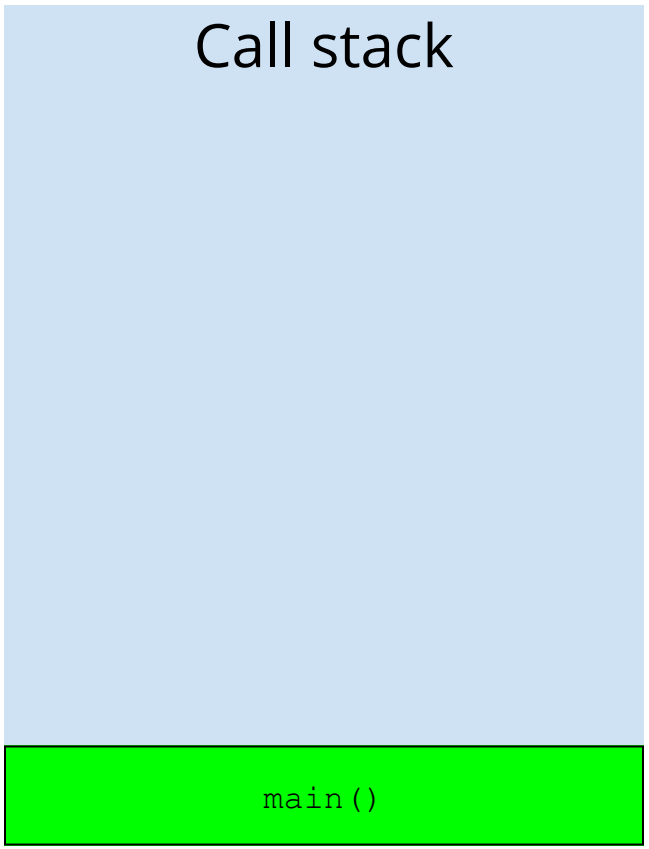

node
JS

CALL STACK
&
EVENT LOOP

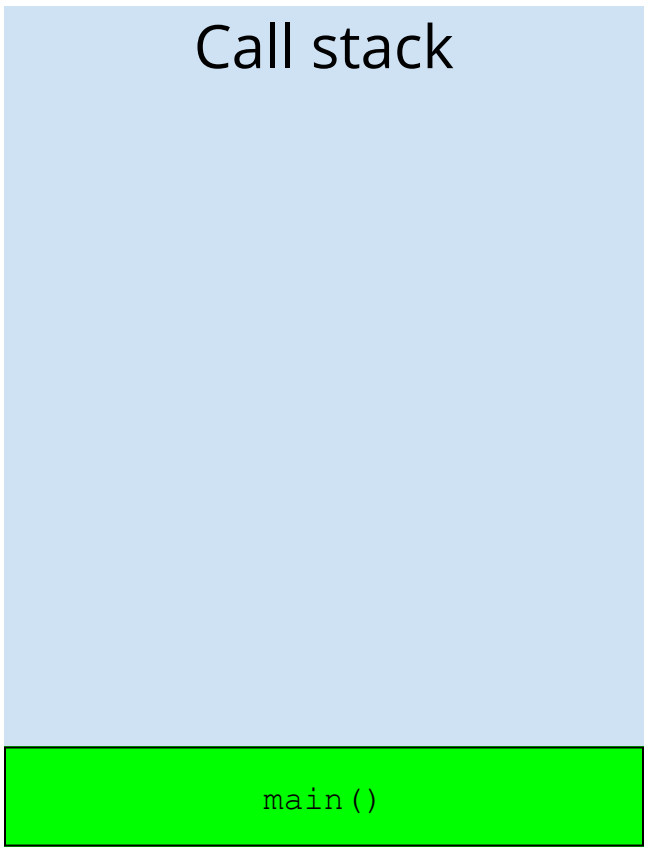
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```

Call stack

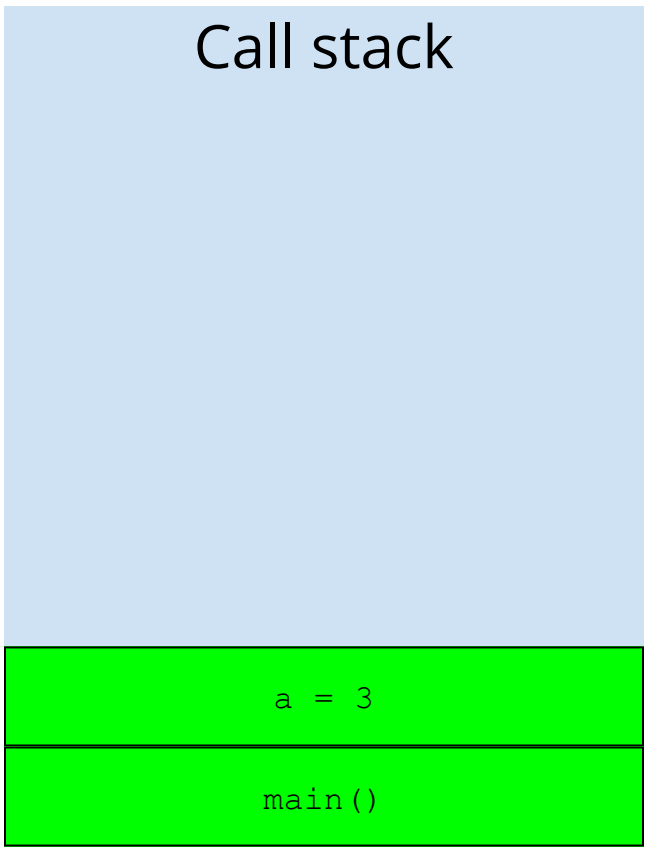
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



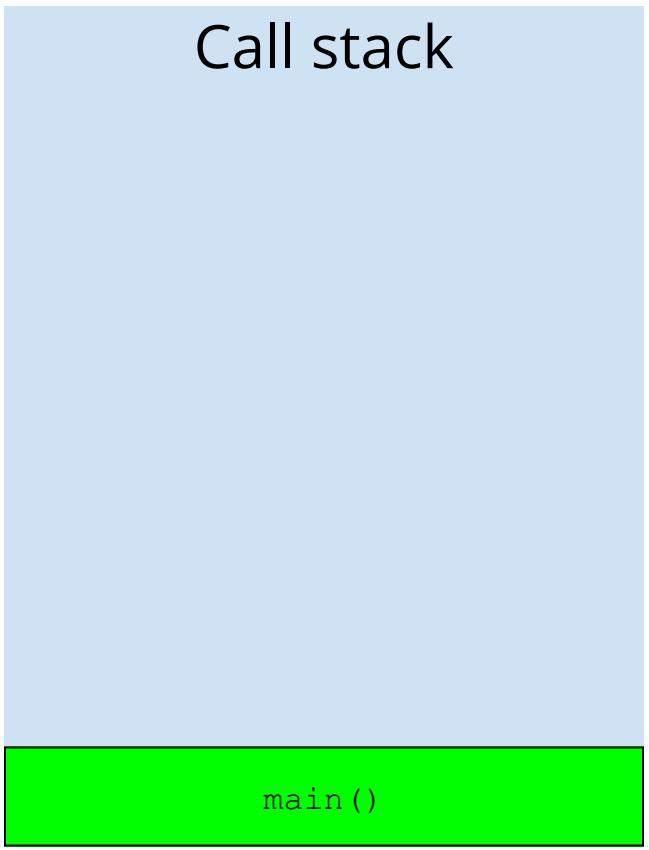
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



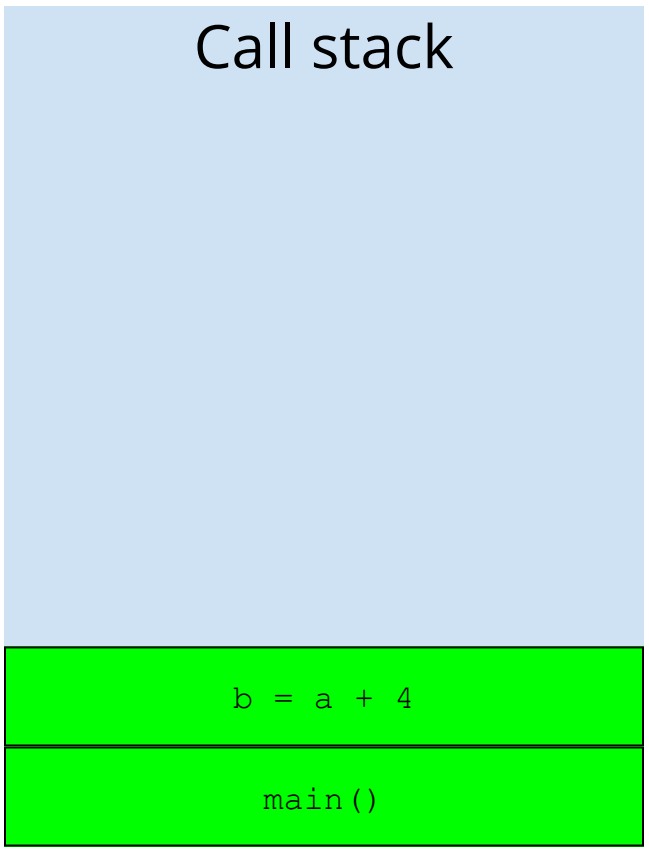
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



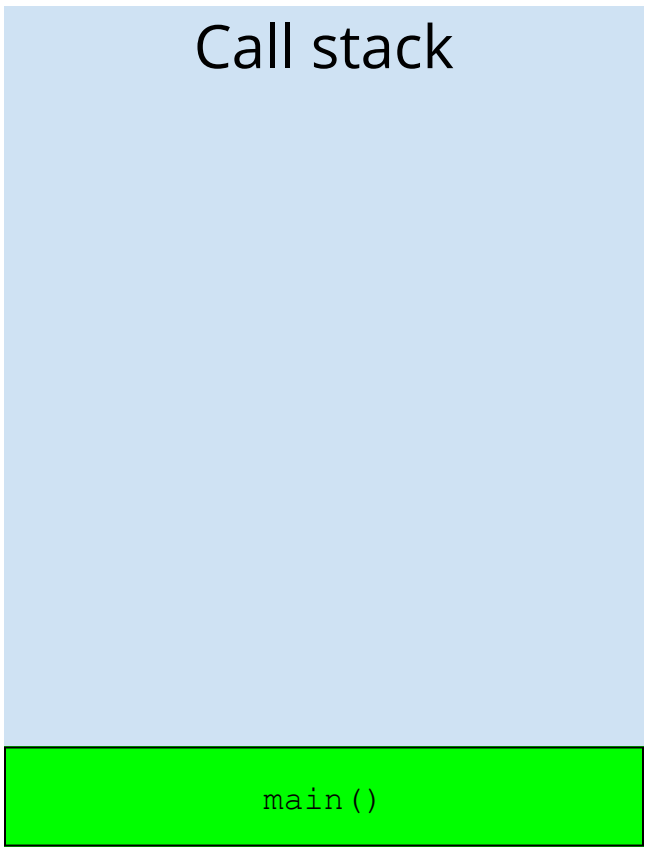
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



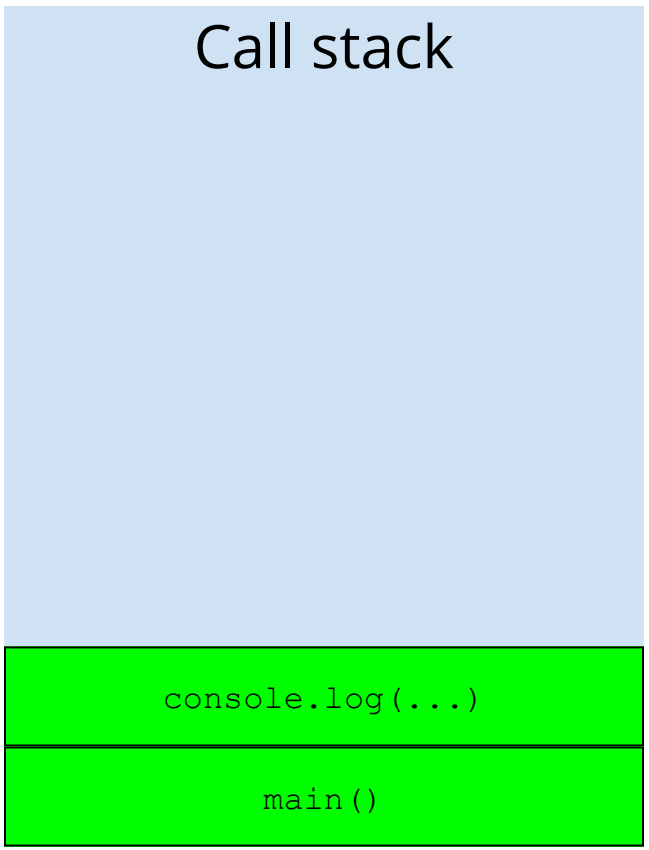
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



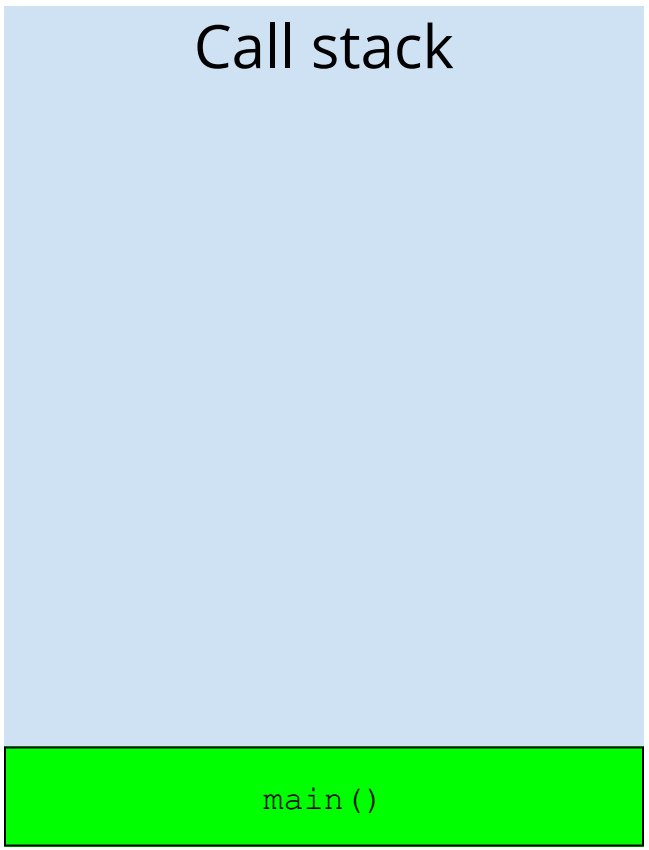
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```



