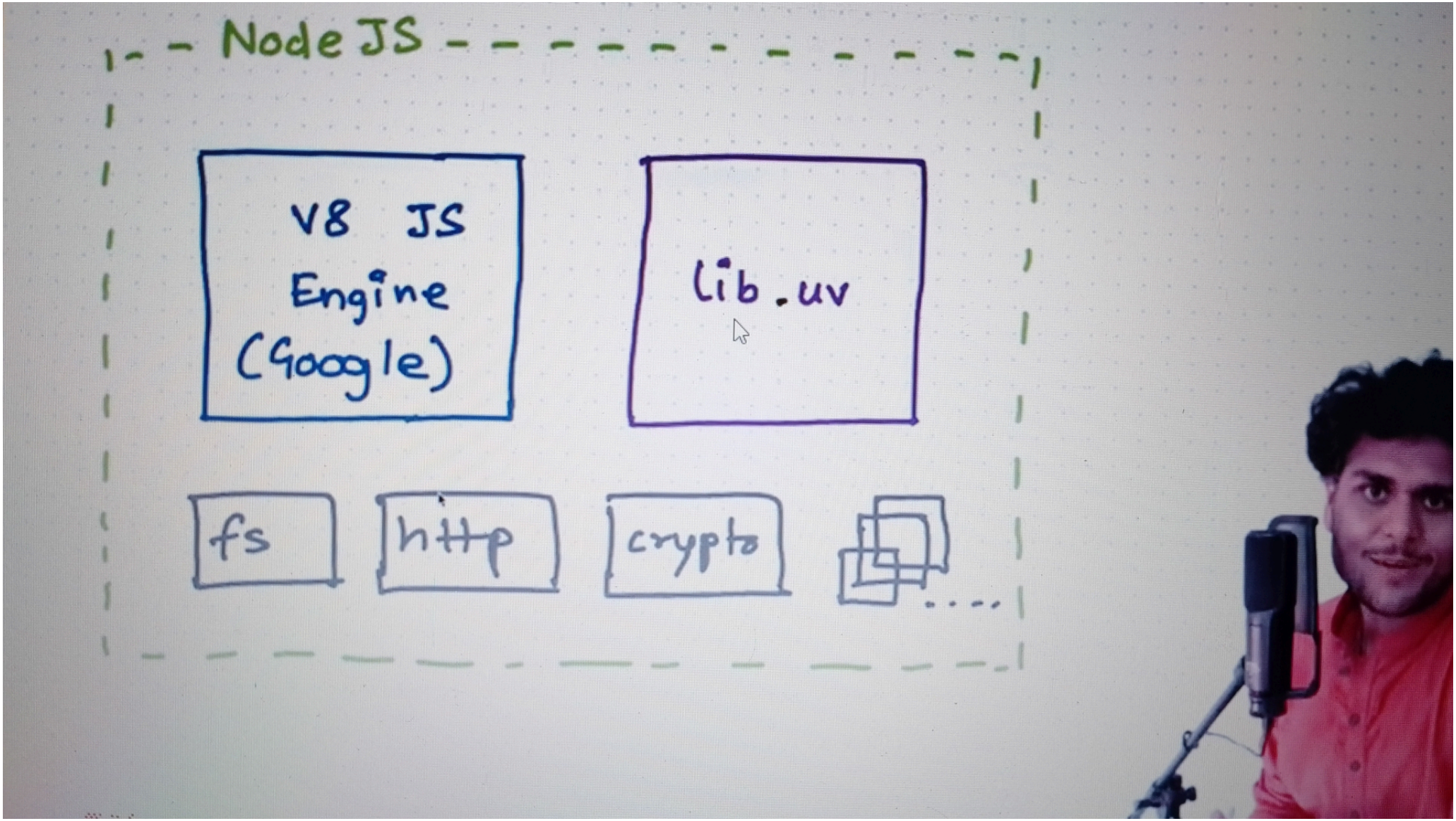
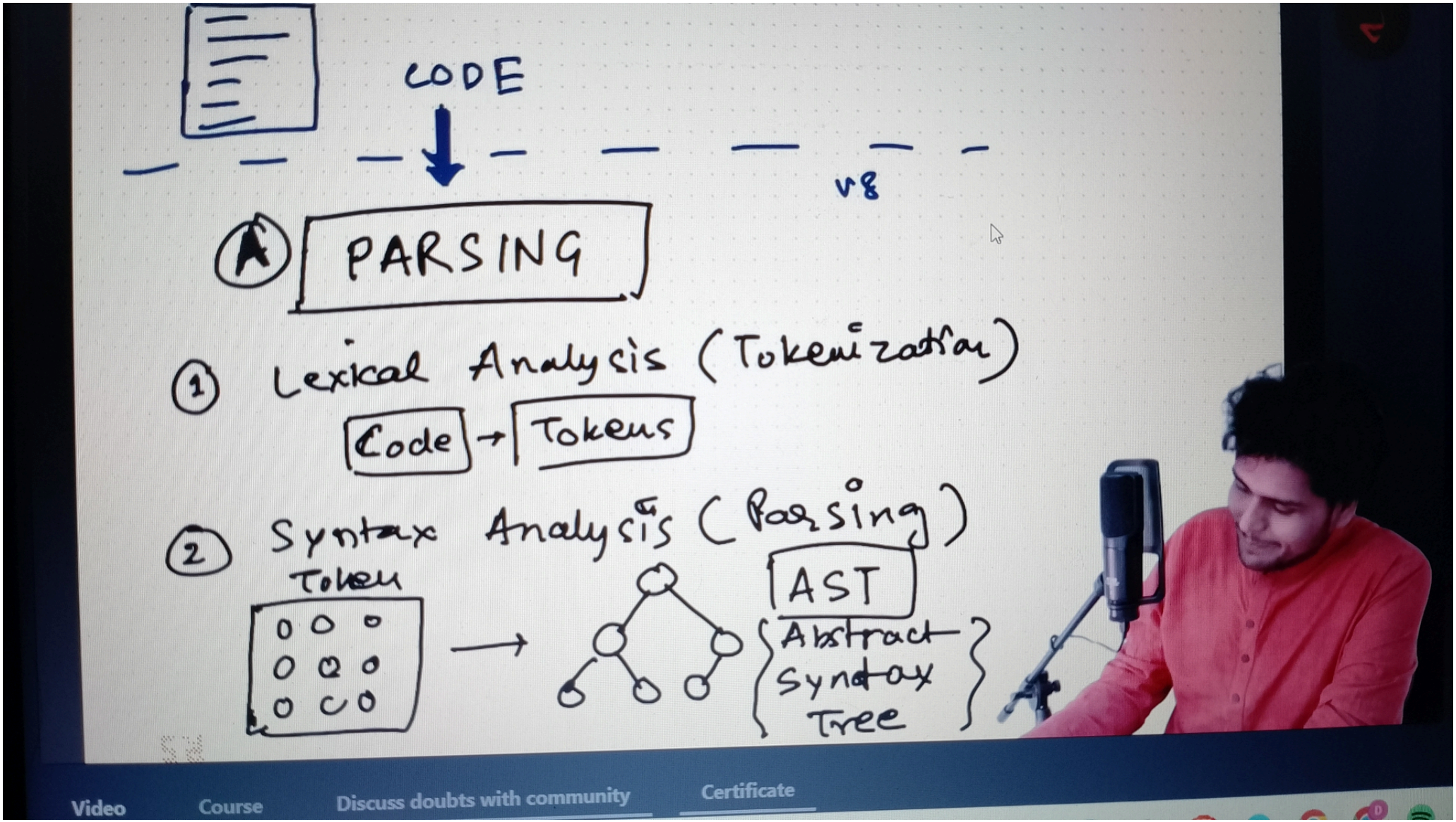
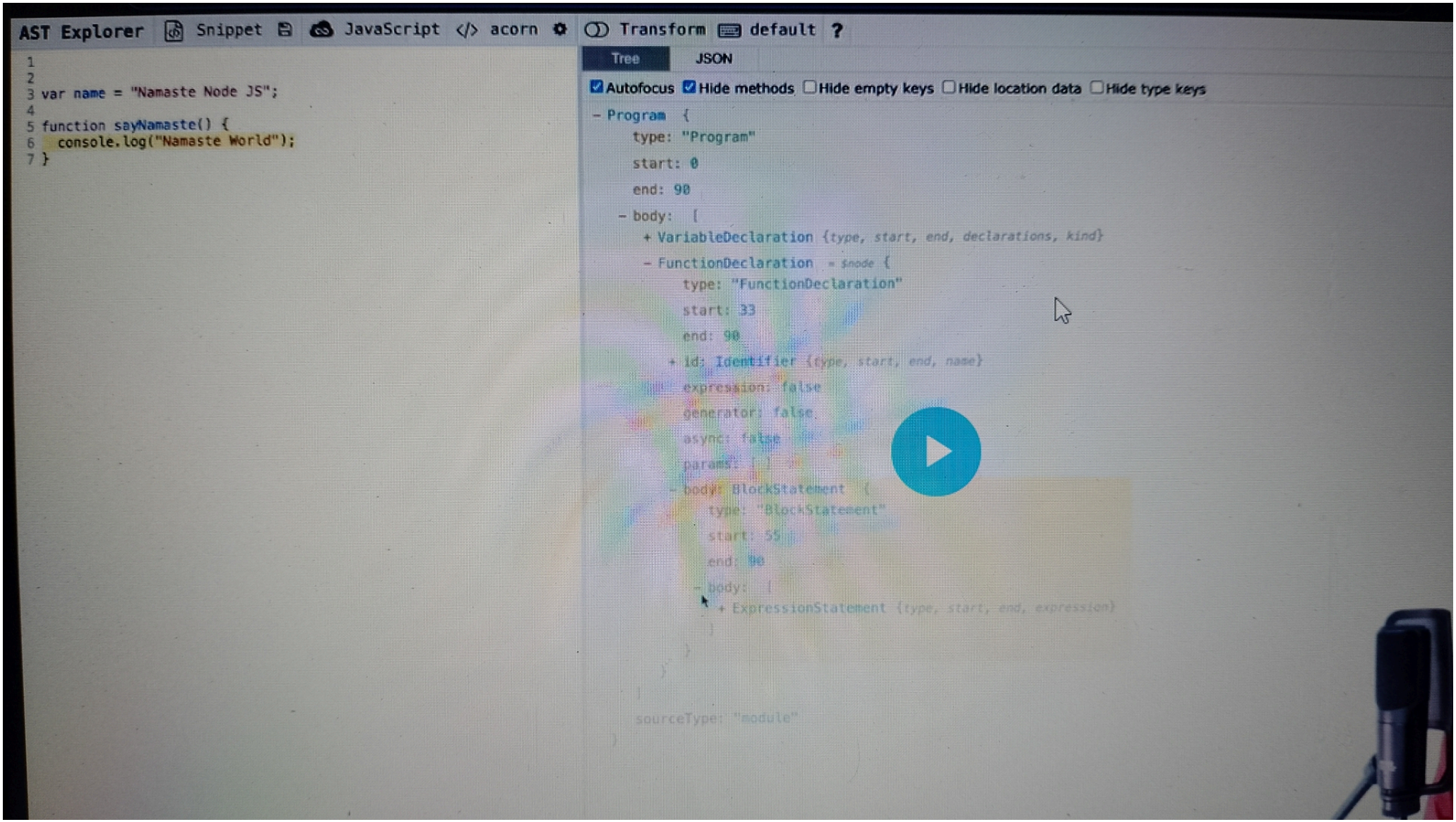
