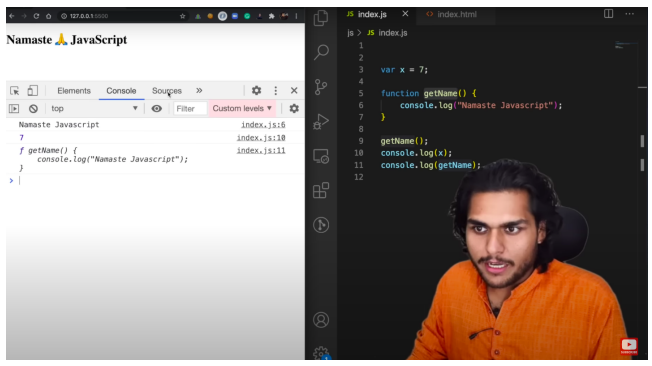
Hoisting in JavaScript

Tuesday, 15 August 2023 8:29 PM

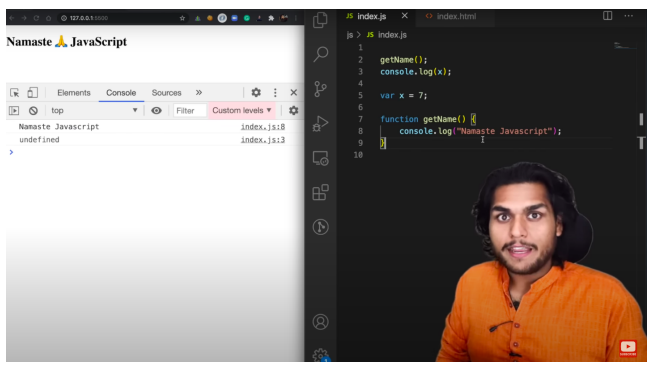
Hoisting in JS

→ Hoisting is a phenomena in JS where we can access a variable or fⁿ even before their initialization.

Note: If we try to console.log a fⁿ before its initialization then the whole fⁿ gets consoled in JS, this is because during memory creation phase the whole fⁿ code gets stored in memory.



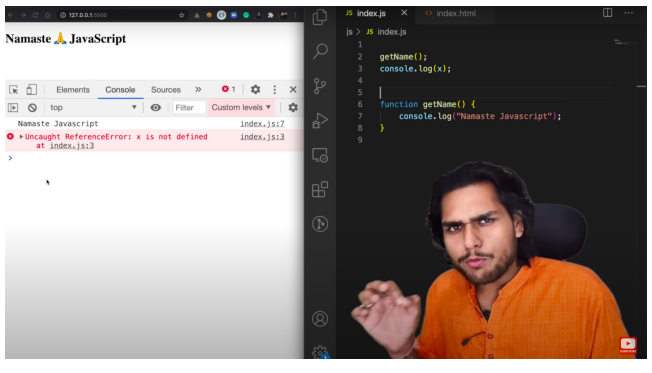
(Initial code o/p)



→ During memory creation phase, variables are given undefined value and fⁿ are stored as it is in memory. So if we try to access a variable before its initialization then it gives us undefined whereas it prints the whole fⁿ code if we try to console the fⁿ.

Q. Difference between undefined & not defined?

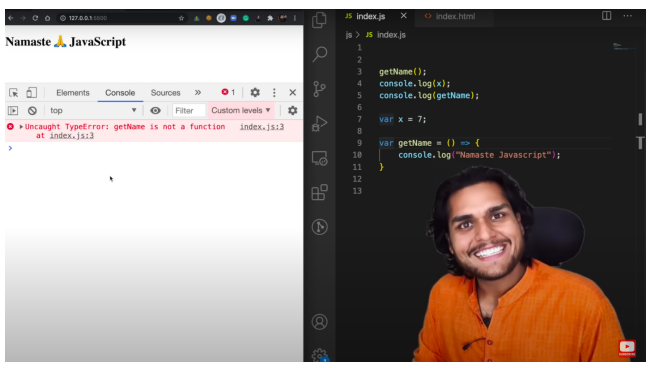
ans.

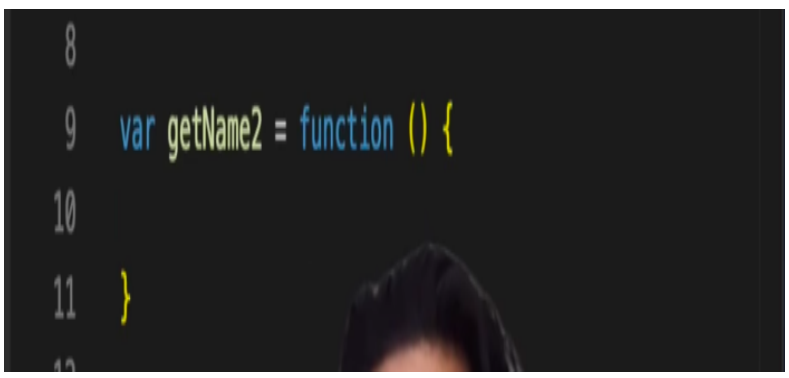


If a variable is accessed before its initialization then since JS has stored it as undefined during memory creation phase.

thus it prints undefined.

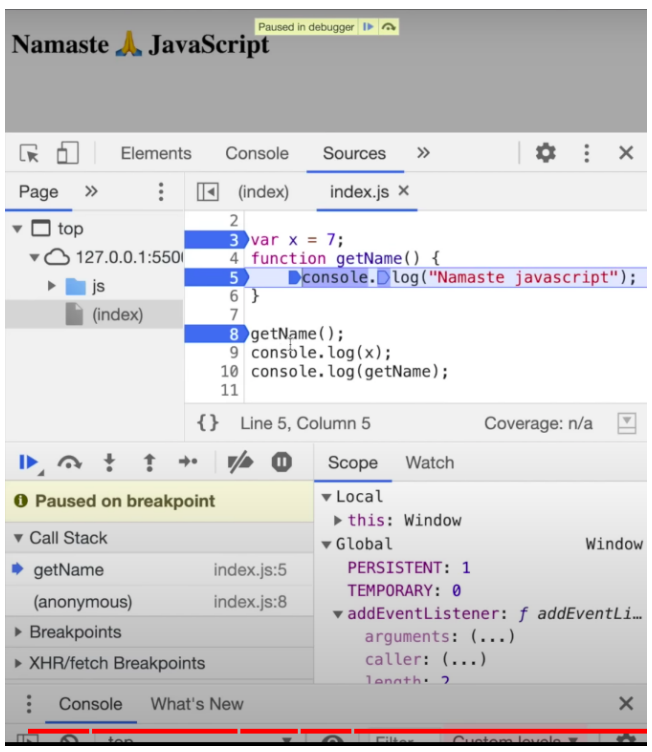
Whereas, if a variable has not been declared and it is accessed then it gives the above reference error of not defined as it is not present in the global memory.

→ 



If we use an arrow fn, then the fn is treated like a variable only. So, during memory creation phase this fn is stored as undefined, that's why we get the above error.

Note: In 2nd case also fn is treated like a variable.



→ Real time working of a call stack.