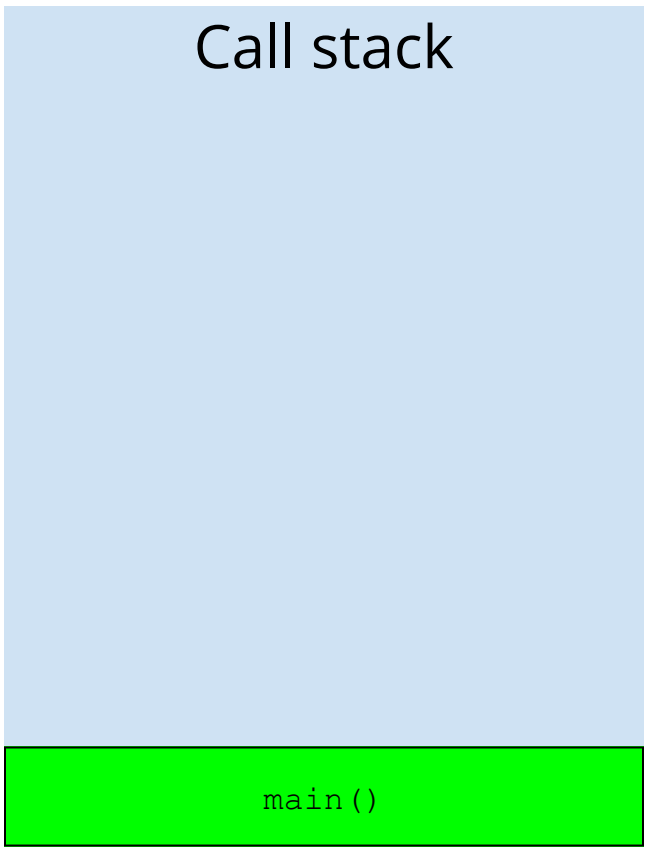
```
1 const a = 3;
2
3 const b = a + 4;
4
5 console.log('b = ', b);
```

Call stack

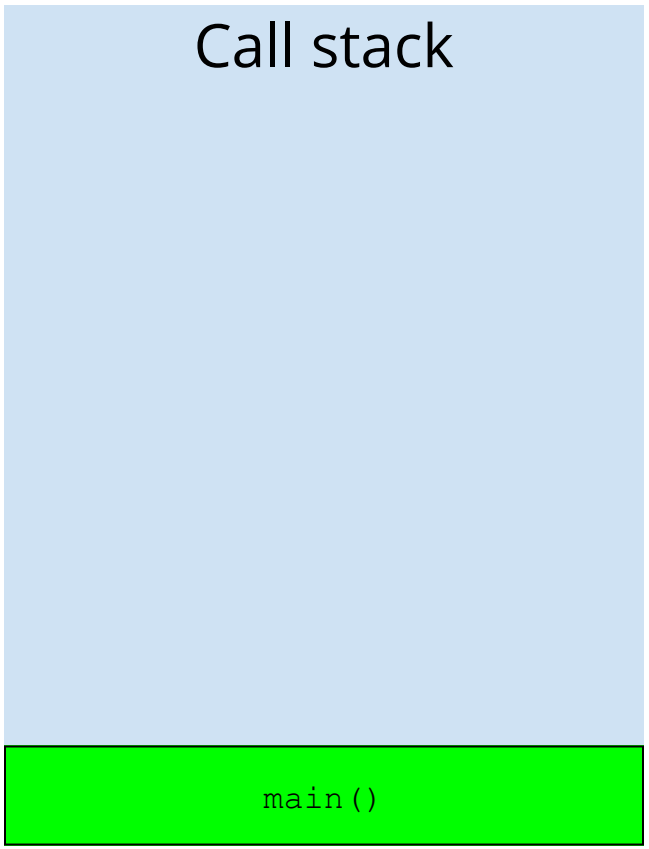
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```

Call stack

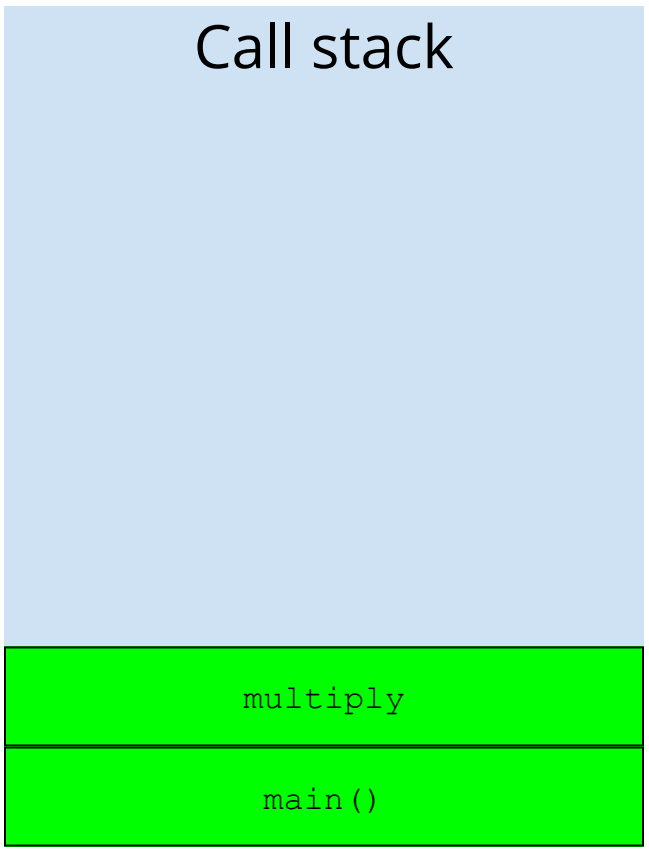
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



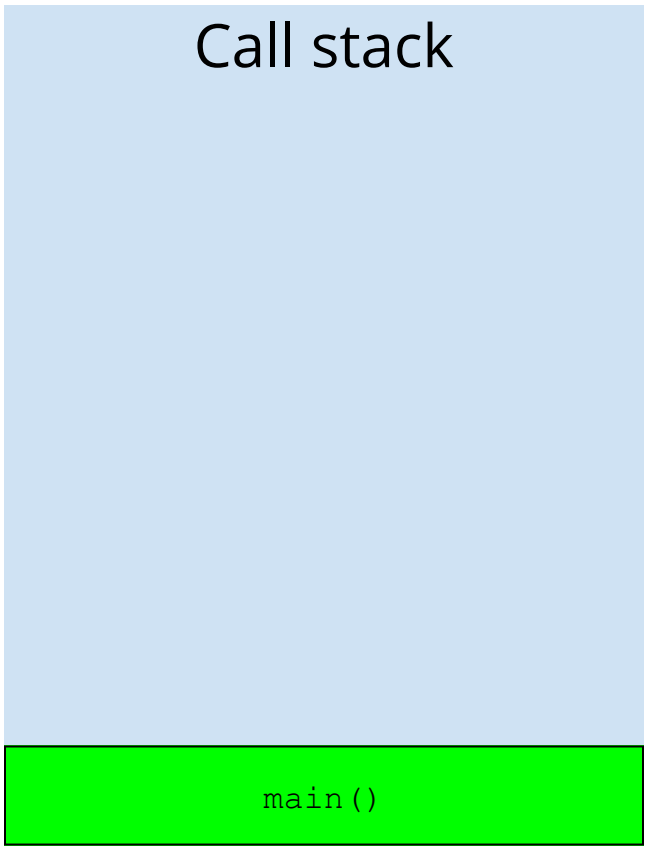
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



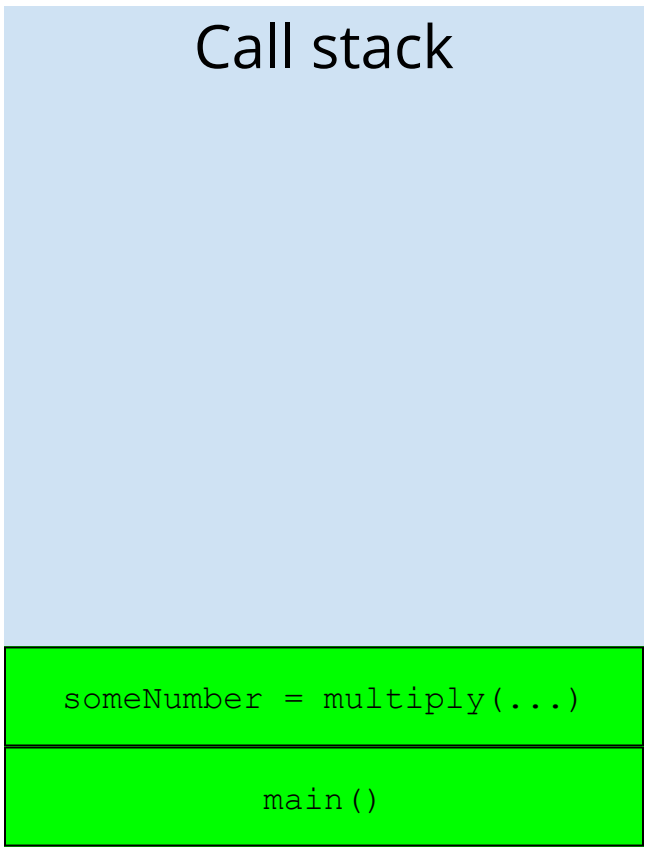
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



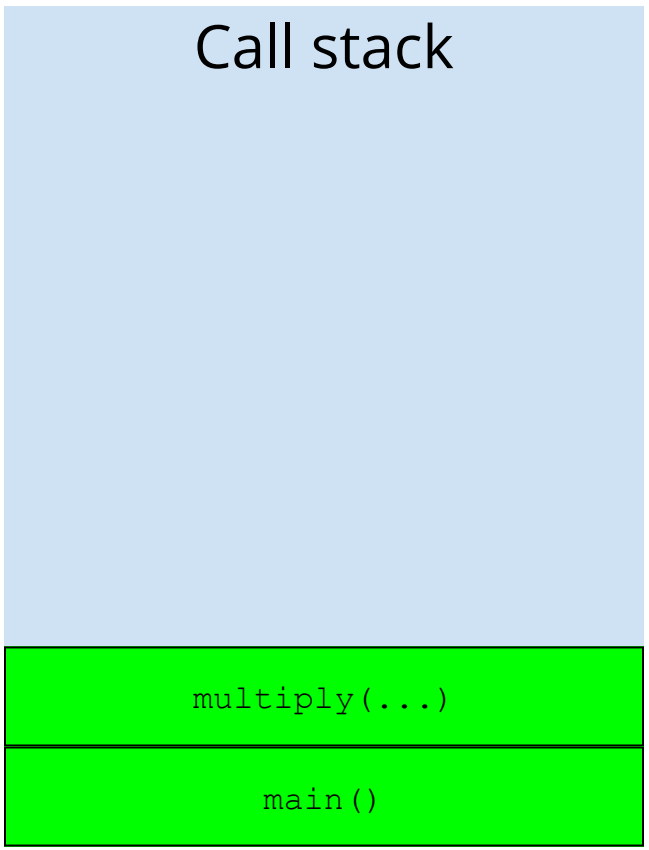
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



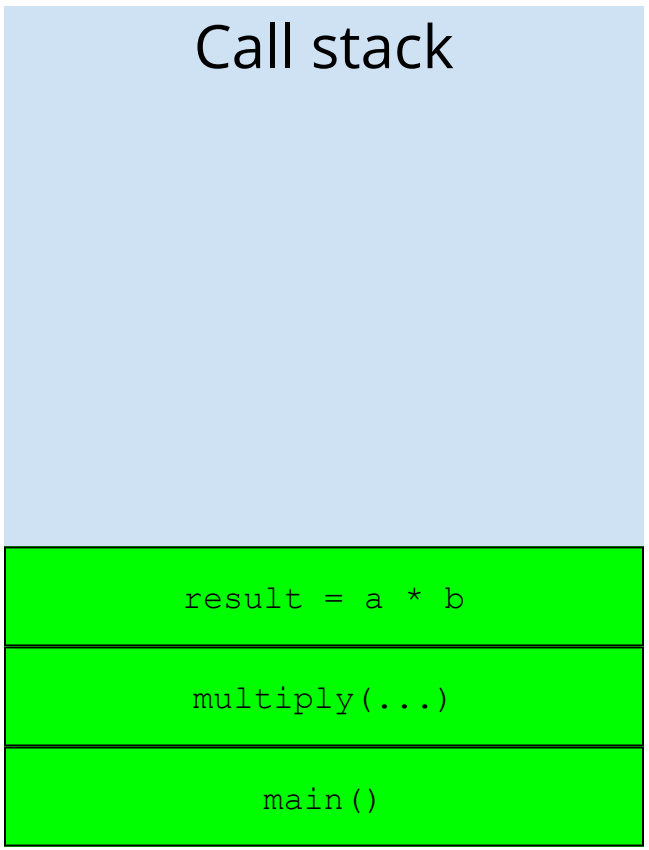
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



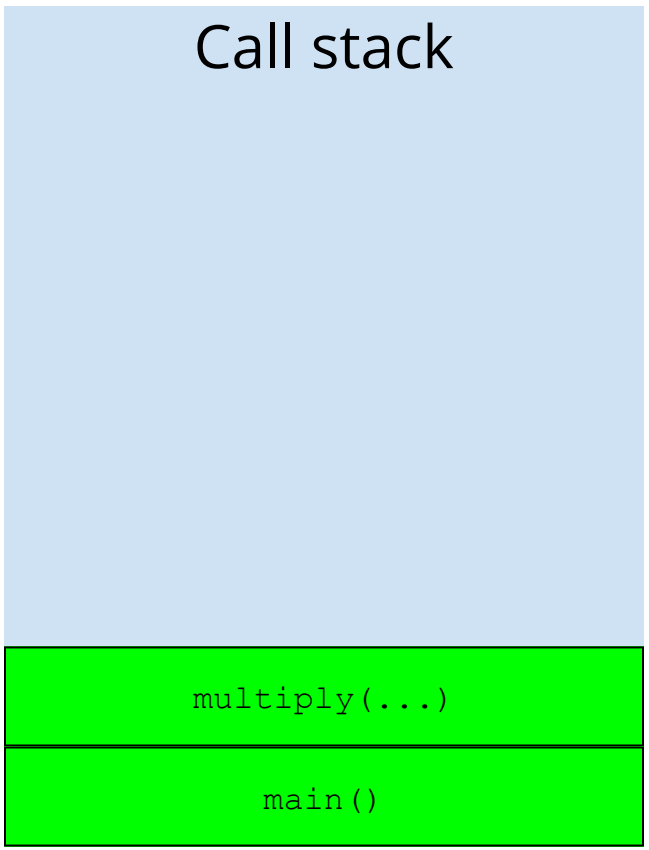
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