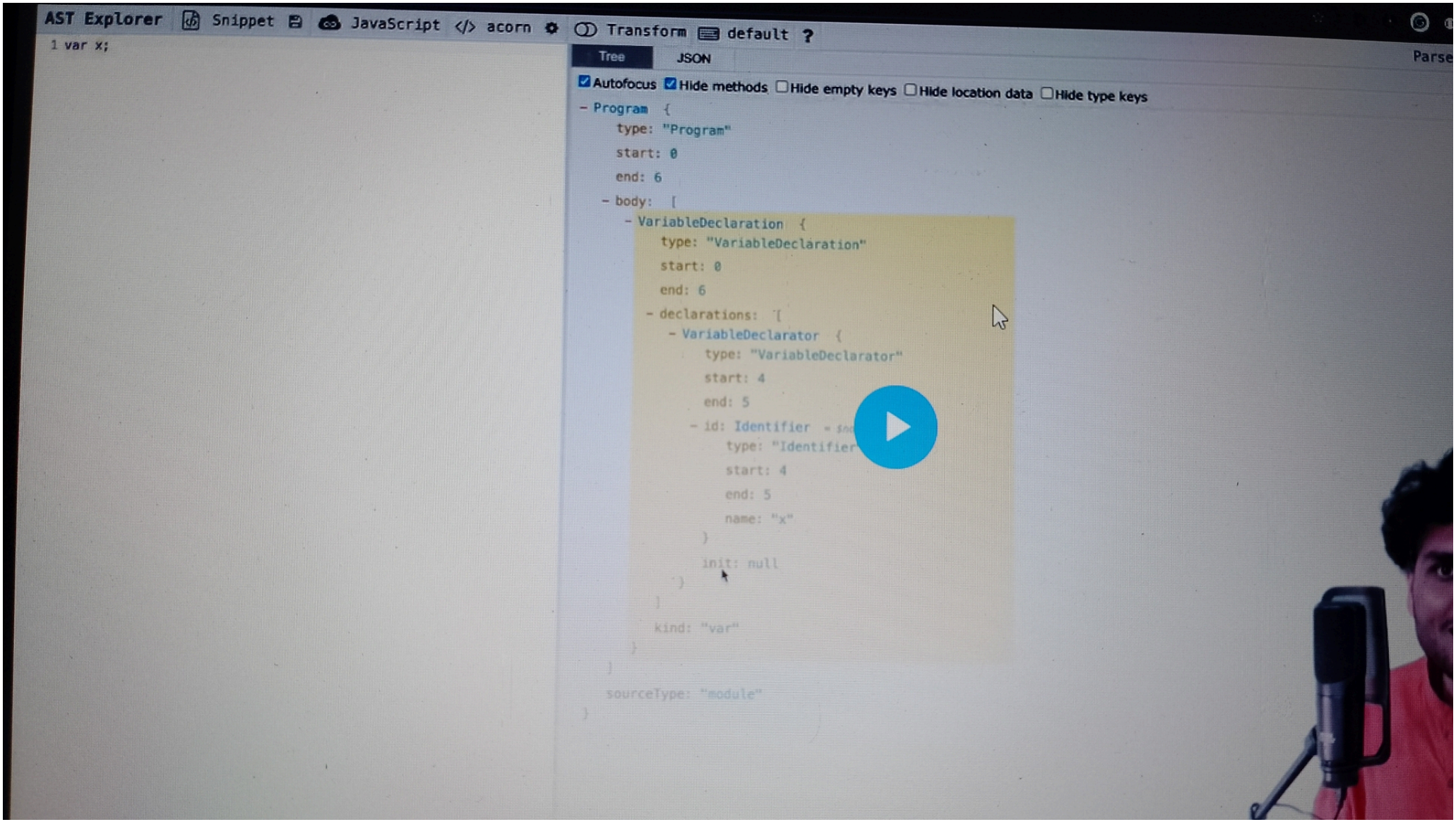
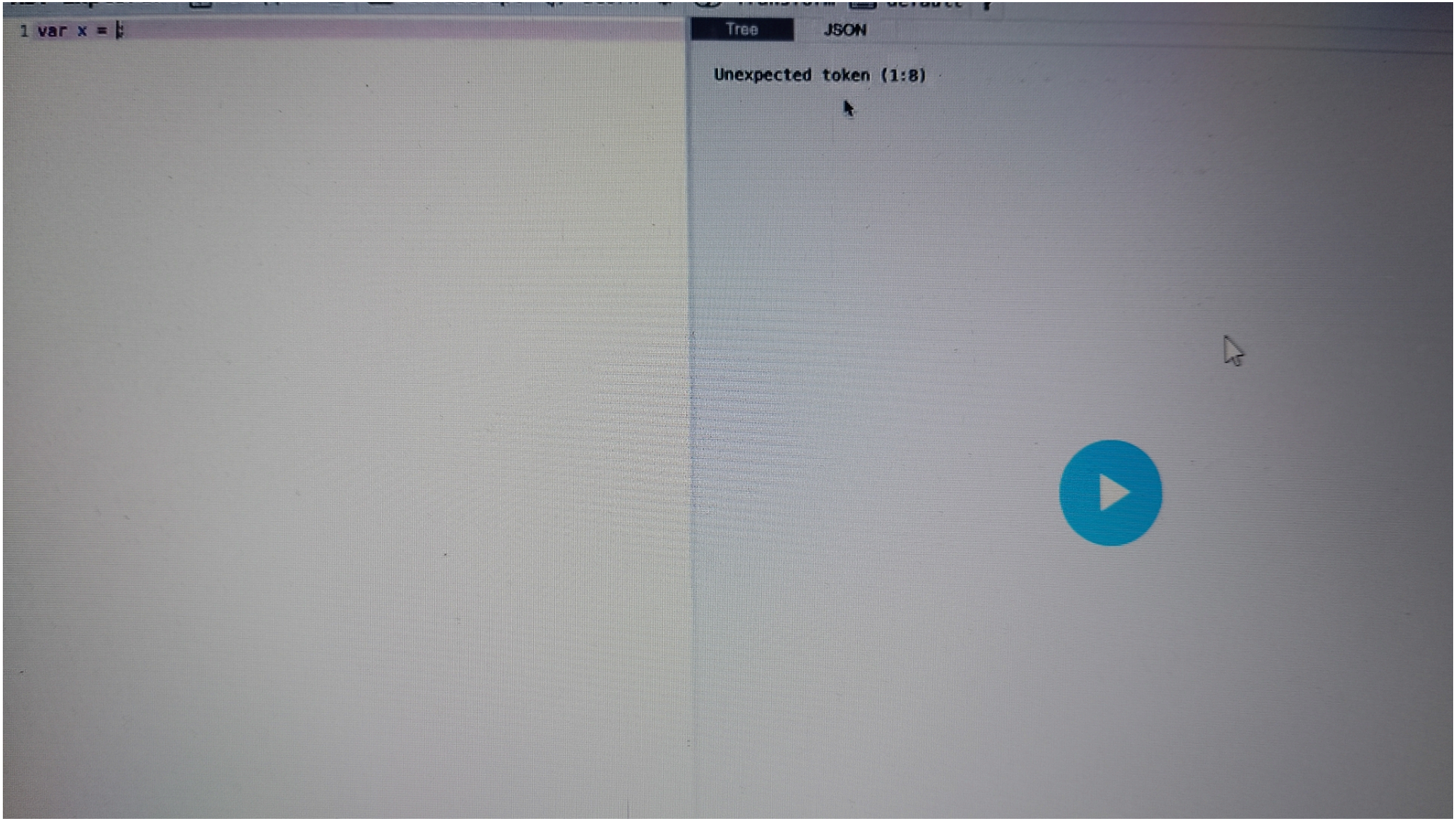
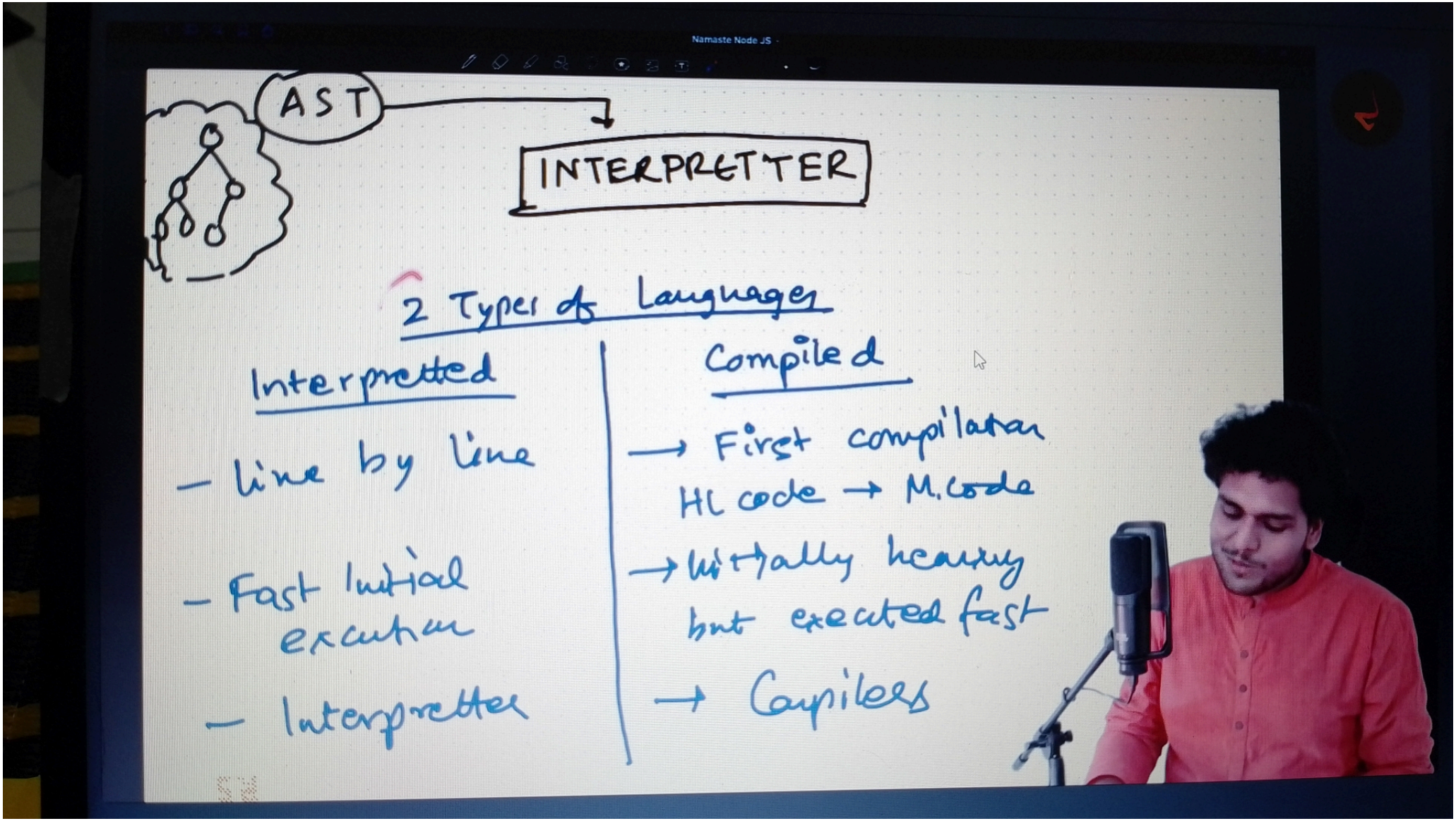

