Date: 01/01/2024

DAY 1 - JAVASCRIPT

#100Daysofcode

Execution content:

- Everything in Javascript happens inside an Execution content.

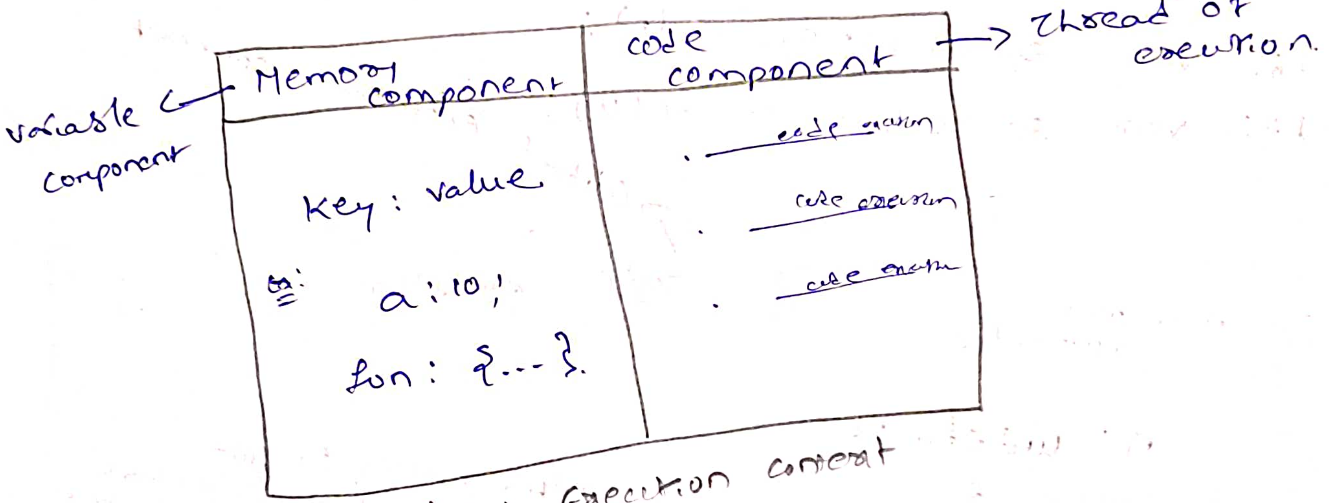


fig: 1 Execution content

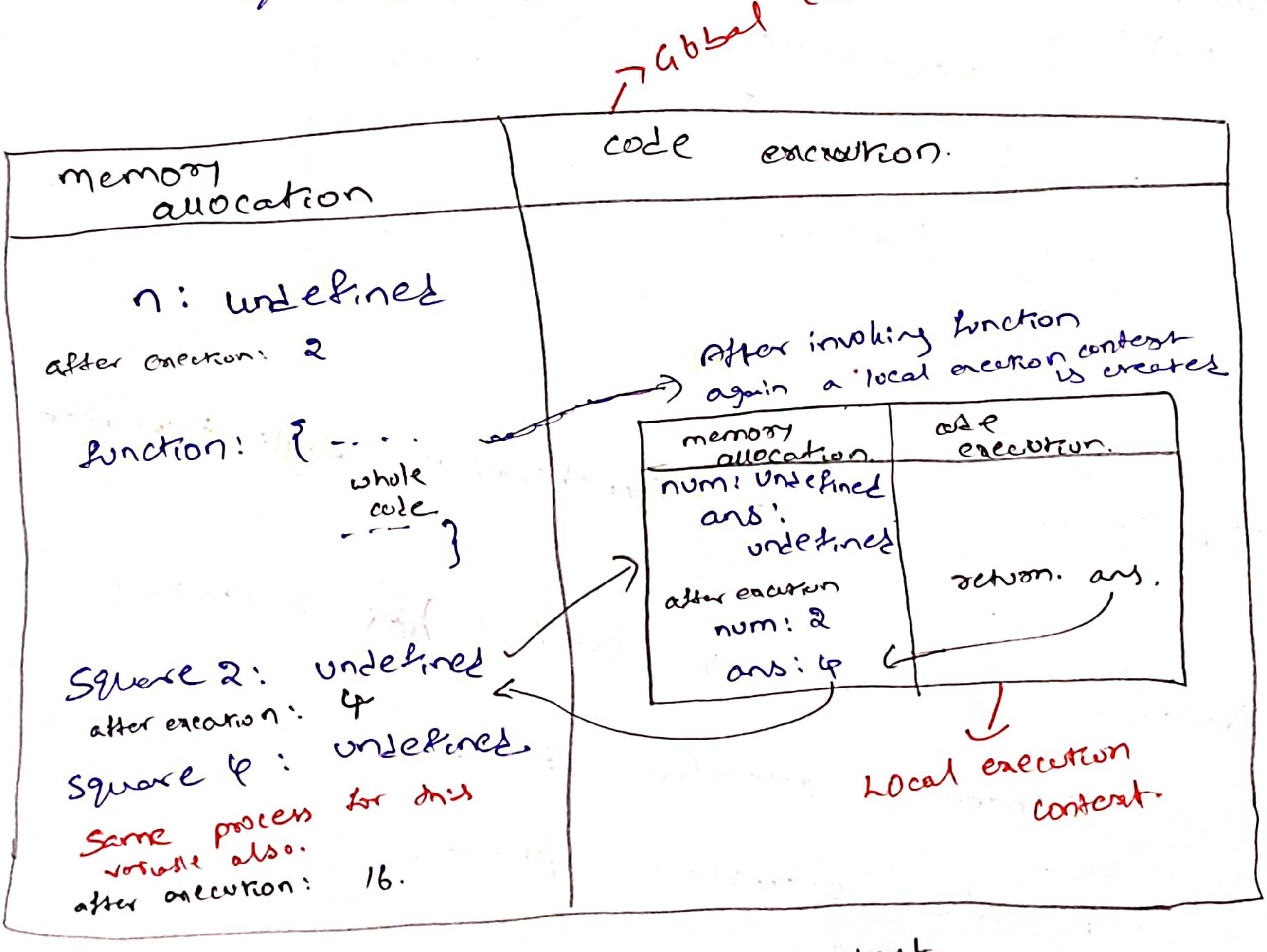
- JavaScript is a synchronous single-threaded language.
- Execution content is created ~~bias~~ in 2 phases.

  1. memory allocation phase
  2. code execution phase.

- When the code execution is started JS engine creates a Global Execution content.
- For every function invocation a local execution content is created.

code:

```
Var     n=2;
function     square(num) {
          var   ans =  num  *  num ;
          return   ans;
}
var    square2 = square(n);
var    square4  =  square(4);
```



Global execution context

memory allocation | code execution

n: undefined