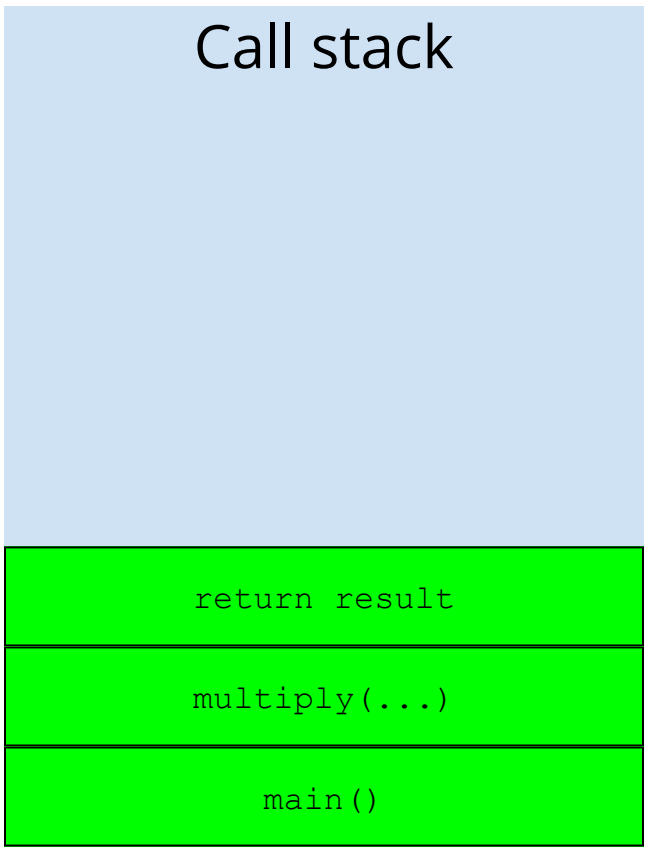
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



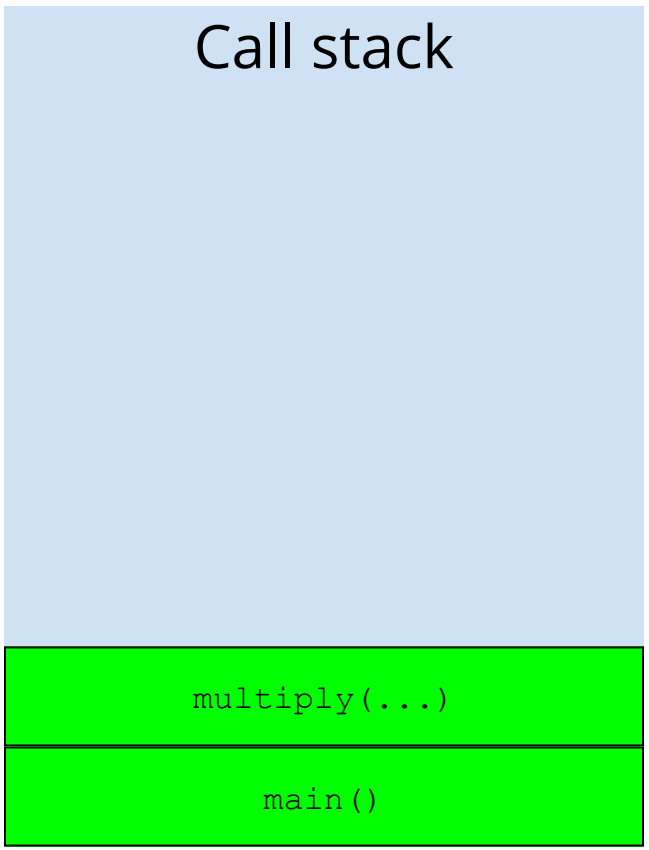
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



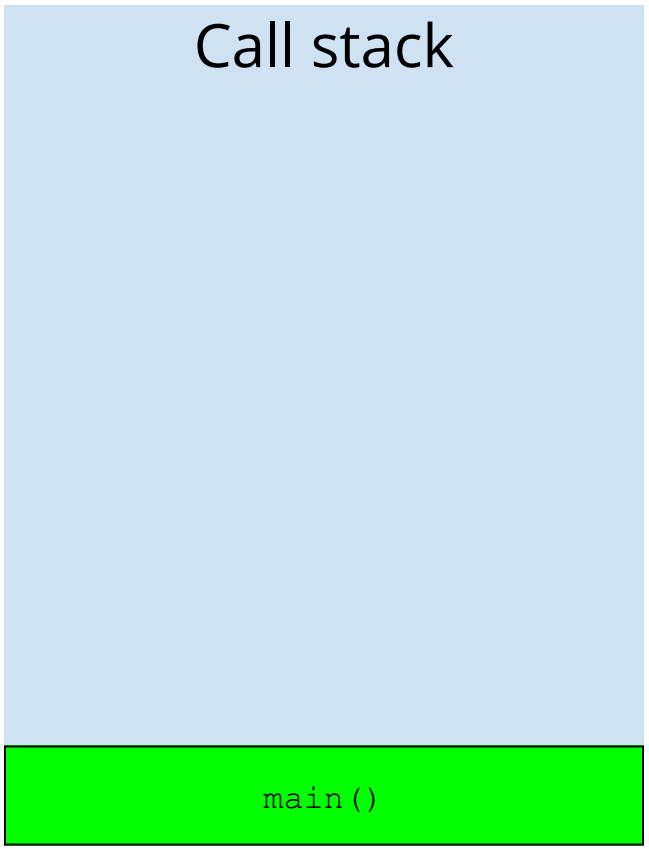
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



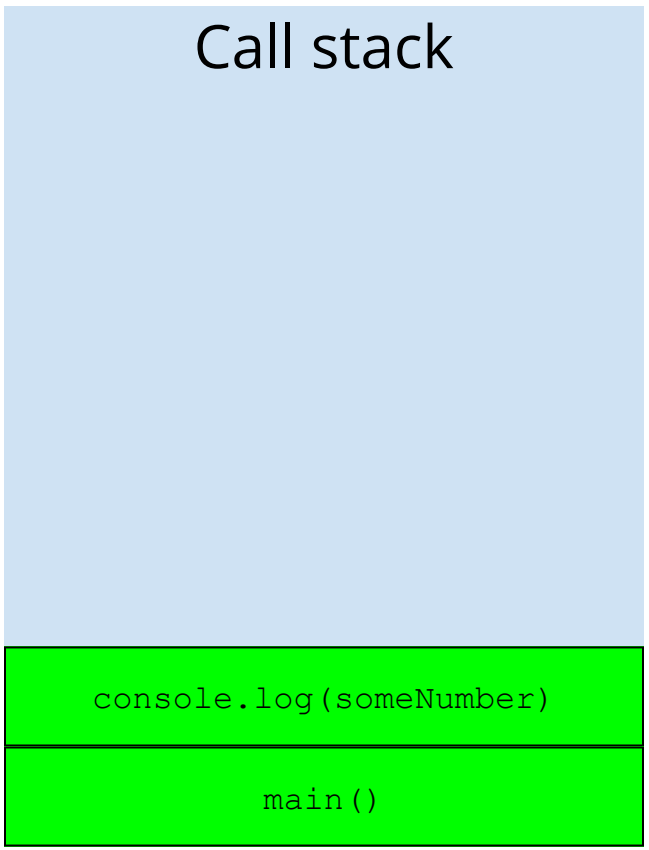
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



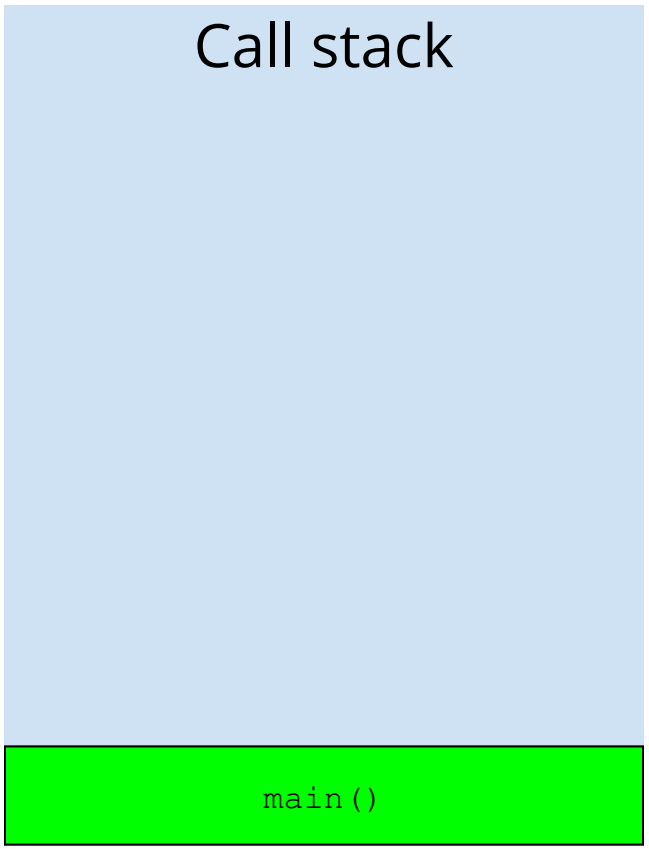
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```



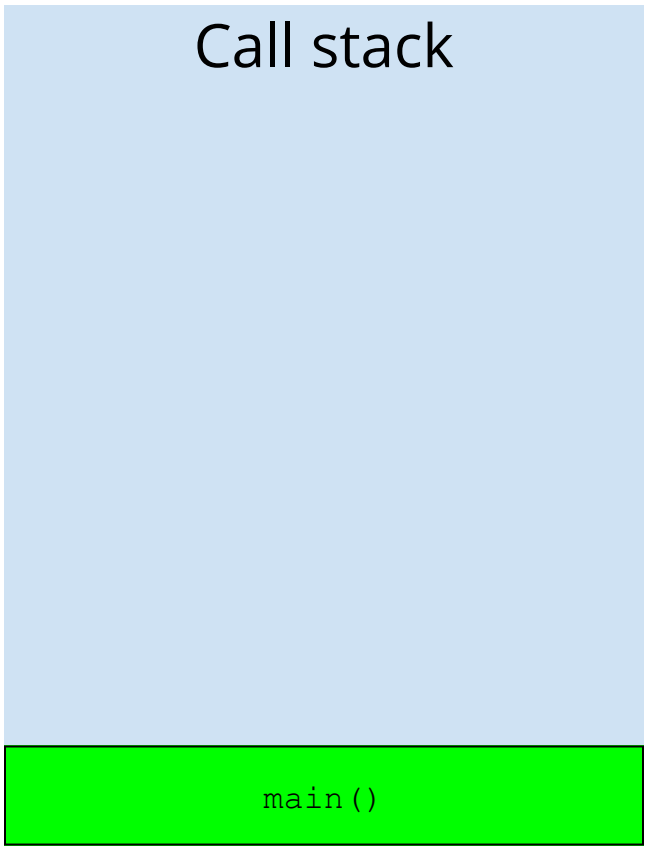
```
1 function multiply(a, b) {
2   const result = a * b;
3   return result;
4 }
5
6 const someNumber = multiply(5, 4);
7
8 console.log(someNumber);
```

Call stack

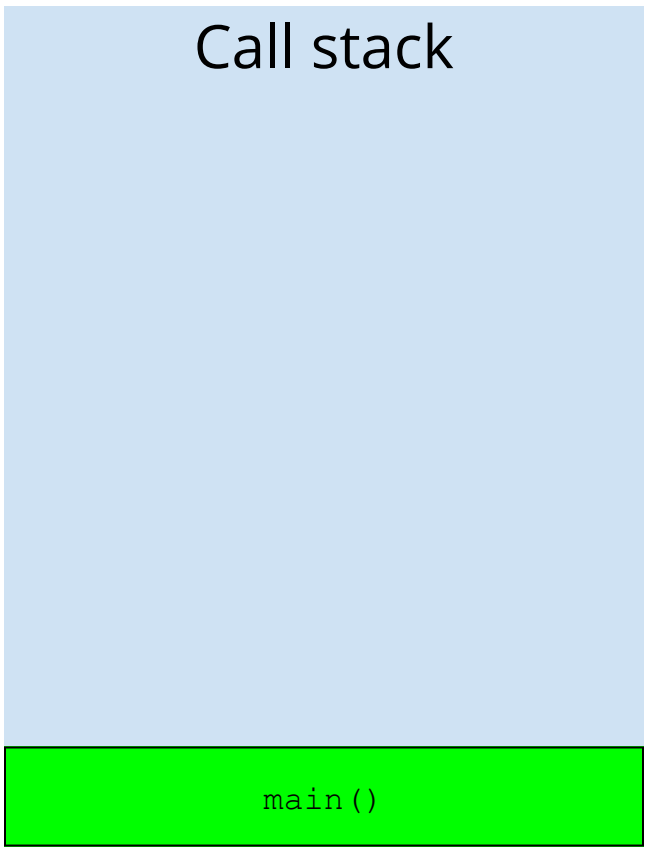
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

Call stack

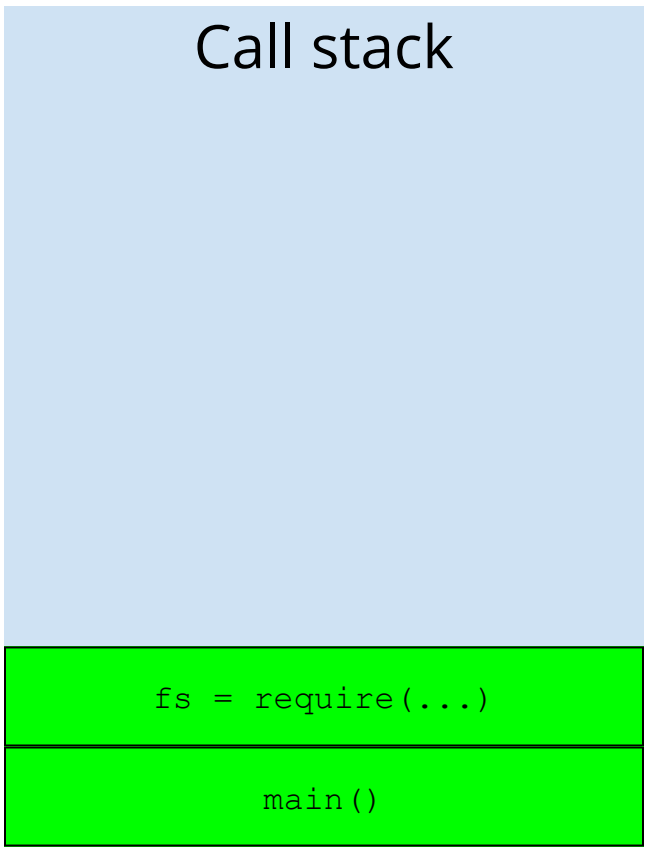
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



