

episode 04

How functions work / Variable Environment

[also called as memory creation phase]

[function execution and invoking function]

code

```

- var x=1;
  a();
  b();
  (console.log(x);
  function a() {
    var x=10;
    console.log(x);
  }
  function b() {
    var x=100;
    console.log(x);
  }

```

Answers in console

10
100
1

Global execution context

variable environment

Memory	Code				
x: undefined	var x=1				
a: f...3	<table border="1"> <thead> <tr> <th>M</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>10</td> </tr> </tbody> </table>	M	C	x	10
M	C				
x	10				
b: f...3	<table border="1"> <thead> <tr> <th>M</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>100</td> </tr> </tbody> </table>	M	C	x	100
M	C				
x	100				
	console.log(x)				

Execution context due to function invocation of a();
[due to invocation of b();]

call stack

	Execution context due to b()
	Execution context due to a()
GEC	Global execution context

code / context

summary :- Whenever a function is invoked in javascript a functional execution context is created and memory is allocated. Once the memory is allocated to the variables and functions, then code is executed synchronously, one line at a time.