Node JS
V8 JS Engine (Google)
lib.uv
fs
http
crypto

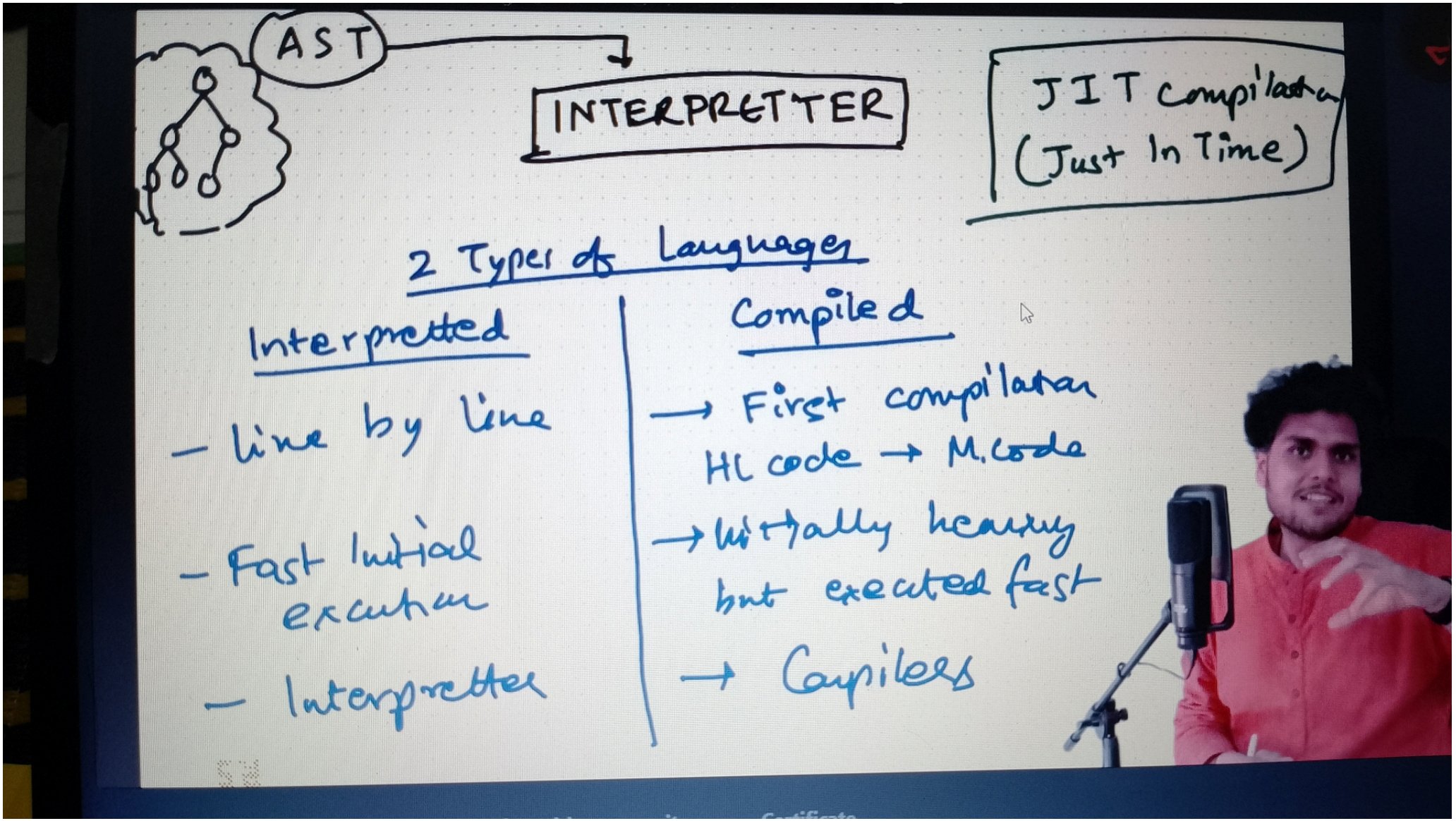

CODE
V8
A
PARSING
1 Lexical Analysis (Tokenization)
Code → Tokens
2 Syntax Analysis (Parsing)
Token
AST
Abstract
Syntax
Tree
Video
Course
Discuss doubts with community
Certificate