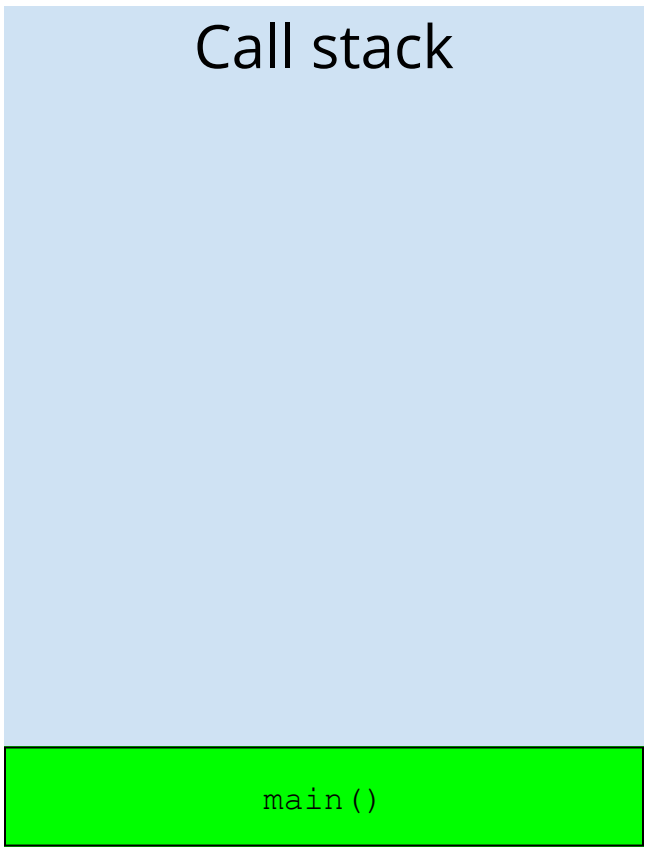
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



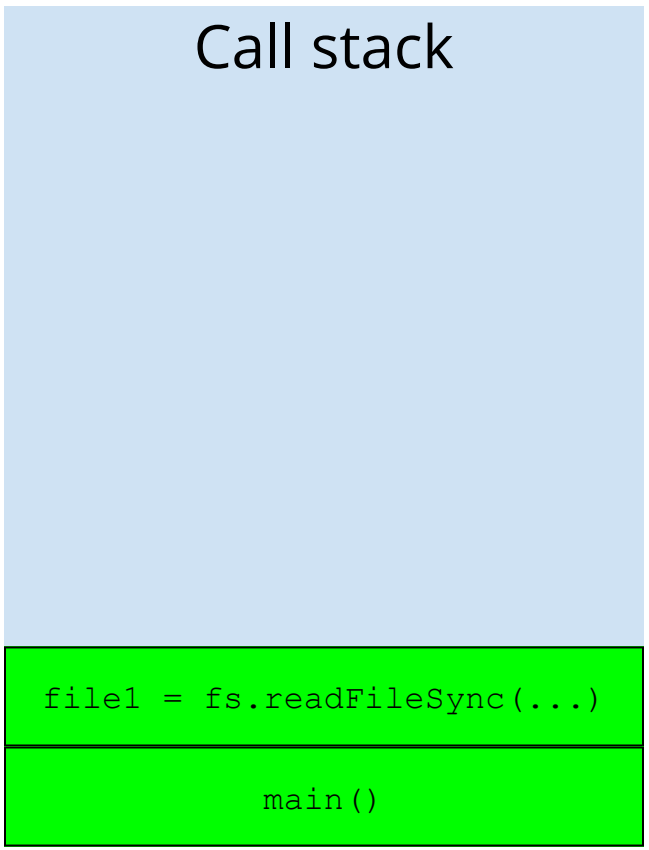
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



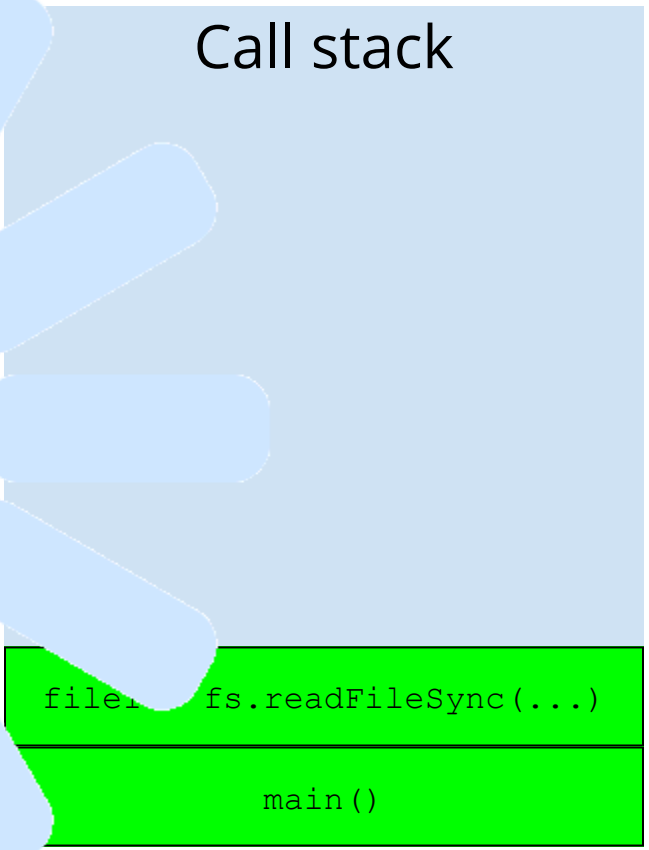
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



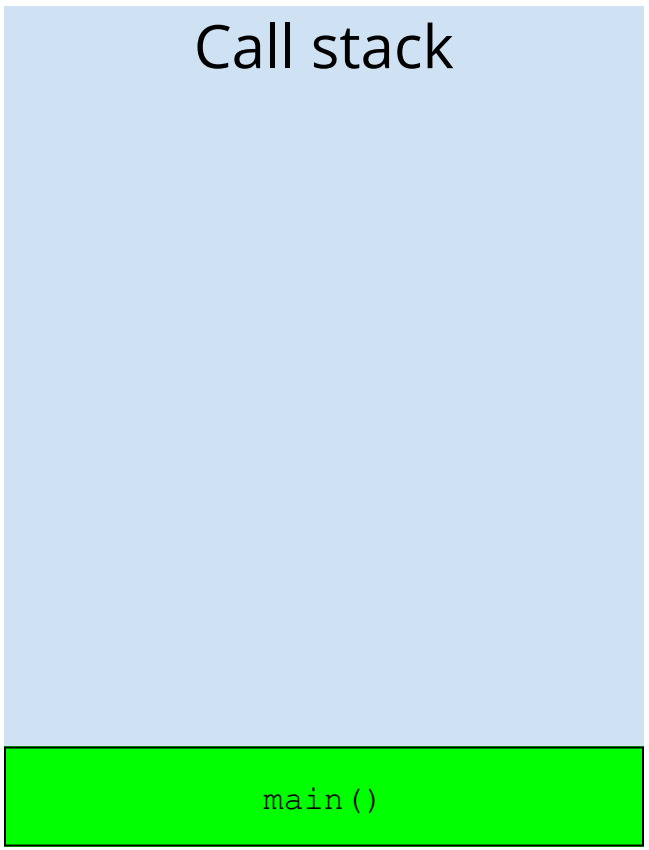
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



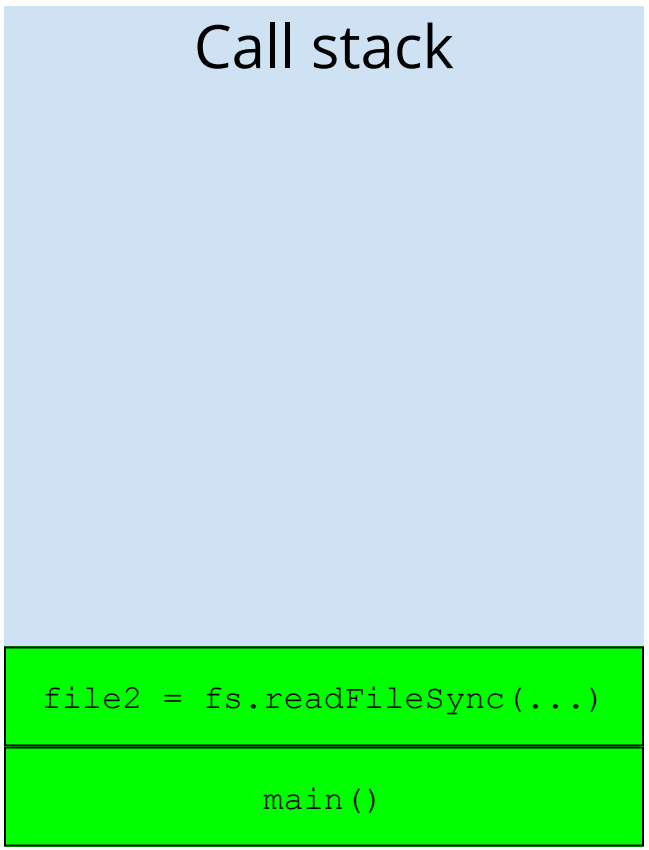
```
1  const fs = require('fs');
2
3
4  const file1 = fs.r                c('  e1.txt');
5
6  const file2 = fs.readFil          'file2.txt');
7
8  const file3 = f                   c('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3
```

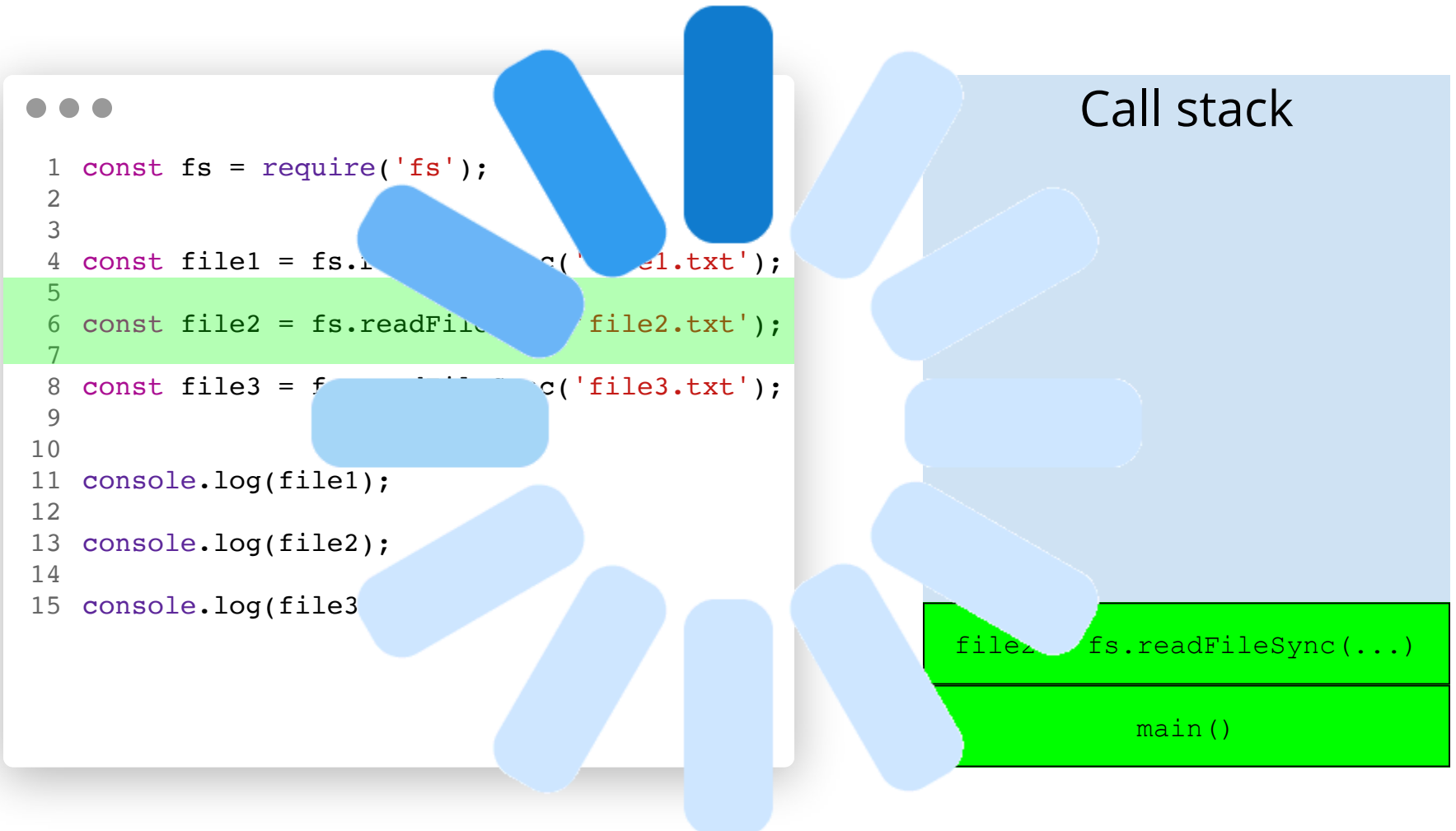


```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

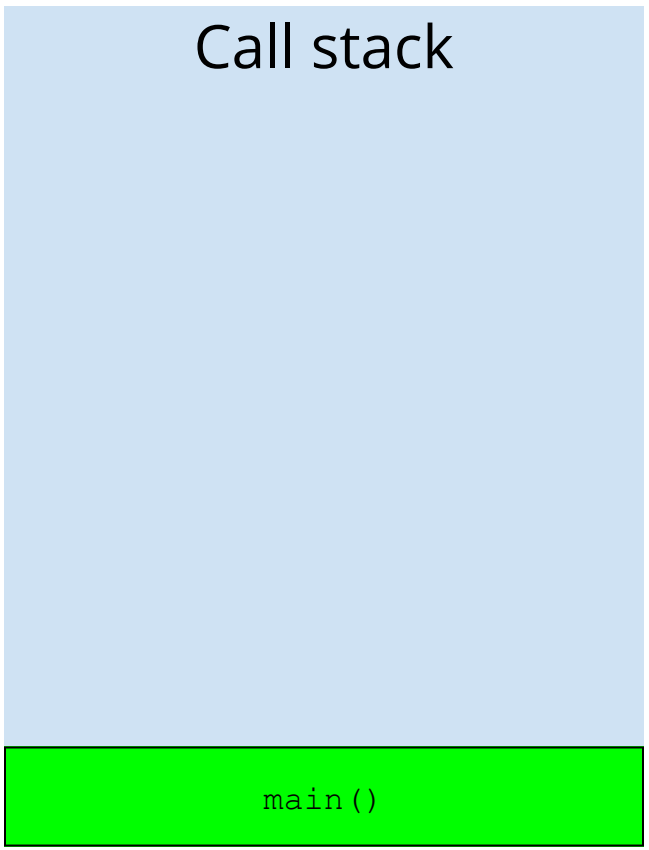