after execution: 2

function: { . . . whole code . . . }

square2: undefined
after execution: 4

square4: undefined
Same process for this variable also.
after execution: 16.

After invoking function again a local execution context is created

memory allocation | code execution

num: undefined
ans: undefined
after execution
num: 2
ans: 4

return. ans.

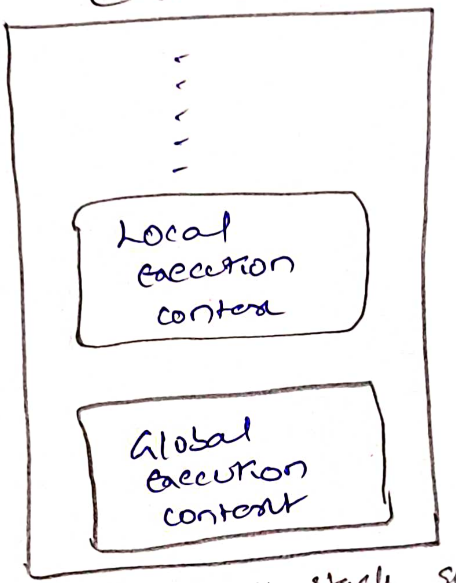
Local execution context.

fig 2: Execution context.

- Javascript controls these type of complex executions through a call stack.

- callstack maintains the order of execution of execution contexts.

call stack. → Also known as



1. Execution context stack
2. program stack
3. control stack
4. Runtime stack
5. Machine stack.

fig 3: call stack queue.

- After all the code execution is completed the callstack becomes empty.