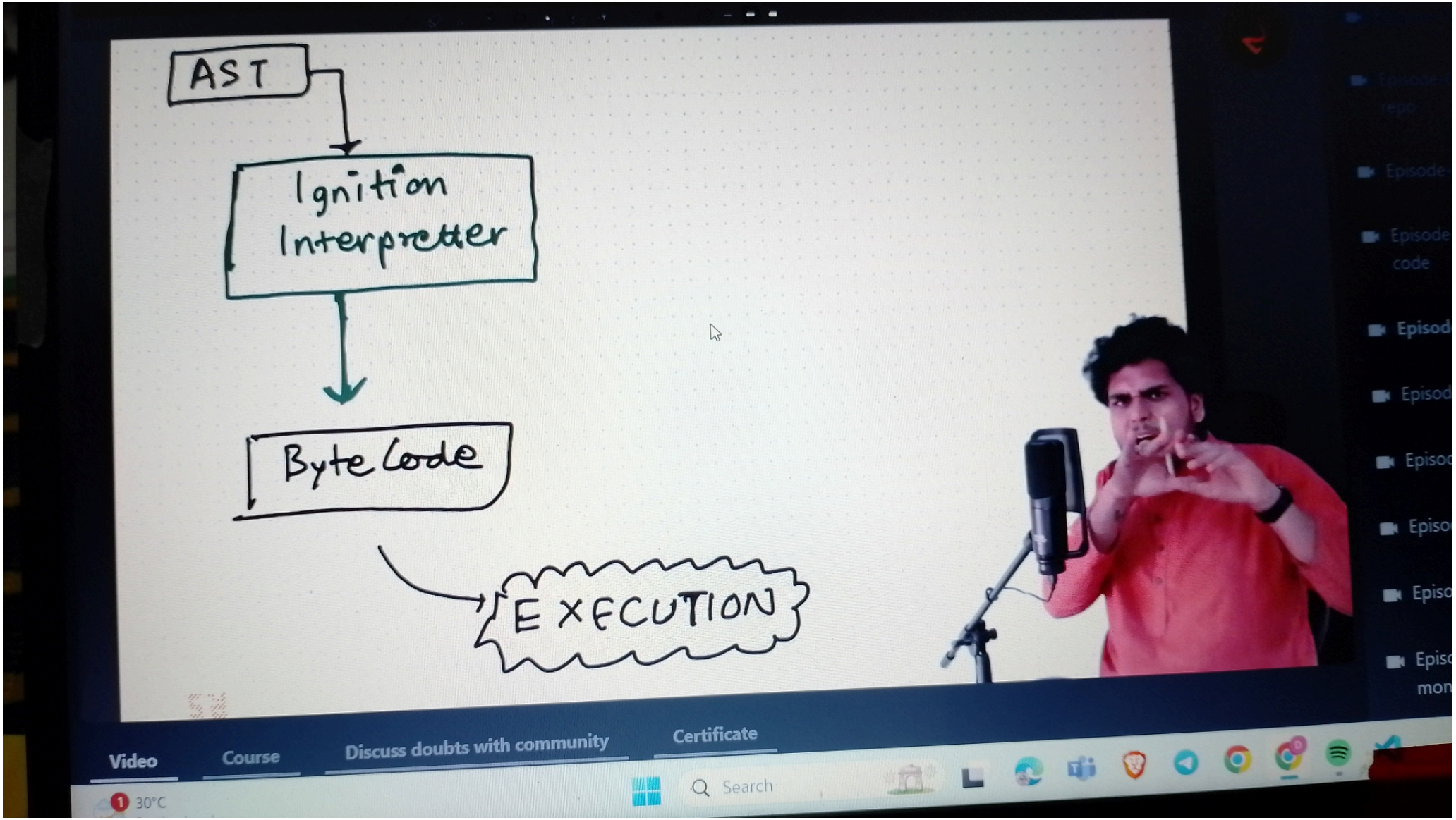

AST Explorer
Snippet
JavaScript
acorn
Transform
default
Tree
JSON
Autofocus
Hide methods
Hide empty keys
Hide location data
Hide type keys
var name = "Namaste Node JS";
function sayNamaste() {
console.log("Namaste World");
}
Program {
type: "Program"
start: 0
end: 90
body: [
VariableDeclaration {type, start, end, declarations, kind}
FunctionDeclaration = $node {
type: "FunctionDeclaration"
start: 33
end: 90
id: Identifier {type, start, end, name}
expression: false
generator: false
async: false
params: []
body: BlockStatement {
type: "BlockStatement"
start: 55
end: 90
body: [
ExpressionStatement {type, start, end, expression}
]
}
}
]
sourceType: "module"
}

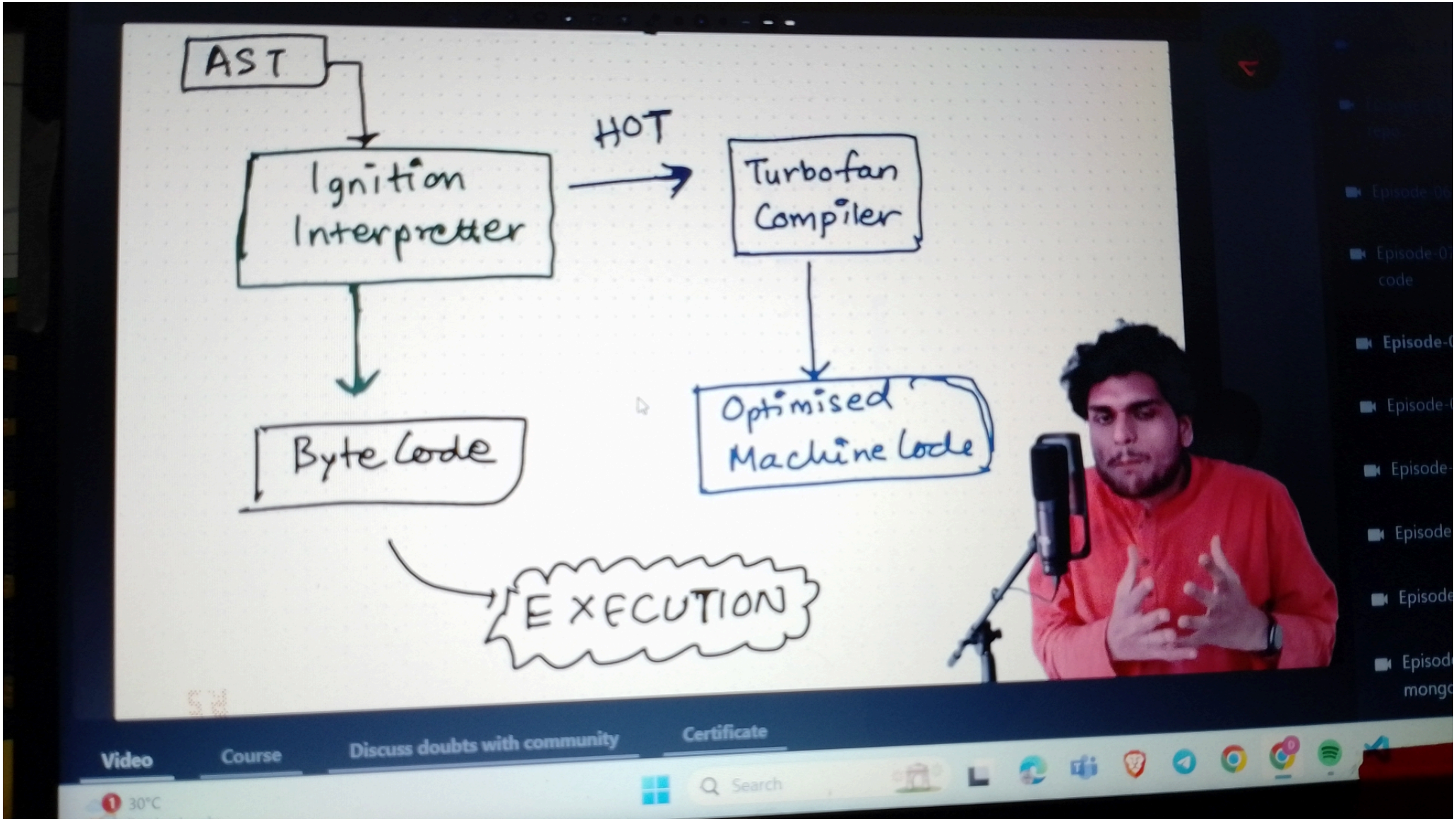

AST Explorer
Snippet
JavaScript
acorn
Transform
default
1 var x;
Tree
JSON
Parse
Autofocus
Hide methods
Hide empty keys
Hide location data
Hide type keys
- Program {
type: "Program"
start: 0
end: 6
- body: [
- VariableDeclaration {
type: "VariableDeclaration"
start: 0
end: 6
- declarations: [
- VariableDeclarator {
type: "VariableDeclarator"
start: 4
end: 5
- id: Identifier
type: "Identifier"
start: 4
end: 5
name: "x"
init: null
kind: "var"
sourceType: "module"

