



## What is the Call Stack?

The **call stack** is a mechanism in JavaScript used to manage the execution of functions. It's a **data structure (LIFO - Last In, First Out)** that keeps track of function calls in the order they need to be executed. It helps the JavaScript engine know where to return after completing a function.

---

```
function first() {  
  console.log("First function");  
  second();  
}
```

```
function second() {  
  console.log("Second function");  
  third();  
}
```

```
function third() {  
  console.log("Third function");  
}
```

```
first();
```



## Step-by-Step Call Stack Execution:

1. `first()` is called:
  - `first` is pushed onto the stack.
2. Inside `first`, `second()` is called:
  - `second` is pushed onto the stack.
3. Inside `second`, `third()` is called:
  - `third` is pushed onto the stack.
4. `third()` completes:
  - `third` is popped off the stack.

5. `second()` completes:

- `second` is popped off the stack.

6. `first()` completes:

- `first` is popped off the stack.

## Key Points

- The call stack operates on LIFO (Last In, First Out).
- It keeps track of function execution order.
- When a function completes, it is removed (popped) from the stack.
- Errors like "stack overflow" happen when the stack exceeds its maximum size.