```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

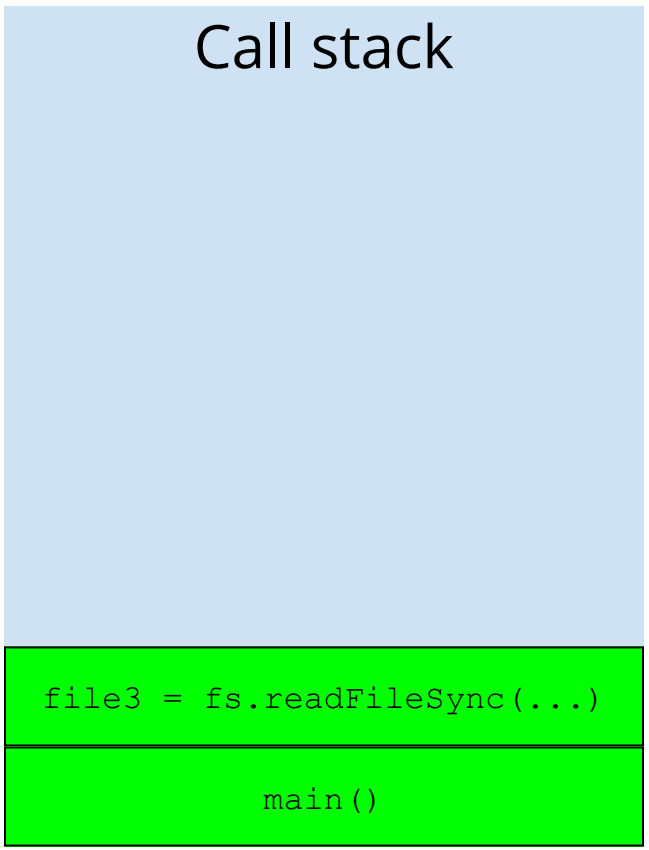


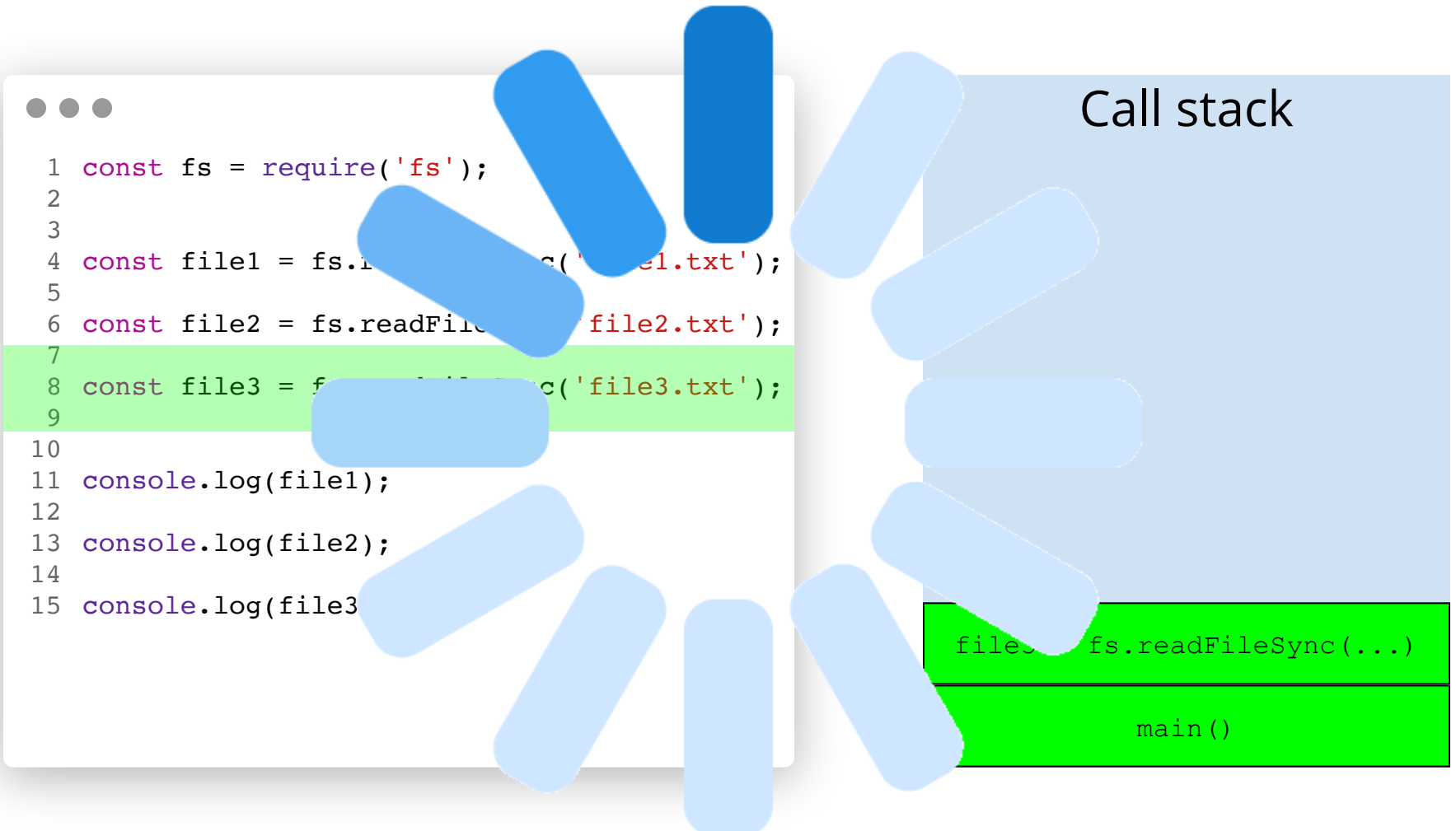

1 const fs = require('fs');
2
3
4 const file1 = fs.
1.txt');
5
6 const file2 = fs.readFi
file2.txt');
7
8 const file3 = f
('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3
Call stack
file
fs.readFileSync(...)
main()

```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

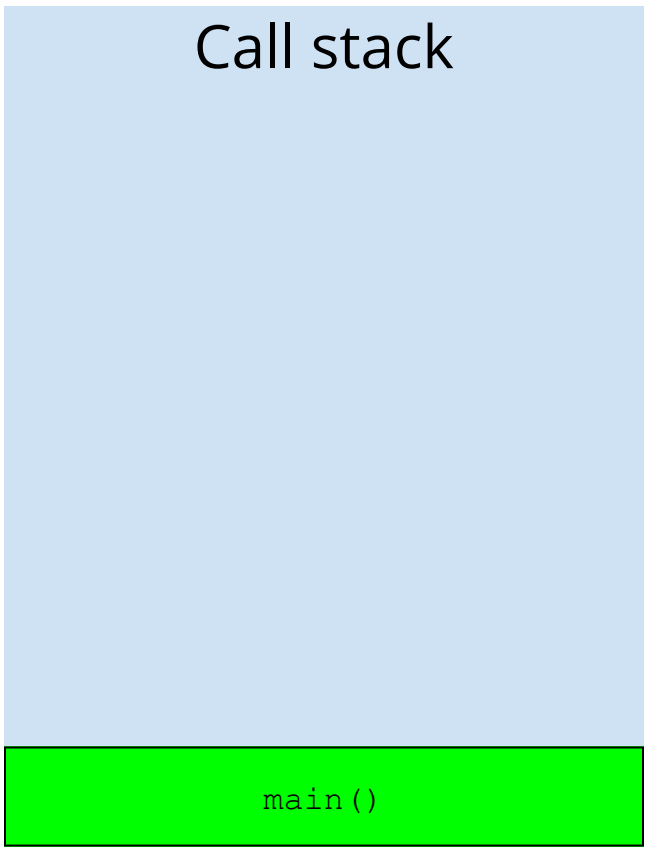


```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

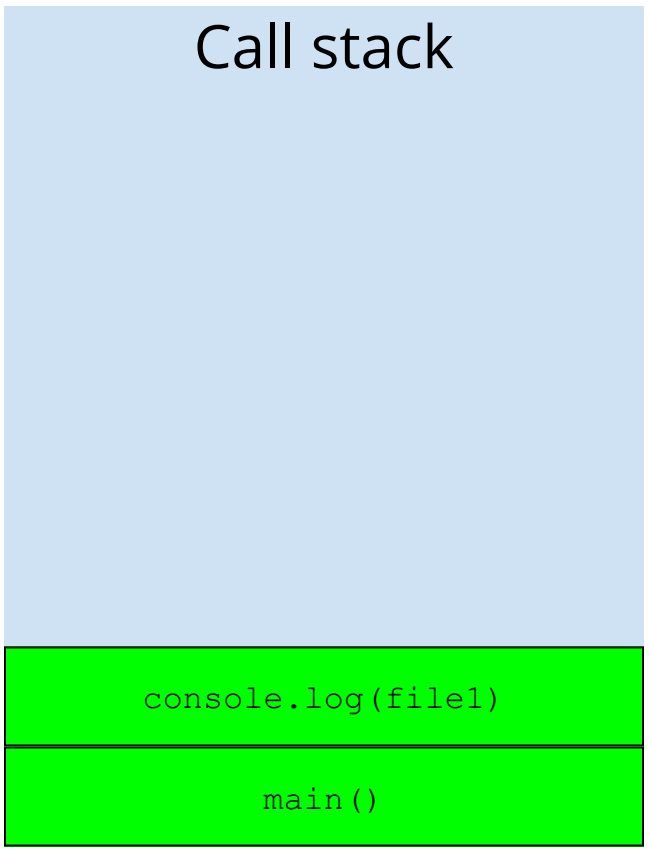



1 const fs = require('fs');
2
3
4 const file1 = fs.
1.txt');
5
6 const file2 = fs.readFi
file2.txt');
7
8 const file3 = f
('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3
Call stack
file
fs.readFileSync(...)
main()

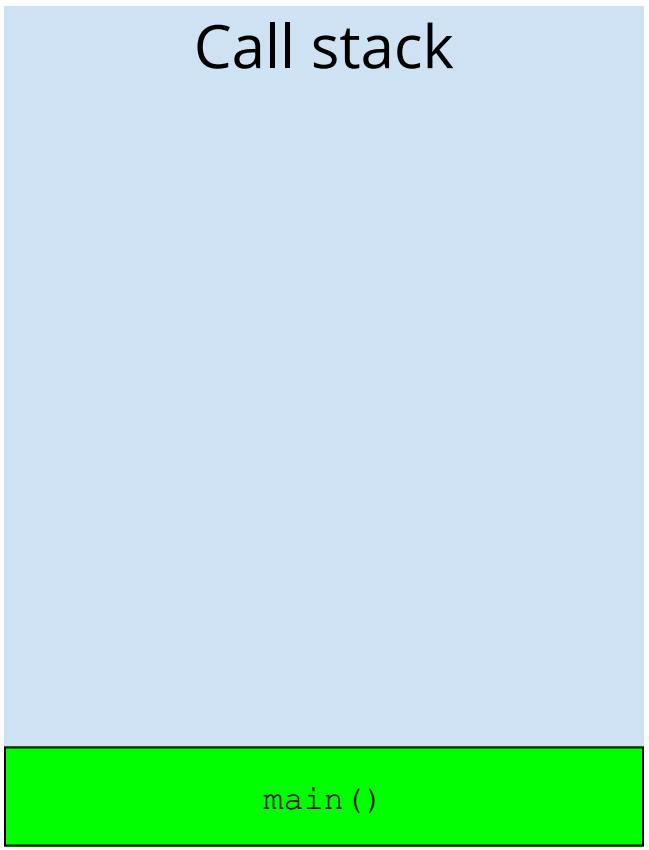
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



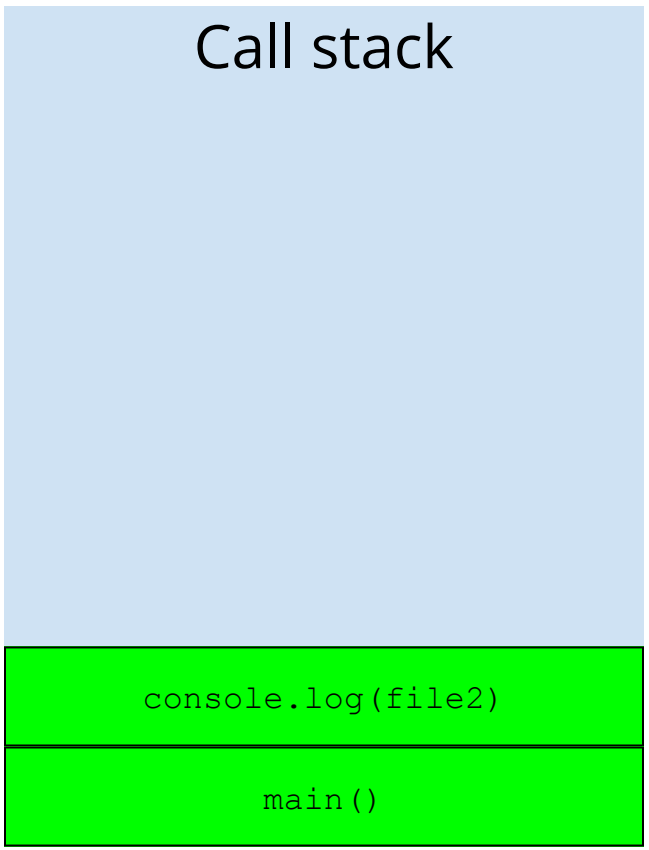
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