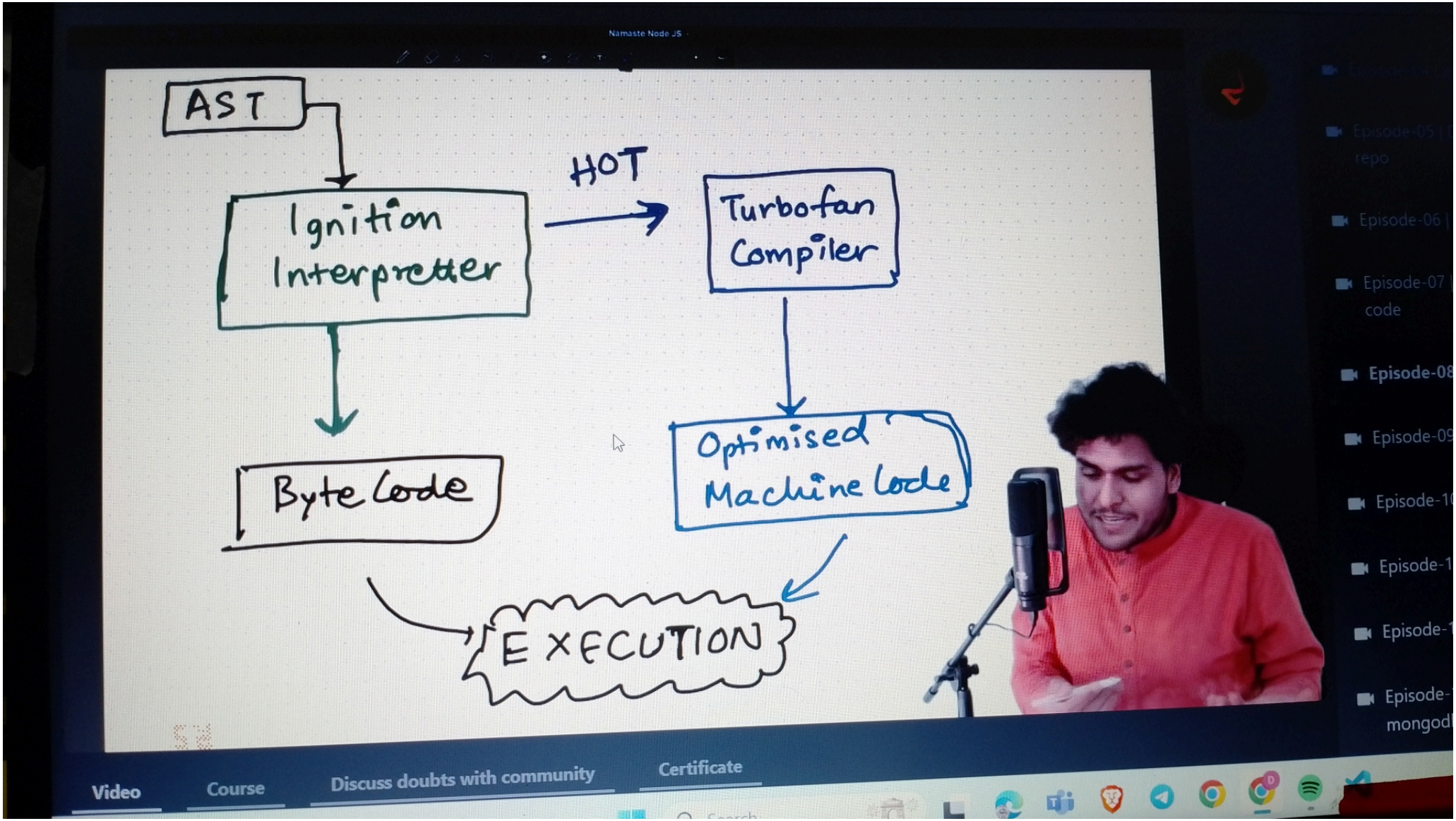

1 var x = {
Tree
JSON
Unexpected token (1:8)

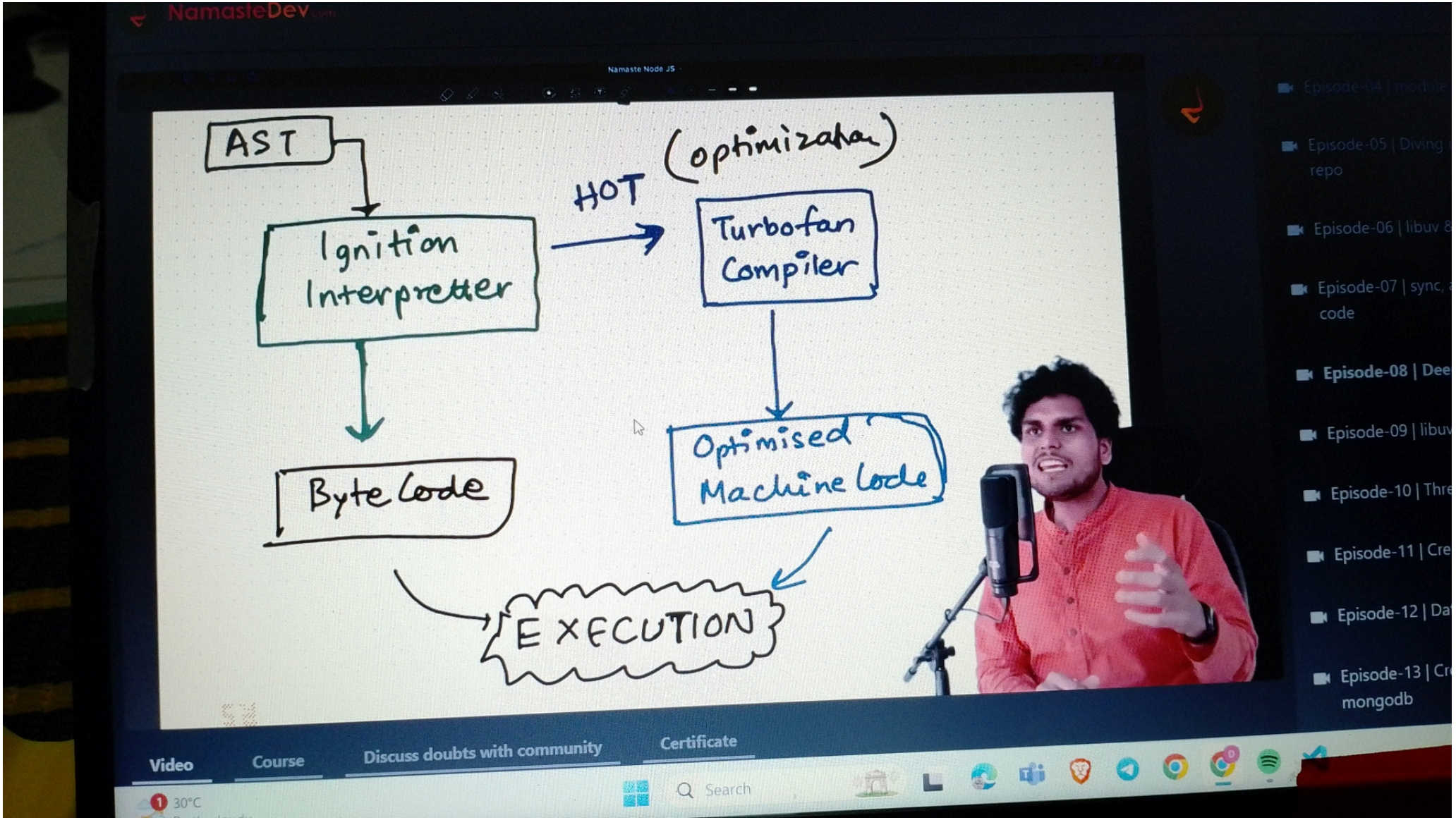

Namaste Node JS
AST
INTERPRETTER
2 Types of Languages
Interpreted
- Line by line
- Fast Initial execution
- Interpretter
Compiled
→ First compilation HL code → M. code
→ Initially heavy but executed fast
→ Compilers

