

JavaScript

Under The Hood

Part 2: Execution Context

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

JS





<https://vizle.offnote.co>

Contact us: vizle@offnote.co

This document was generated automatically by **Vizle**

Your **Personal Video Reader Assistant**

Learn from Videos **Faster** and **Smarter**

VIZLE **PRO / BIZ**

PDF, PPT ~~Watermarks~~

- Convert *entire* videos
- *Customize* to retain all essential content
- Include Spoken *Transcripts*
- Customer support

Visit <https://vizle.offnote.co/pricing> to learn more

VIZLE **FREE PLAN**

PDF only ~~Watermarks~~

- Convert videos *partially*
- Slides may be *skipped**
- Usage restrictions
- No Customer support

Visit <https://vizle.offnote.co> to try free

Login to Vizle to unlock more slides*

Execution Context Phases

Memory Creation Phase:

1. Create the global object (browser = window, Node.js = global)
2. Create the 'this' object and bind it to the global object
3. Setup memory heap for storing variables and function references
4. Store functions and variables in global execution context and set to "undefined"

Execution Phase:

1. Execute code line by line
2. Create a new execution context for each function call

```

1 var x = 100
2 var y = 50
3 function getSum(n1, n2) {
4   var sum = n1 + n2
5   return sum
6 }
7 var sum1 = getSum(x, y)
8 var sum2 = getSum(10, 5)

```



Function EC Creation Phase:

Line 3: $n1$ & $n2$ variables are allocated memory and stores “undefined”

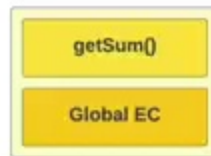
Line 4: sum variable is allocated memory and stores “undefined”

Function EC Execution Phase:

Line 3: $n1$ & $n2$ are assigned 100 and 50

Line 4: Calculation is done and 150 is put into the sum variable

Line 5: return tells the function EC to return to the global EC with value of sum (150)



CALL STACK

main.js x index.html

main.js > ...

```
1 var x = 100;
2 var y = 50;
3
4 function getSum(n1, n2) {
5   var sum = n1 + n2;
6   return sum;
7 }
8
9 var sum1 = getSum(x, y);
10 var sum2 = getSum(10, 5);
11
```

Port: 5500 ✓ Spell ✓ Prettier

JavaScript Under The Hood

Paused in debugger

Elements Console Recorder Sources

Page index.html main.js x

127.0.0.1:5500

index.html main.js

```
1 var x = 100;
2 var y = 50;
3
4 function getSum(n1, n2) {
5   var sum = n1 + n2;
6   return sum;
7 }
8
9 var sum1 = getSum(x, y);
10 var sum2 = getSum(10, 5);
11
```

main.js:1
var x = 100;

Scope

Local

- this: Window
- n1: 100
- n2: 50
- sum: undefined

Global

- GetParams: f e(t)
- alert: f alert()
- atob: f atob()
- blur: f blur()
- btoa: f btoa()
- caches: CacheStorage {}
- cancelAnimationFrame: f cancelAnimationFrame()

Window

Line 5, Column 13

Coverage: n/a

Console What's New Issues

Filter

Default levels 1 Issue: 1 1 hidden

main.js — sandbox

main.js x index.html

main.js > ...

```
1 console.log(x);
2
3 var x = 100;
4 var y = 50;
5
6 console.log(getSum(x, y));
7
8 function getSum(n1, n2) {
9   var sum = n1 + n2;
10  return sum;
11 }
12
```

JavaScript Under The Hood

127.0.0.1:5500/index.html

Elements Console Recorder Sources

Filter

undefined

150

main.js:1

main.js:6

Console What's New Issues

Port: 5500

Quokka

Spell

Prettier

Vizle



<https://vizle.offnote.co>

Contact us: vizle@offnote.co

This document was generated automatically by **Vizle**

Your **Personal Video Reader Assistant**

Learn from Videos **Faster** and **Smarter**

VIZLE **PRO / BIZ**

PDF, PPT ~~Watermarks~~

- Convert *entire* videos
- *Customize* to retain all essential content
- Include Spoken *Transcripts*
- Customer support

Visit <https://vizle.offnote.co/pricing> to learn more

VIZLE **FREE PLAN**

PDF only ~~Watermarks~~

- Convert videos *partially*
- Slides may be *skipped**
- Usage restrictions
- No Customer support

Visit <https://vizle.offnote.co> to try free

Login to Vizle to unlock more slides*