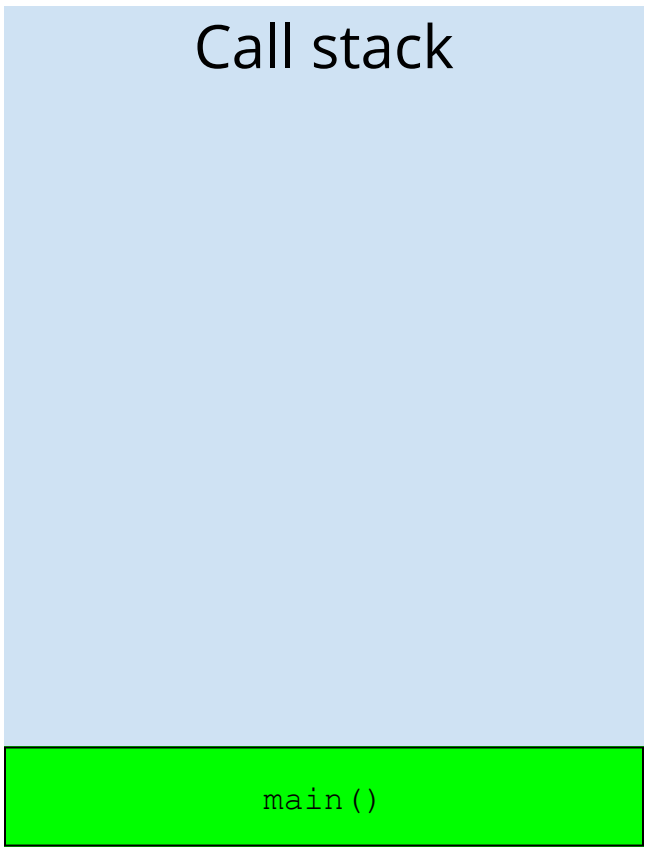
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



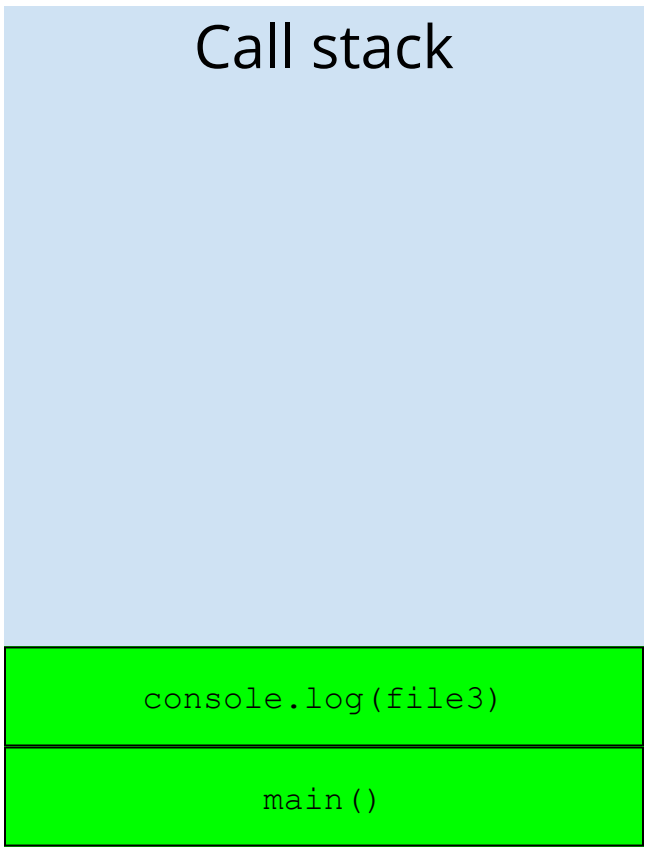
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



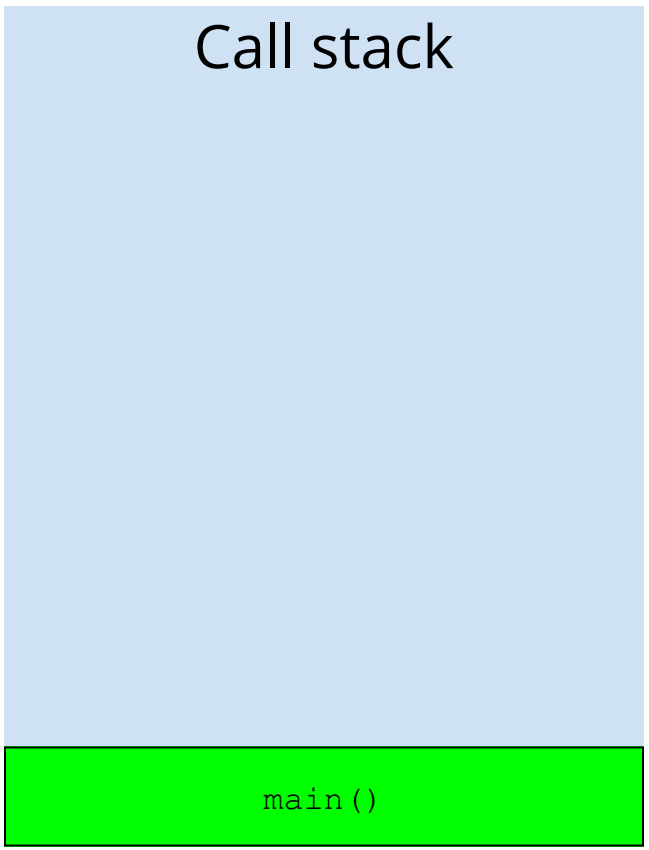
```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```



```
1  const fs = require('fs');
2
3
4  const file1 = fs.readFileSync('file1.txt');
5
6  const file2 = fs.readFileSync('file2.txt');
7
8  const file3 = fs.readFileSync('file3.txt');
9
10
11 console.log(file1);
12
13 console.log(file2);
14
15 console.log(file3);
```

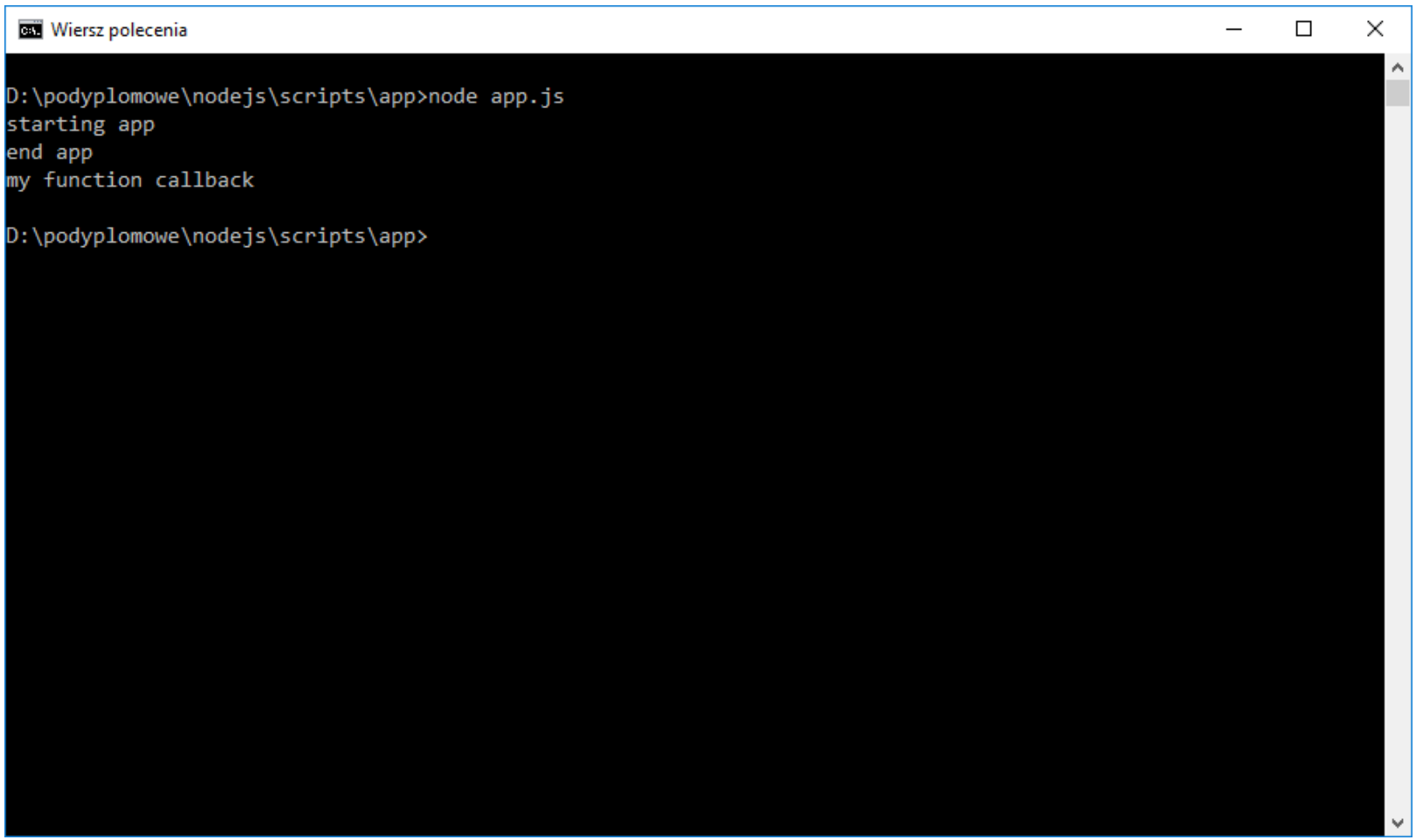
Call stack

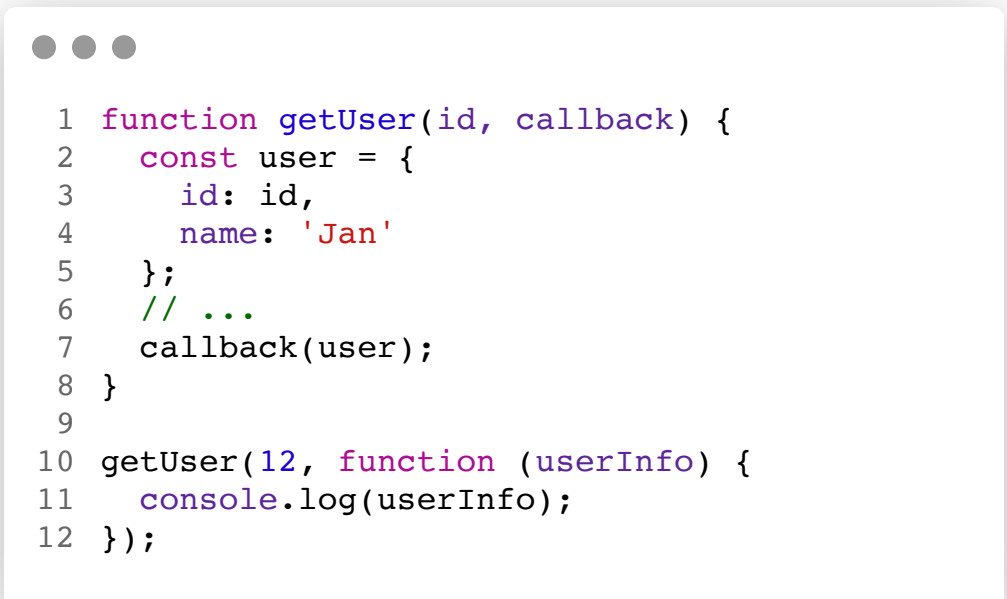
ASYNC

CALLBACK

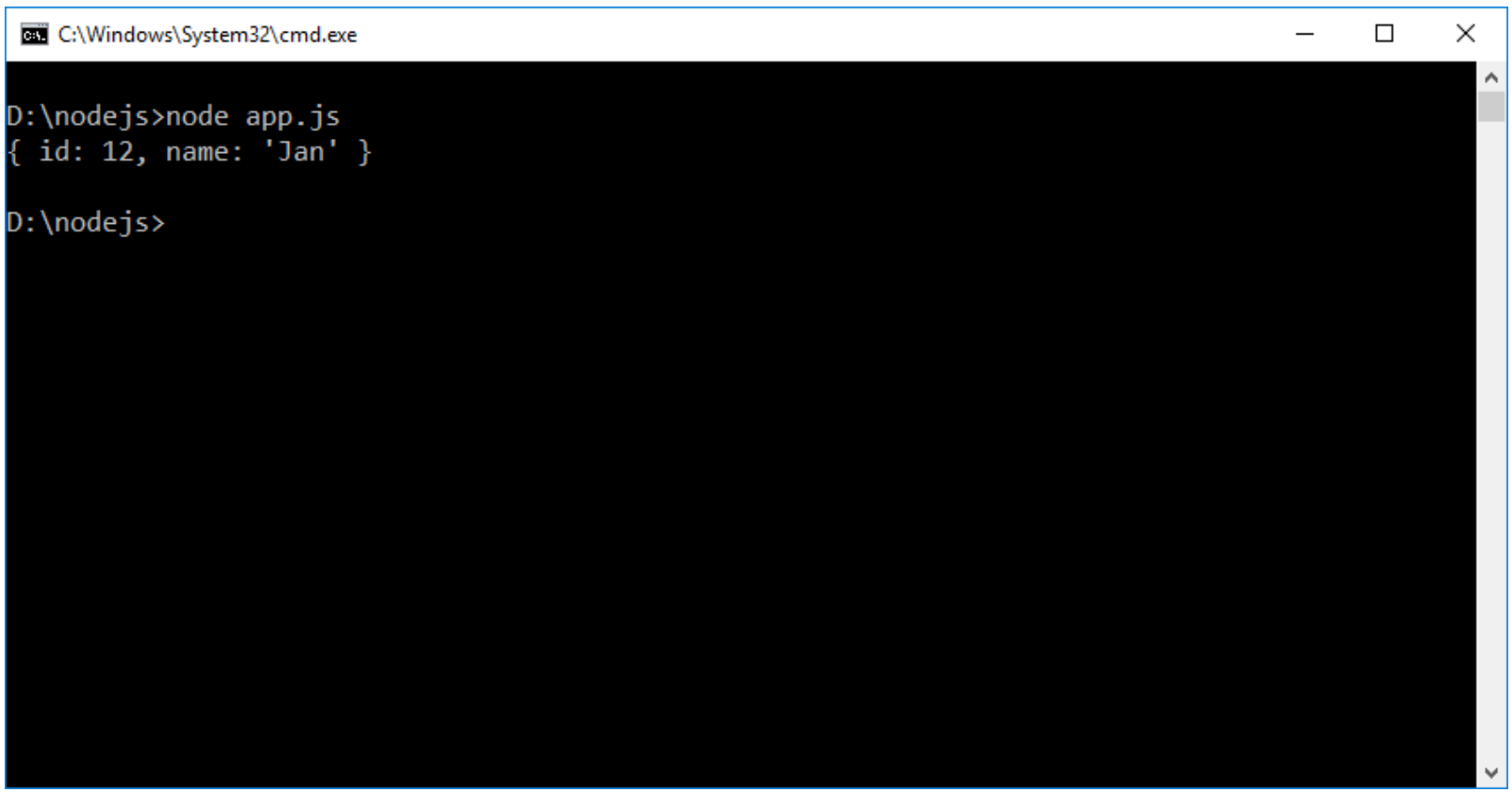
Funkcja która zostanie wykonana po określonej czynności