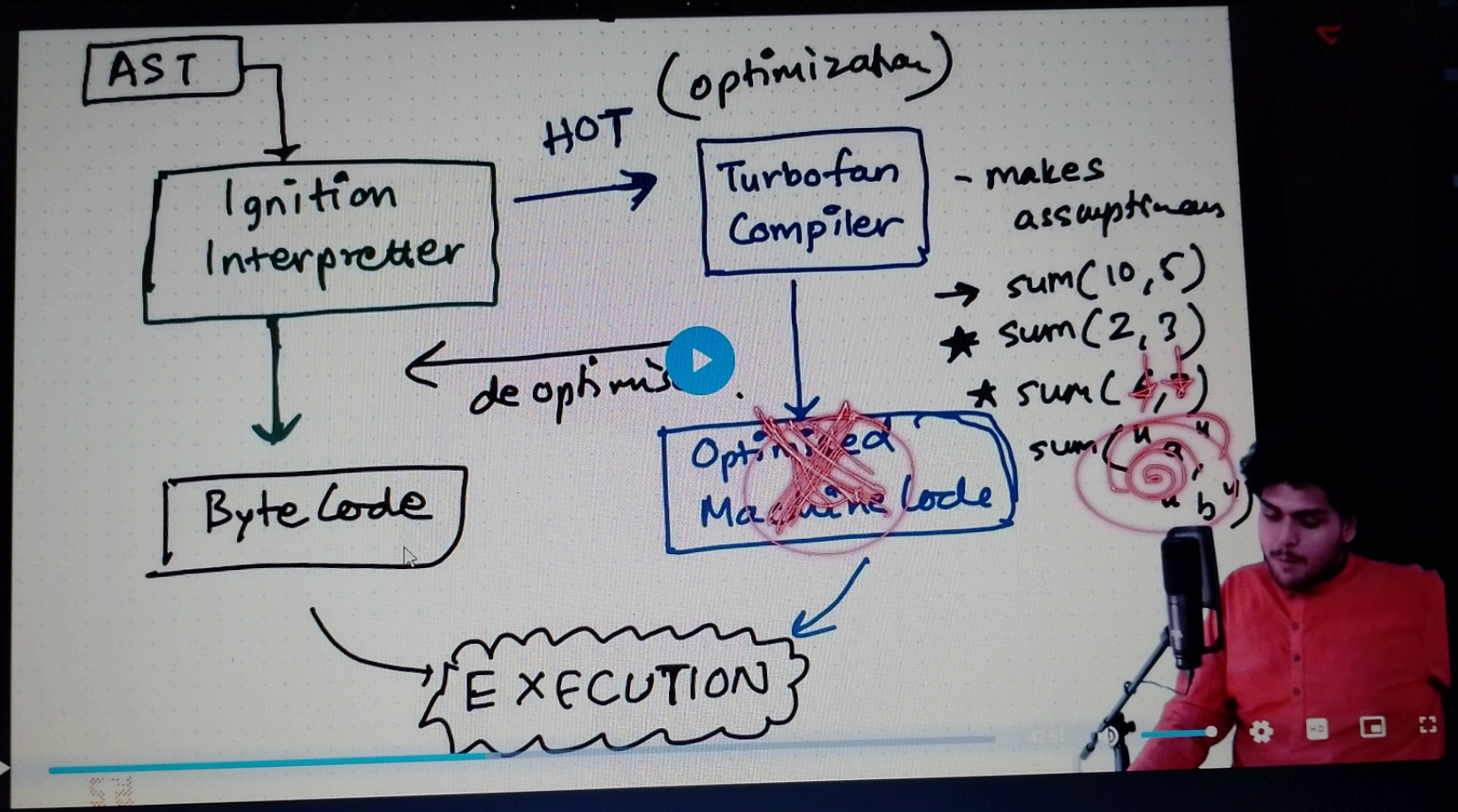

AST
INTERPRETTER
JIT compilation
(Just In Time)
2 Types of Languages
Interpreted
- Line by line
- Fast Initial execution
- Interpretter
Compiled
→ First compilation
HL code → M. code
→ Initially heavy
but executed fast
→ Compilers


AST
Ignition Interpreter
Byte Code
EXECUTION
Video
Course
Discuss doubts with community
Certificate
30°C
Search

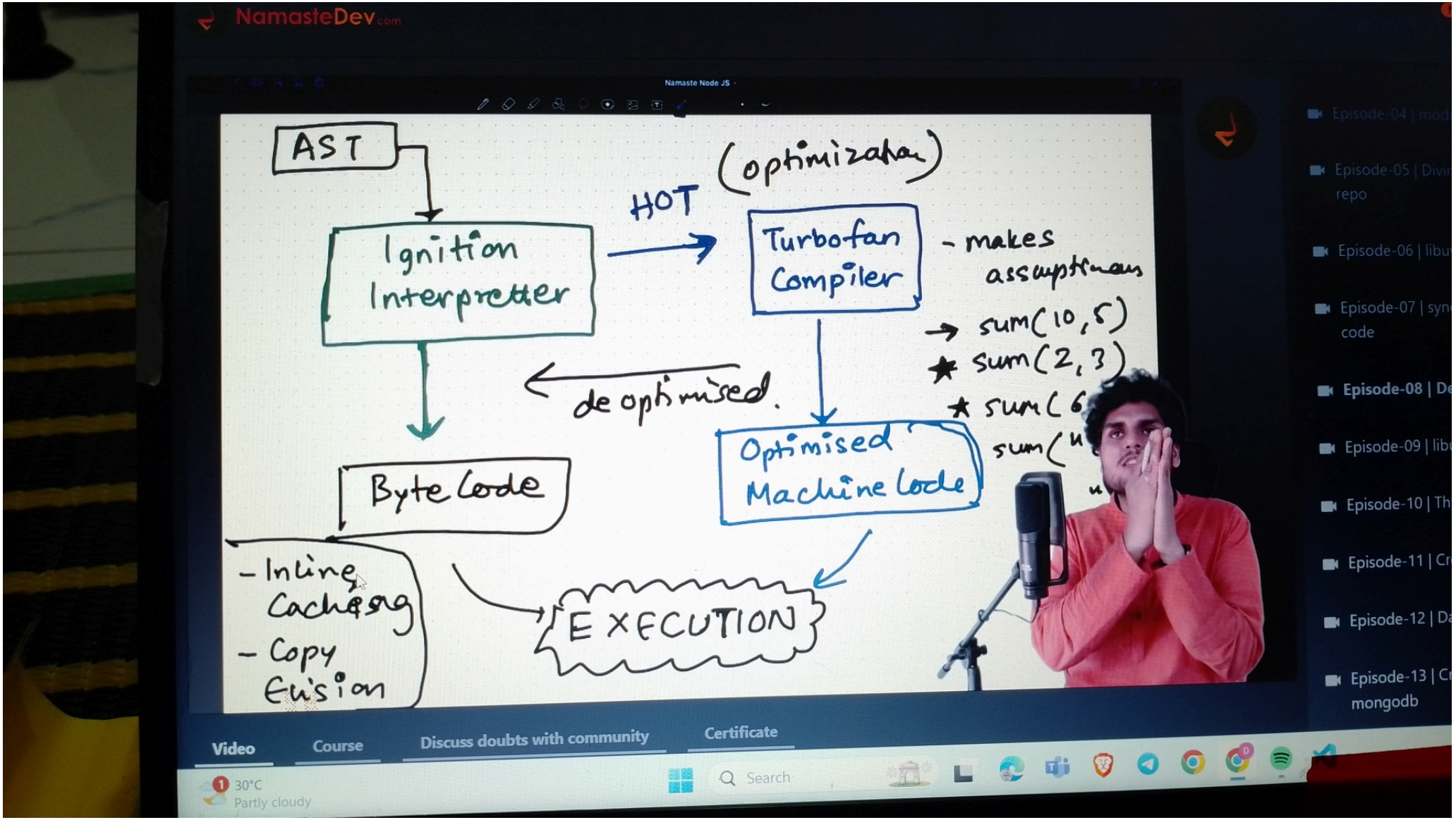

AST
Ignition Interpreter
HOT
Turbofan Compiler
Byte Code
Optimised Machine Code
EXECUTION
Video
Course
Discuss doubts with community
Certificate
Search
30°C

