episode	How javascript code is executed, coll stack
	Phase Phase
phase (memory greation phase rode execution phase
byass	n : undefined node is stored n: 2 hothing to do with 2nd
able:	> n: undefined in code 15.54000 n: 2 line]
vojuble_	= square: finite whole (s) Again execution context will
function —	> n: undefined > n: undefined > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > square: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) \(\text{unote (ode is stored)} \) > sequare: \(\frac{1}{2} \) > sequare: \(\frac{1}{2} \) \(\frac{1}{2} \) > sequare: \(\frac{1}{2} \) \(\fra
June	function invoke
	Memora component code component
	Memory component code component Again two phases
	Memory reation phase code execution phase
	Again two phases memory reation phase num: undefined ons: calculation i.e.
	Again two phases memory reation phase num: undefined ans: undefined ans: calculation i.e.
	Again two phases memory (reation phase rode execution phase num: undefined num: 2 ans: undefined ans: calculation i.e. Two get this value numxnum = 2×2=4
	Again two phases memory reation phase num: undefined ons: calculation i.e.

Global ex Contex	ecution = Innerexec innerexec context = Context = Context deleted
	example which we consider in code execution:
F	106
	vas n=2; parametes [Hose]
	function square (num) i when ans gets
	vor ans = num * num; seturned then
	getugn ons; inner execution
4	3 argument (ontext will be
, ×	Von square2 = square(n); deleted
	Call Stack [all above critical things are managed]
, hi ji ka jiha	call stack [all above (sitical things are managed] by constack
	Tuhen a new function
, . ·	- Long new takes
	execution context 2 Twhen a new context
	execution context 1 execution context 1 execution context 1 execution context 1
	wood T again Calobal execution content but
Eni	1) have deat
[0]	deted now popea out at very fixst at bottom side
ing a second	
` .	- Call stack maintain the line of
	execution" of execution contexts.
, 1	Thereally of PARLUTION TURLENTS.
	- Call stack fancy names:
	- Cull stack tancy names: - ① execution context stack
	2 program stuck
	3 Control stack
	9 runtime stack
	5 machina chack
	STUCE STUCE
13.40	All homes are call stack itself.
	TUIT STUCK ITSELT.
	V