```
1 console.log('starting app');
2
3 setTimeout(function () {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```


Wiersz polecenia
D:\podyplomowe\nodejs\scripts\app>node app.js
starting app
end app
my function callback
D:\podyplomowe\nodejs\scripts\app>



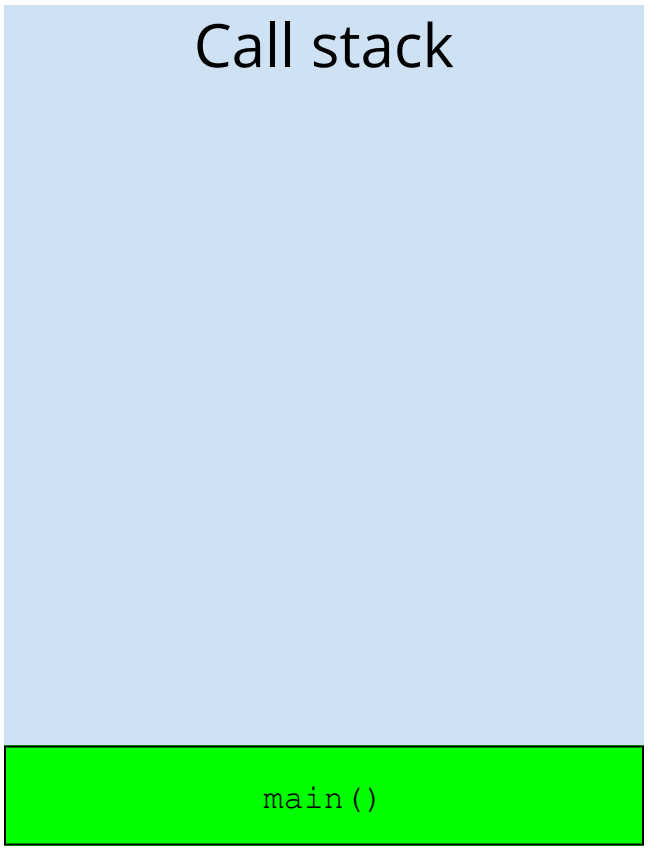
```
function getUser(id, callback) {
  const user = {
    id: id,
    name: 'Jan'
  };
  // ...
  callback(user);
}

getUser(12, function (userInfo) {
  console.log(userInfo);
});
```


C:\Windows\System32\cmd.exe
D:\nodejs>node app.js
{ id: 12, name: 'Jan' }
D:\nodejs>

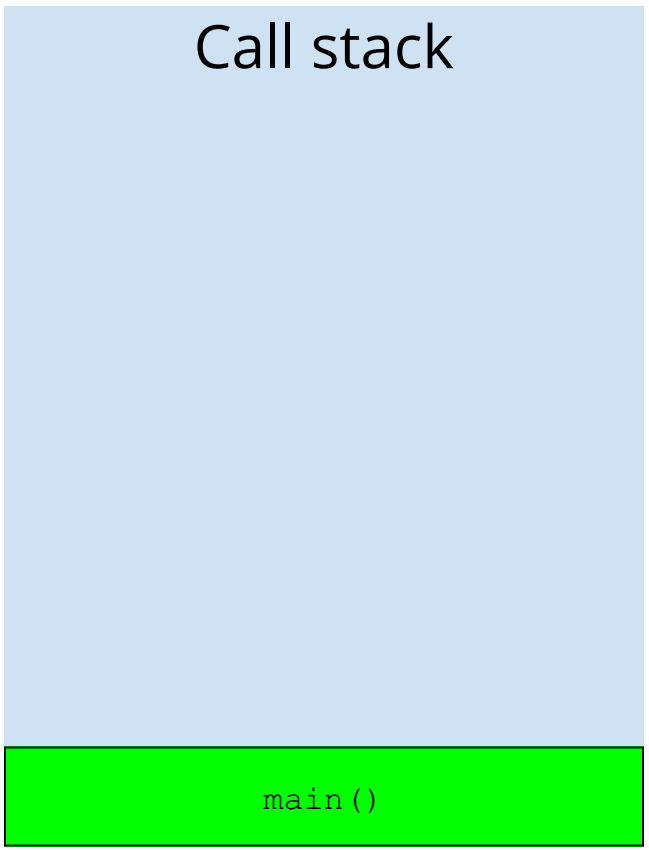
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```

Call stack

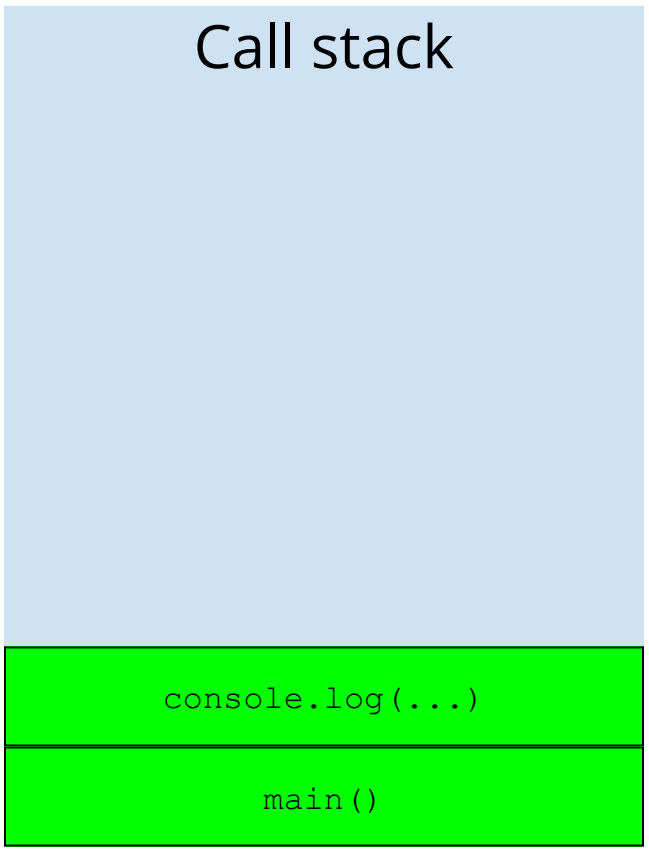
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



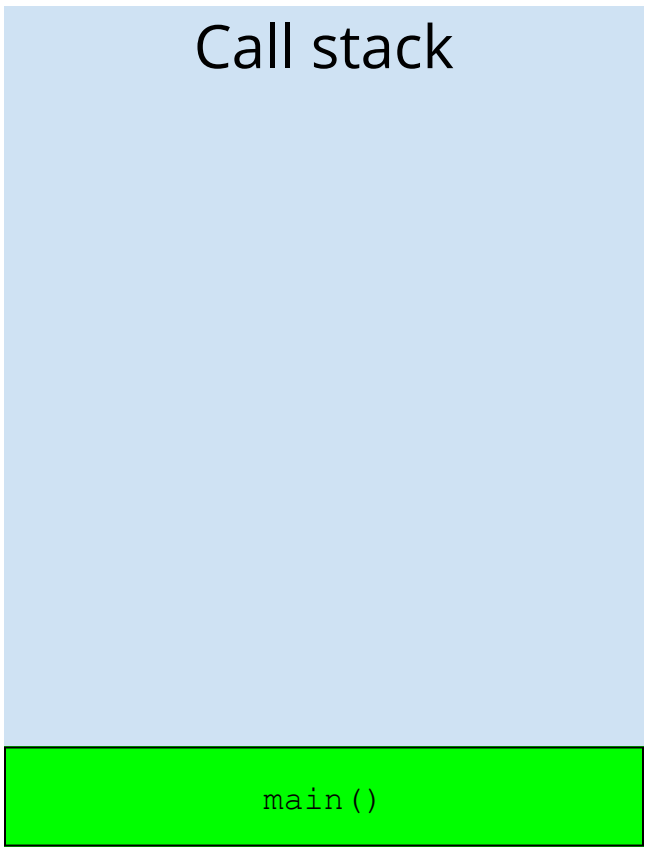
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



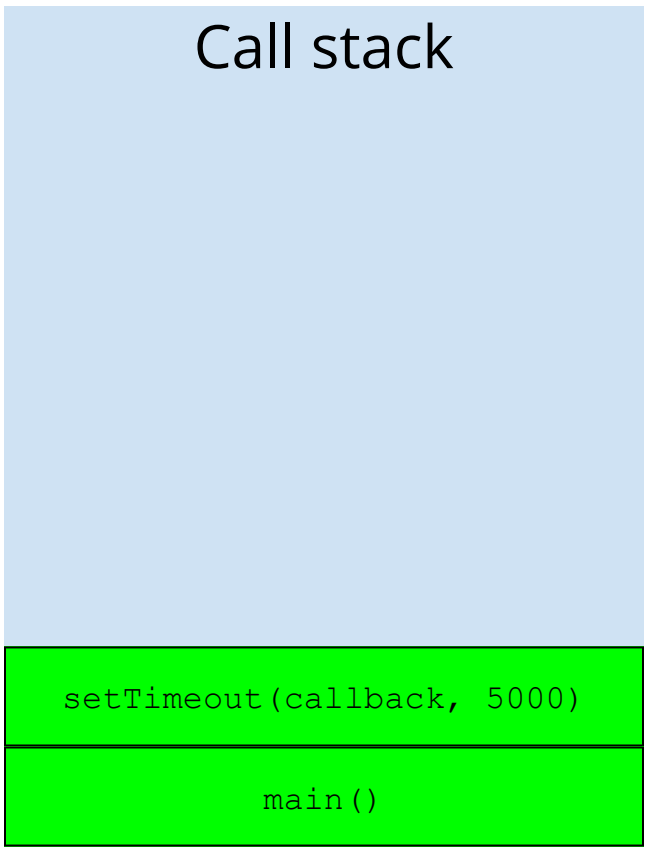
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



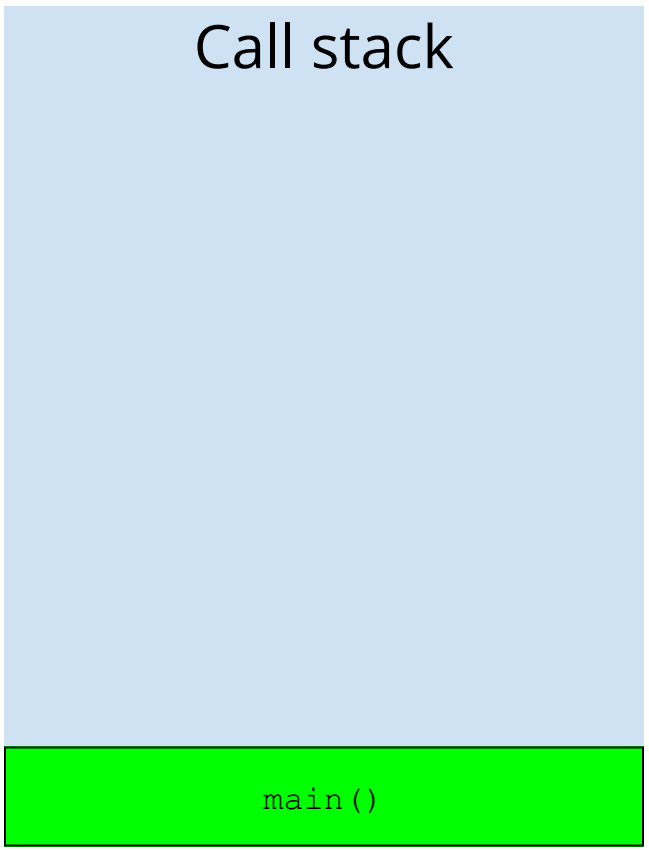
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



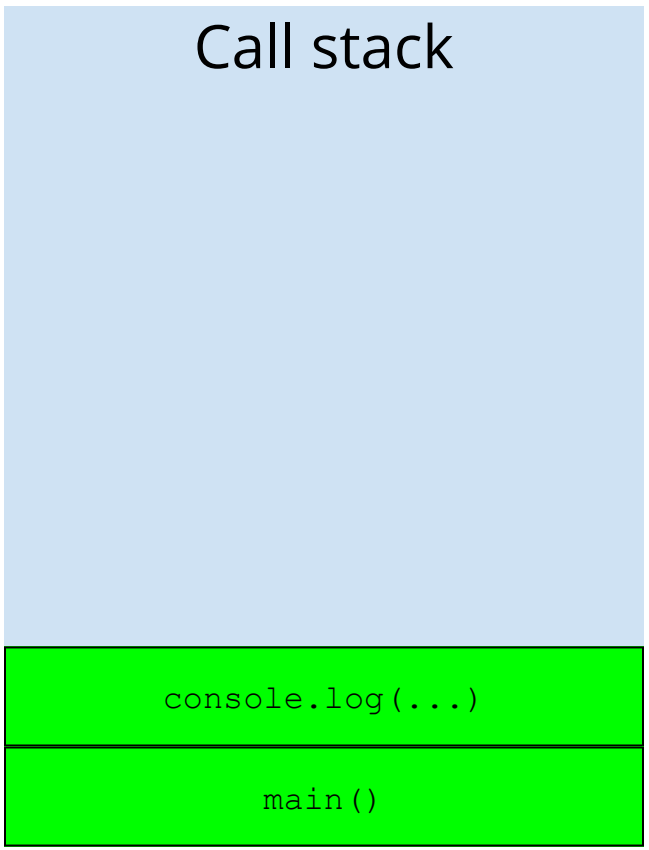
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