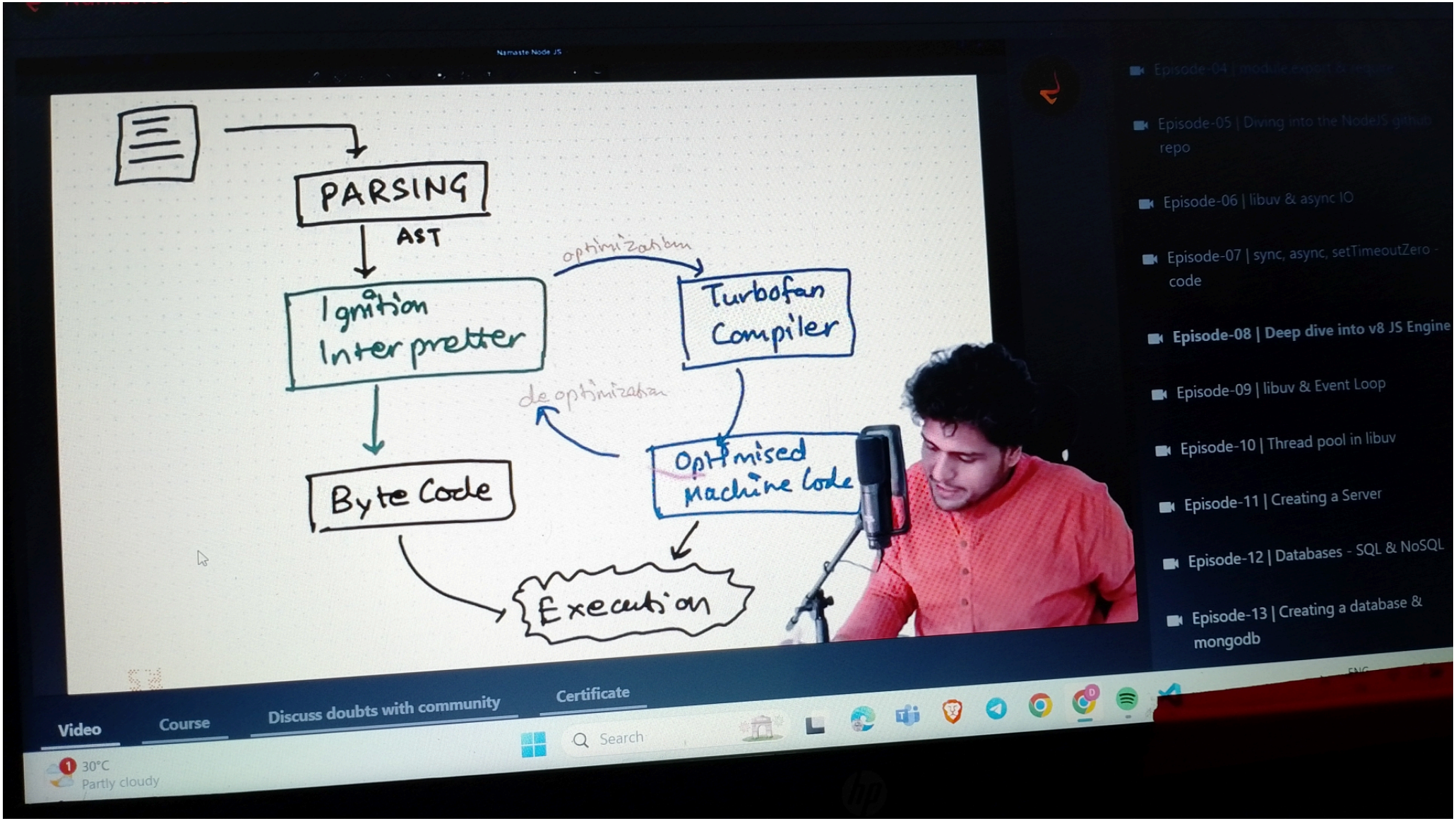

Namaste Node JS
AST
Ignition Interpreter
HOT
Turbofan Compiler
Byte Code
Optimised Machine Code
EXECUTION
Episode-05
repo
Episode-06
Episode-07
code
Episode-08
Episode-09
Episode-10
Episode-11
Episode-12
Episode-13
mongodb
Video
Course
Discuss doubts with community
Certificate

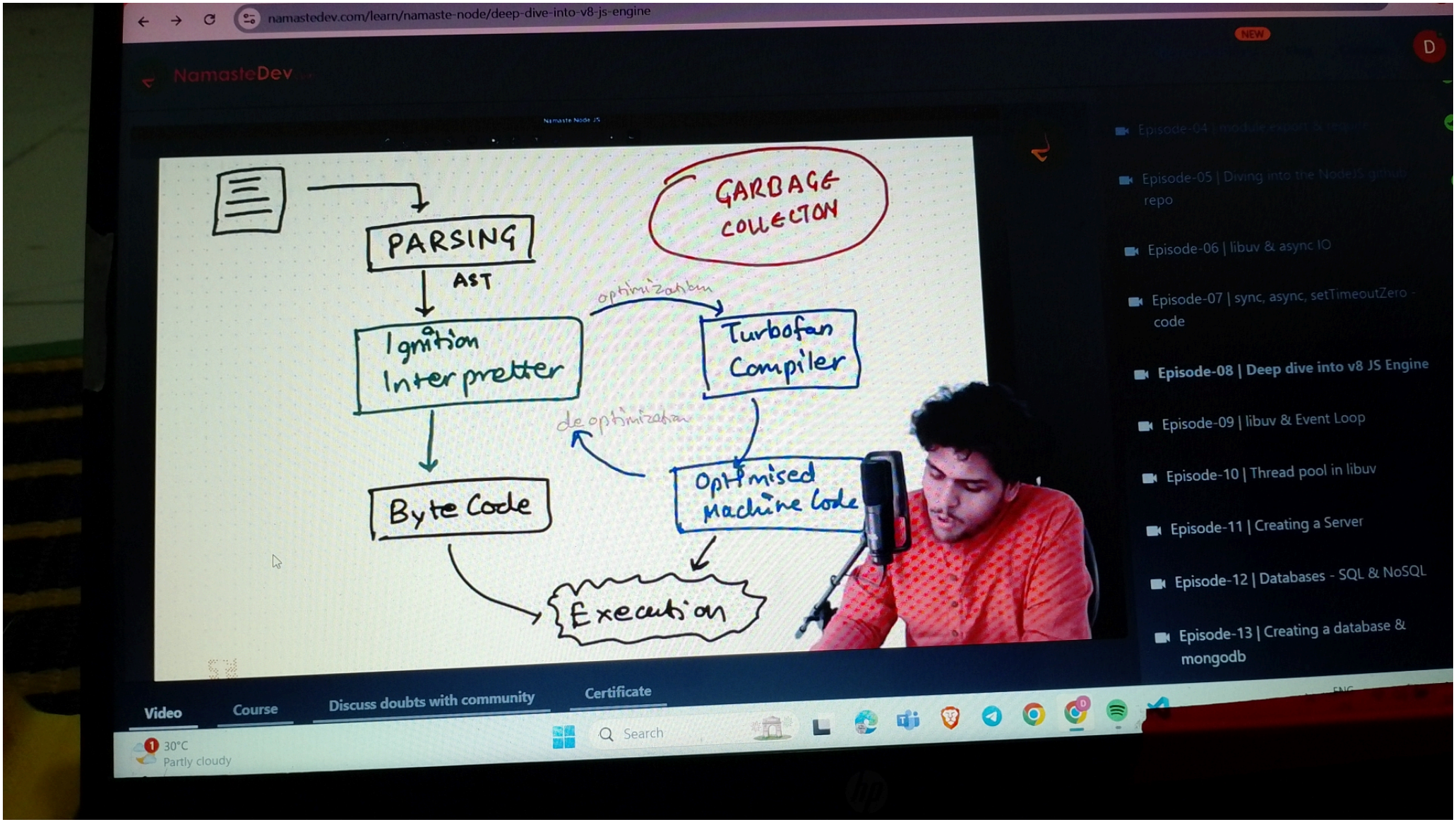

NamasteDev
Namaste Node JS
AST
Ignition Interpreter
HOT
(optimization)
Turbofan Compiler
Byte Code
Optimised Machine Code
EXECUTION
Episode-04 | modules
Episode-05 | Diving repo
Episode-06 | libuv
Episode-07 | sync, code
Episode-08 | Dee
Episode-09 | libuv
Episode-10 | Thre
Episode-11 | Cre
Episode-12 | Da
Episode-13 | Cr mongodb
Video
Course
Discuss doubts with community
Certificate
30°C
Search

