

☰ Hoisting in JavaScript 🔥 (variables & functions) | Namaste JavaScript Ep. 3



127.0.0.1:5500

Namaste 🙏 JavaScript

Elements Console Sources >>

top Filter Custom levels

Namaste Javascript index.js:6

7 index.js:10

>

JS index.js X index.html

```
js > JS index.js
1
2
3   var x = 7;
4
5   function getName() {
6     console.log("Namaste Javascript");
7   }
8
9   getName();
10  console.log(x);
11
```



127.0.0.1:5500

Namaste 🙏 JavaScript

Elements Console Sources >>

top Filter Custom levels

Namaste Javascript index.js:9

undefined index.js:3

```
f getName() {  
  console.log("Namaste Javascript");  
}
```


>

hoisting is a phenomena by which we can access a variable or a function even before declaring it.

Function starting with let, var or const are not defined by default in memory execution phase

JS index.js X index.html

```
js > JS index.js  
1  
2  getName();  
3  console.log(x);  
4  console.log(getName);  
5  
6  var x = 7;  
7  
8  function getName() {  
9    console.log("Namaste Javascript");  
10 }  
11
```



☰ Hoisting in JavaScript 🔥 (variables & functions) | Namaste JavaScript Ep. 3



127.0.0.1:5500

Namaste 🙏 JavaScript

Elements Console Sources >> 1

top Filter Custom levels

Namaste Javascript [index.js:7](#)

✖ ▶ Uncaught ReferenceError: x is not defined [index.js:3](#)


>

JS index.js x index.html

```
js > JS index.js
1
2  getName();
3  console.log(x);
4
5
6  function getName() {
7    console.log("Namaste Javascript");
8  }
9
```



Namaste 🙏 JavaScript



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The top toolbar includes icons for navigation, a search bar, and a 'Filter' button. The console output shows a function call `getName()` at `index.js:11`, which logs the message `"Namaste Javascript"` to the console.

JS index.js X index.html

```
js > JS index.js
1
2 // getName();
3 // console.log(x);
4
5 var x = 7;
6
7 function getName() {
8     console.log("Namaste Javascript");
9 }
10
11 console.log(getName);
```



127.0.0.1:5500

Namaste 🙏 JavaScript

Elements Console Sources >>


top Filter Custom levels

```
f getName() {  
  console.log("Namaste Javascript");  
}
```

index.js:4

JS index.js index.html

```
js > JS index.js  
1  
2 // getName();  
3 // console.log(x);  
4 console.log(getName);  
5  
6 var x = 7;  
7  
8 function getName() {  
9   console.log("Namaste Javascript");  
10 }  
11
```



Hoisting in JavaScript 🔥 (variables & functions) | Namaste JavaScript Ep. 3

Namaste 🙏 JavaScript

Elements Console Sources >> | ⚙️ ⋮ ✕

▶️ 🔇 top ▼ | 👁️ Filter Custom levels ▼ | ⚙️

Namaste Javascript	index.js:9
undefined	index.js:3
f getName() { console.log("Namaste Javascript"); }	index.js:4

>

JS index.js ✕ <> index.html

```
js > JS index.js
1
2  getName();
3  console.log(x);
4  console.log(getName);
5
6  var x = 7;
7
8  function getName() {
9    console.log("Namaste Javascript");
10 }
11
```



JS main.js

JS main.js > ...

```
1 var x = 6;
2 function printName() {
3   console.log("namaste world");
4 }
5 let summer = () => {
6   console.log("Happy faces");
7 }
8
9 printName();
10 console.log(printName);
11 console.log(x);
12 summer();
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

Code + -

```
● (base) summerkoushal@summers-MacBook-Air test % node "/Users/summerkoushal/Desktop/college/skills/web/test/main.js"
namaste world
[Function: printName]
6
Happy faces
○ (base) summerkoushal@summers-MacBook-Air test %
```

Run Testcases Sign in to Jira Sign in to Bitbucket 0 0

Spaces: 4 UTF-8 LF {} JavaScript Go Live Prettier

Hoisting in JavaScript 🔥 (variables & functions) | Namaste JavaScript Ep. 3

Namaste 🙏 JavaScript

Elements Console Sources >> 1 | ⚙️ ⋮ ✕

top Filter Custom levels ⌵ ⚙️

✕ ▶ Uncaught TypeError: getName is not a function index.js:3
at index.js:3

// func2; //ReferenceError: func2 is not initialised - not a func but a var
// func2(); //ReferenceError: func2 is not initialised- not a func but a var

```
let func2 = () => {  
  console.log("function one");  
}
```

```
func2; // print function  
func2(); // invoke the function
```

JS index.js ✕ index.html

```
js > JS index.js  
1  
2  
3  getName();  
4  console.log(x);  
5  console.log(getName);  
6  
7  var x = 7;  
8  
9  var getName = () => {  
10   console.log("Namaste Javascript");  
11 }  
12  
13
```



JS main.js ×

JS main.js > [orange]

```
1  printName();
2  console.log(printName);
3  console.log(x);
4  summer(); //Cannot access 'summer' before initialization
5  orange(); //Cannot access 'orange' before initialization
6  // var, let, const functions and variables are declared uninitialized before code execution. only function
7  // can be extracted before it's declaratoin.
8  var x = 6;
9  function printName() {
10 |   console.log("namaste world");
11 | }
12  let summer = () => {
13 |   console.log("Happy faces");
14 | }
15  let orange = function() {
16 |   console.log("bombay ka ilaka");
17 | }
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

> Code + - □ □ ^ ×

⊗ (base) summerkoushal@summers-MacBook-Air test % node "/Users/summerkoushal/Desktop/college/skills/web/test/main.js"

```
namaste world
[Function: printName]
undefined
/Users/summerkoushal/Desktop/college/skills/web/test/main.js:4
summer();
^
```

ReferenceError: Cannot access 'summer' before initialization

```
at Object.<anonymous> (/Users/summerkoushal/Desktop/college/skills/web/test/main.js:4:1)
at Module._compile (node:internal/modules/cjs/loader:1159:14)
at Module._extensions..js (node:internal/modules/cjs/loader:1213:10)
```

Run Testcases Sign in to Jira Sign in to Bitbucket 0 0

Spaces: 4 UTF-8 LF {} JavaScript Go Live ✓ Prettier

127.0.0.1:5500

Namaste 🙏 JavaScript

Paused in debugger

Elements Console Sources

Page >> (index) index.js x

```
2
3 var x = 7;
4 function getName() {
5   console.log("Namaste javascript");
6 }
7
8 getName();
9 console.log(x);
10 console.log(getName);
11
```

Line 5, Column 5 Coverage: n/a

Paused on breakpoint

Call Stack

- getName index.js:5
- (anonymous) index.js:8

Breakpoints

XHR/fetch Breakpoints

Console What's New

Scope Watch

Local


- this: Window

Global Window

- PERSISTENT: 1
- TEMPORARY: 0
- addEventListener: f addEventLi...
- arguments: (...)
- caller: (...)
- length: 2

JS index.js x index.html

```
js > JS index.js
1
2
3 var x = 7;
4 function getName() {
5   console.log("Namaste javascript");
6 }
7
8 getName();
9 console.log(x);
10 console.log(getName);
11
```



SUBSCRIBE