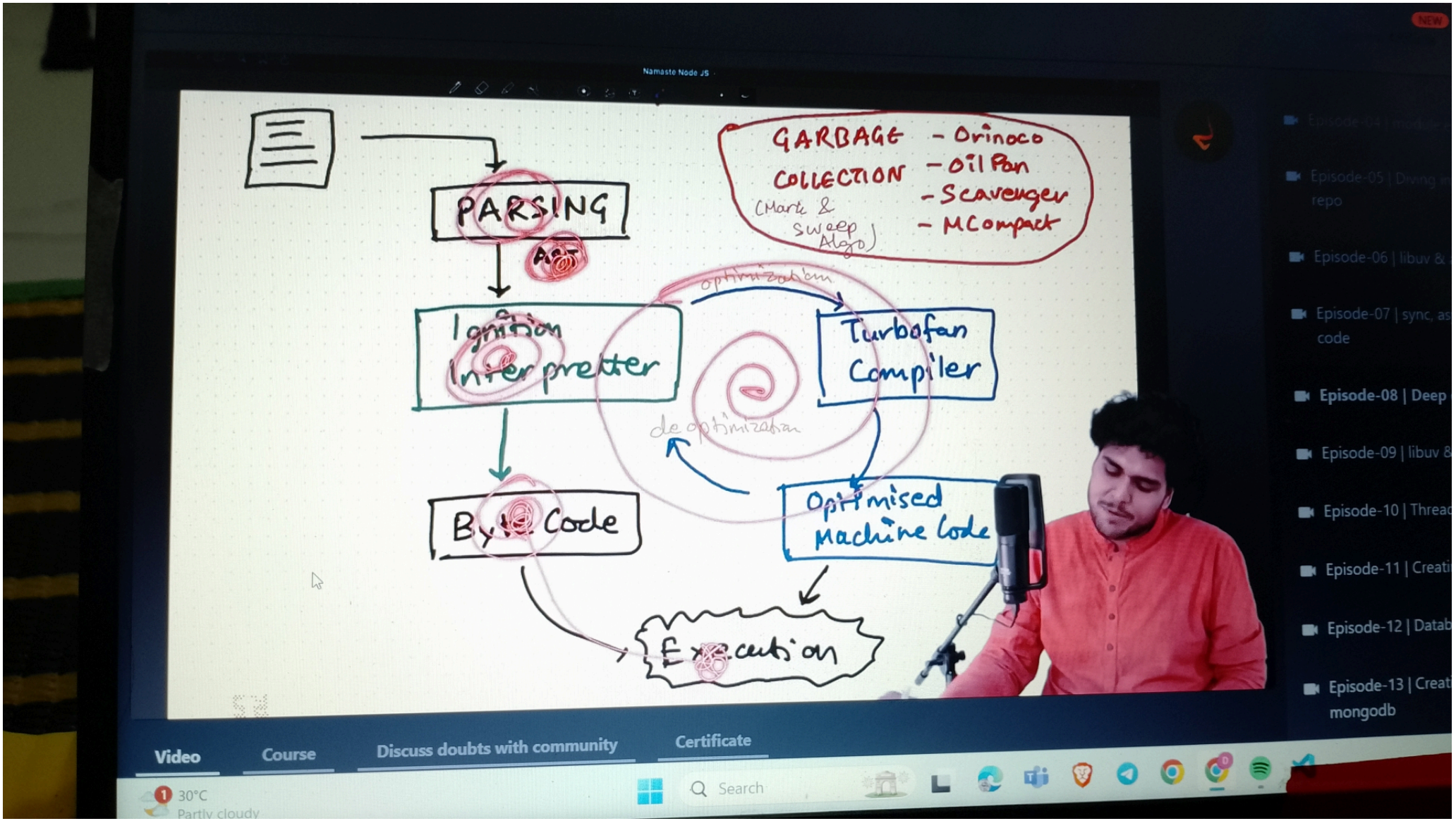


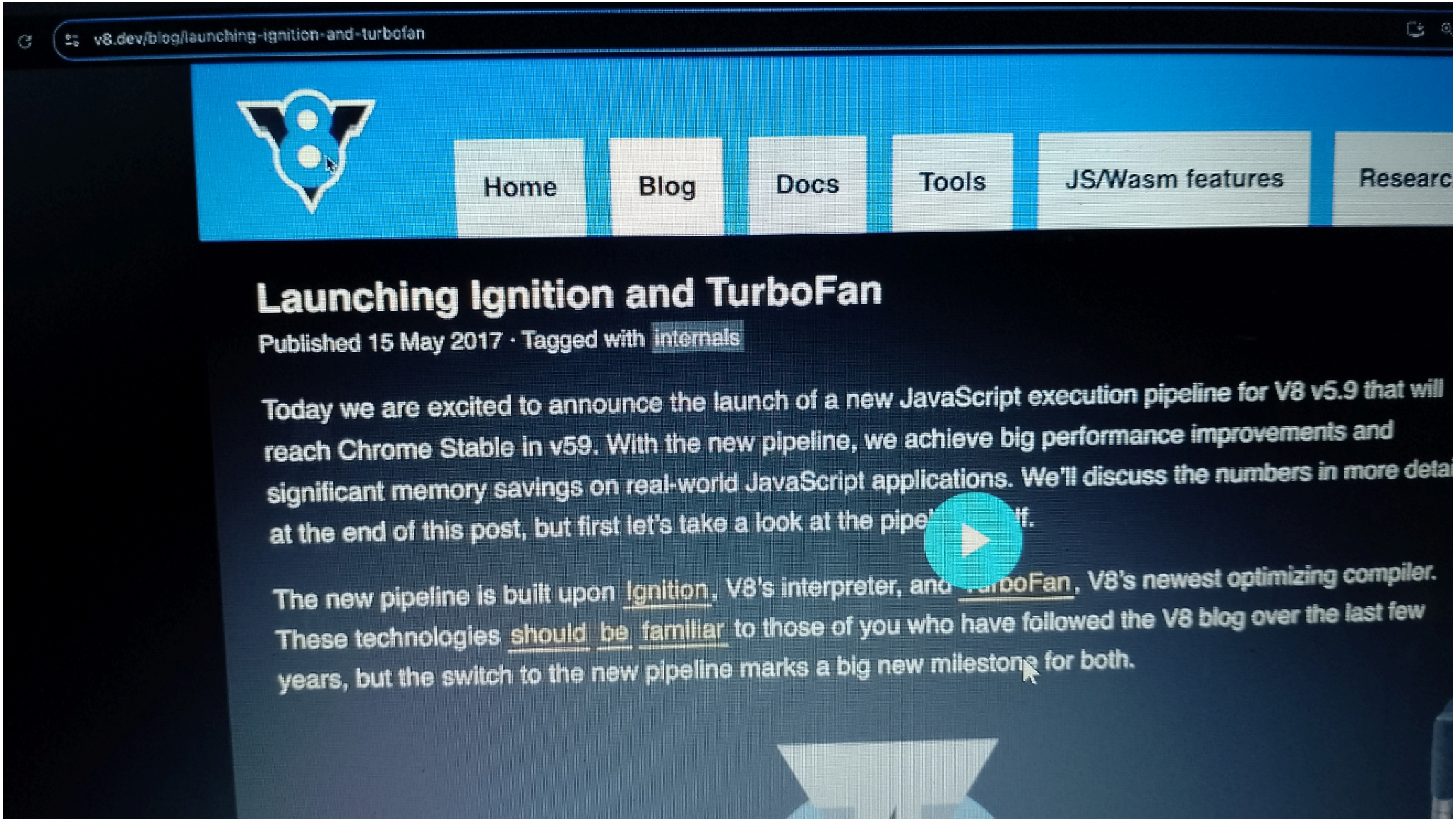
Video
Course
Discuss doubts with community
Certificate
30°C
Partly cloudy
Search


NamasteDev.com
Namaste Node JS
AST
Ignition Interpreter
HOT
(optimization)
Turbofan Compiler
- makes assumptions
sum(10,5)
sum(2,3)
sum(6
sum(u
u
de optimised.
Optimised Machine Code
Byte Code
- Inline Caching
- Copy Elision
EXECUTION
Episode-04 | mod
Episode-05 | Divi repo
Episode-06 | libu
Episode-07 | syn code
Episode-08 | De
Episode-09 | libu
Episode-10 | Th
Episode-11 | Cr
Episode-12 | Da
Episode-13 | Cr mongodb
Video
Course
Discuss doubts with community
Certificate
30°C
Partly cloudy
Search


Namaste Node JS
PARSING
AST
Ignition Interpretter
optimization
Turbofan Compiler
de optimization
Byte Code
Optimised Machine Code
Execution
Episode-04 | module.export & require
Episode-05 | Diving into the NodeJS github repo
Episode-06 | libuv & async IO
Episode-07 | sync, async, setTimeoutZero - code
Episode-08 | Deep dive into v8 JS Engine
Episode-09 | libuv & Event Loop
Episode-10 | Thread pool in libuv
Episode-11 | Creating a Server
Episode-12 | Databases - SQL & NoSQL
Episode-13 | Creating a database & mongodb
Video
Course
Discuss doubts with community
Certificate
Search
30°C
Partly cloudy


namastedev.com/learn/namaste-node/deep-dive-into-v8-js-engine
NamasteDev
PARSING
AST
Ignition Interpretter
optimization
Turbofan Compiler
GARBAGE COLLECTON
deoptimization
Byte Code
Optimised Machine code
Execution
Episode-04 | module.export & require
Episode-05 | Diving into the NodeJS github repo
Episode-06 | libuv & async IO
Episode-07 | sync, async, setTimeoutZero - code
Episode-08 | Deep dive into v8 JS Engine
Episode-09 | libuv & Event Loop
Episode-10 | Thread pool in libuv
Episode-11 | Creating a Server
Episode-12 | Databases - SQL & NoSQL
Episode-13 | Creating a database & mongodb
Video
Course
Discuss doubts with community
Certificate
30°C
Partly cloudy
Search


Namaste Node JS
PARSING
AST
GARBAGE COLLECTION
(Mark & sweep Algo)
- Orinoco
- Oil Pan
- Scavenger
- MCompact
Ignition Interpreter
optimization
Turbofan Compiler
de optimization
Byte Code
Optimised Machine Code
Execution
Episode-05 | Diving
repo
Episode-06 | libuv &
Episode-07 | sync, as
code
Episode-08 | Deep
Episode-09 | libuv &
Episode-10 | Thread
Episode-11 | Creati
Episode-12 | Datab
Episode-13 | Creat
mongodb
Video
Course
Discuss doubts with community
Certificate
30°C
Partly cloudy
Search


v8.dev/blog/launching-ignition-and-turbofan
Home
Blog
Docs
Tools
JS/Wasm features
Launching Ignition and TurboFan
Published 15 May 2017 · Tagged with internals
Today we are excited to announce the launch of a new JavaScript execution pipeline for V8 v5.9 that will reach Chrome Stable in v59. With the new pipeline, we achieve big performance improvements and significant memory savings on real-world JavaScript applications. We'll discuss the numbers in more detail at the end of this post, but first let's take a look at the pipeline itself.
The new pipeline is built upon Ignition, V8's interpreter, and TurboFan, V8's newest optimizing compiler. These technologies should be familiar to those of you who have followed the V8 blog over the last few years, but the switch to the new pipeline marks a big new milestone for both.