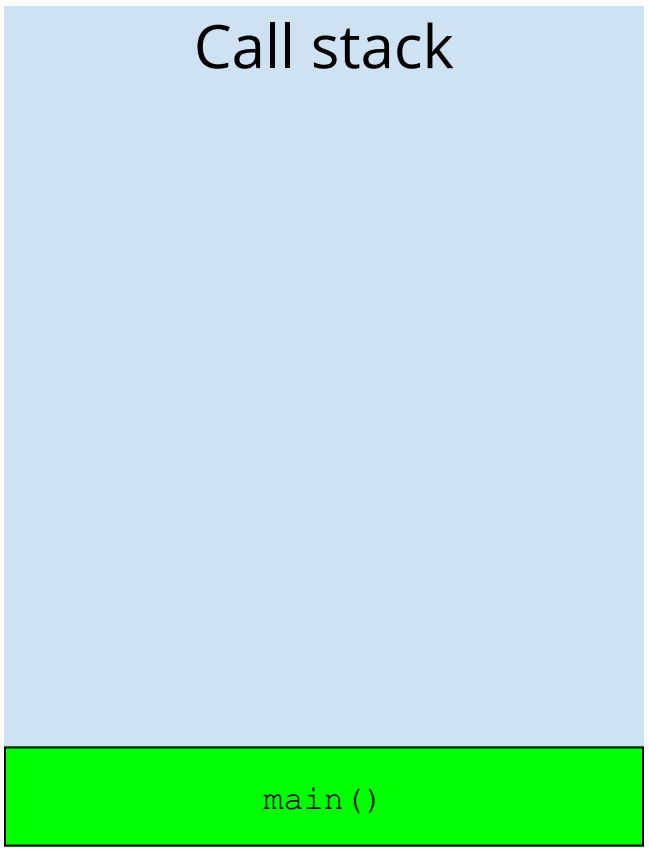
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



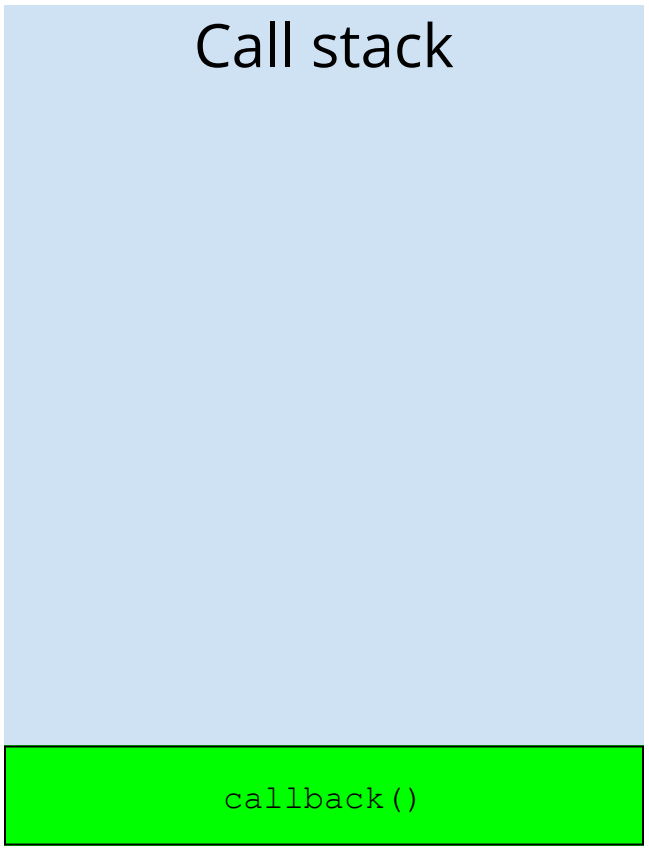
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



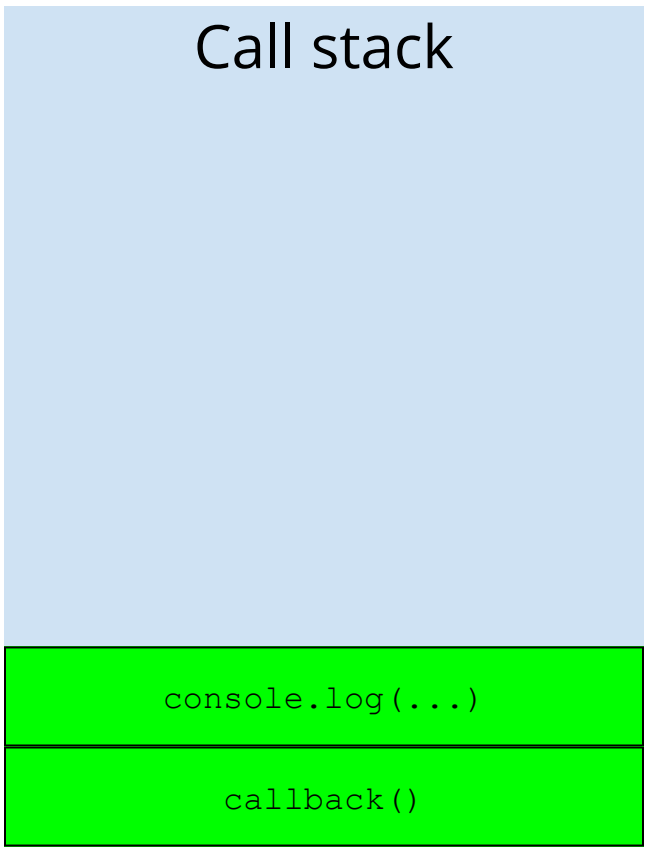
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```

Call stack

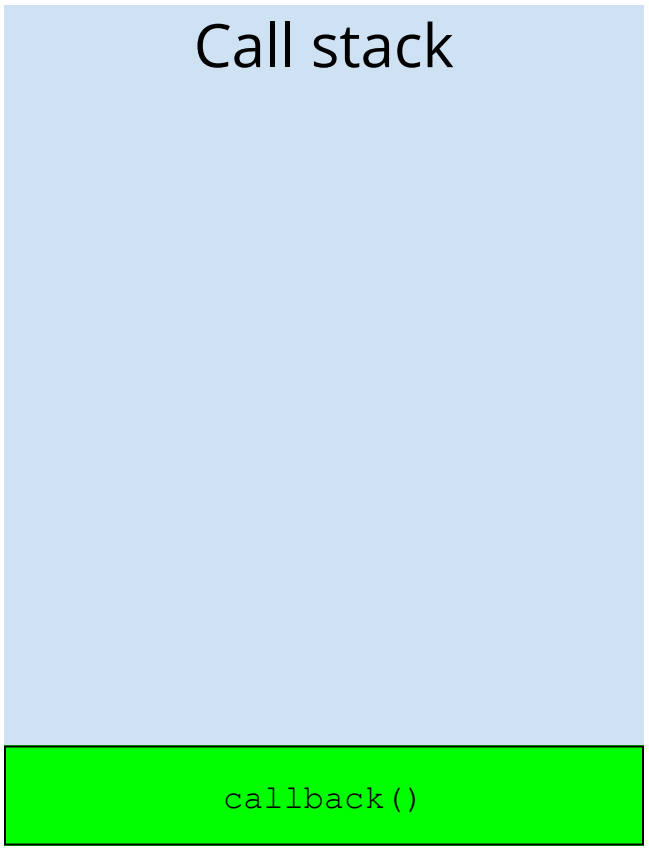
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



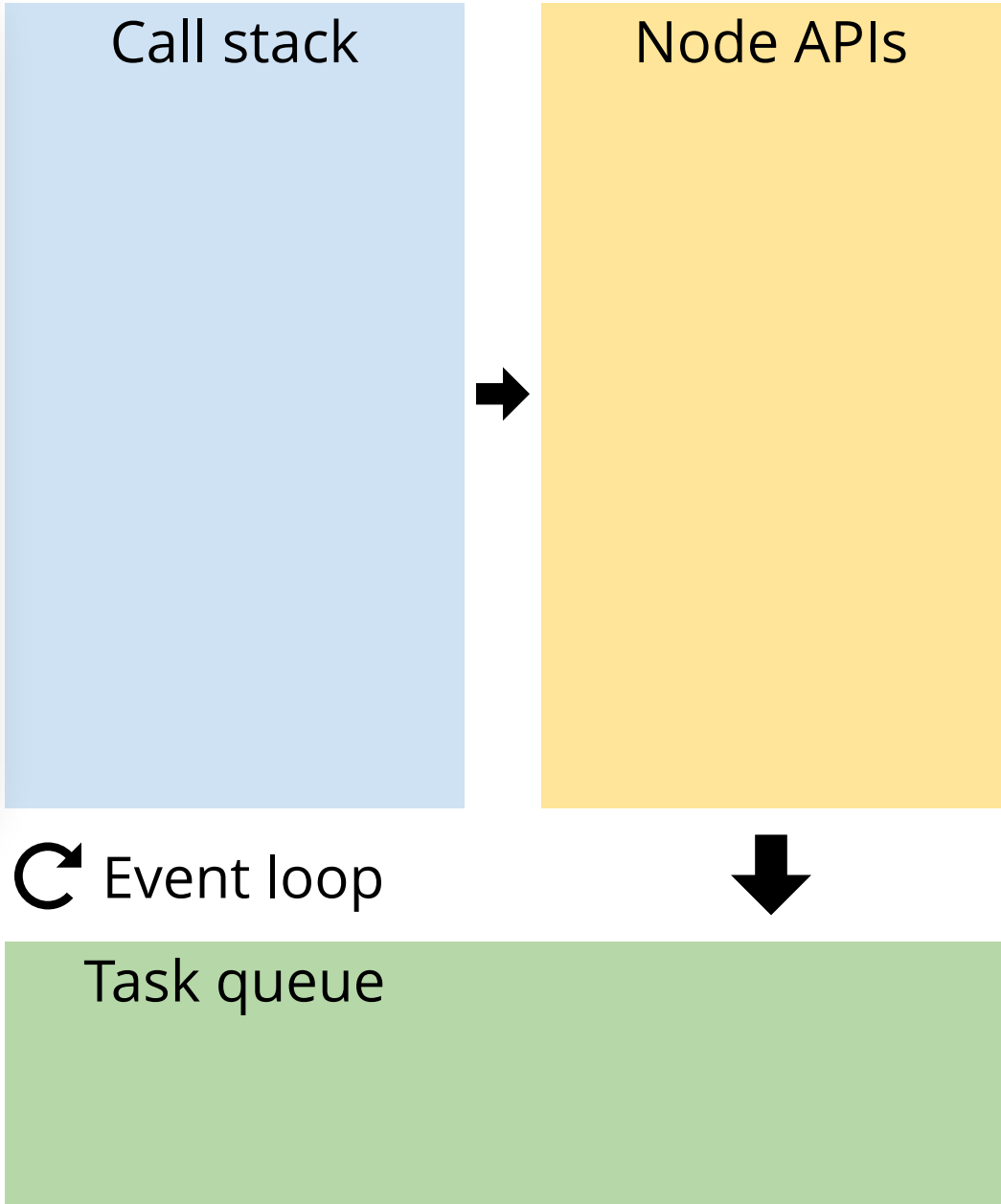
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



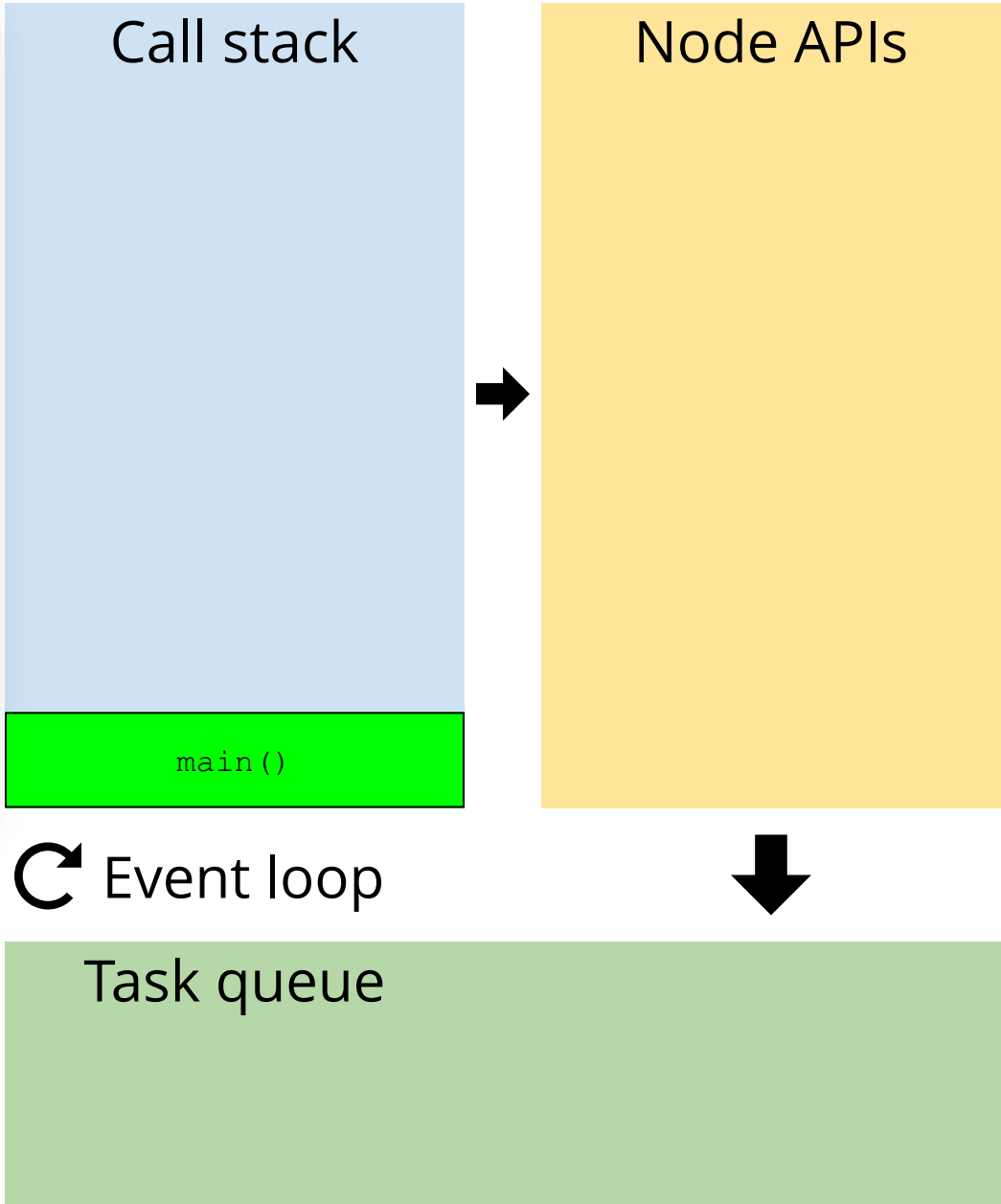
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```

Call stack

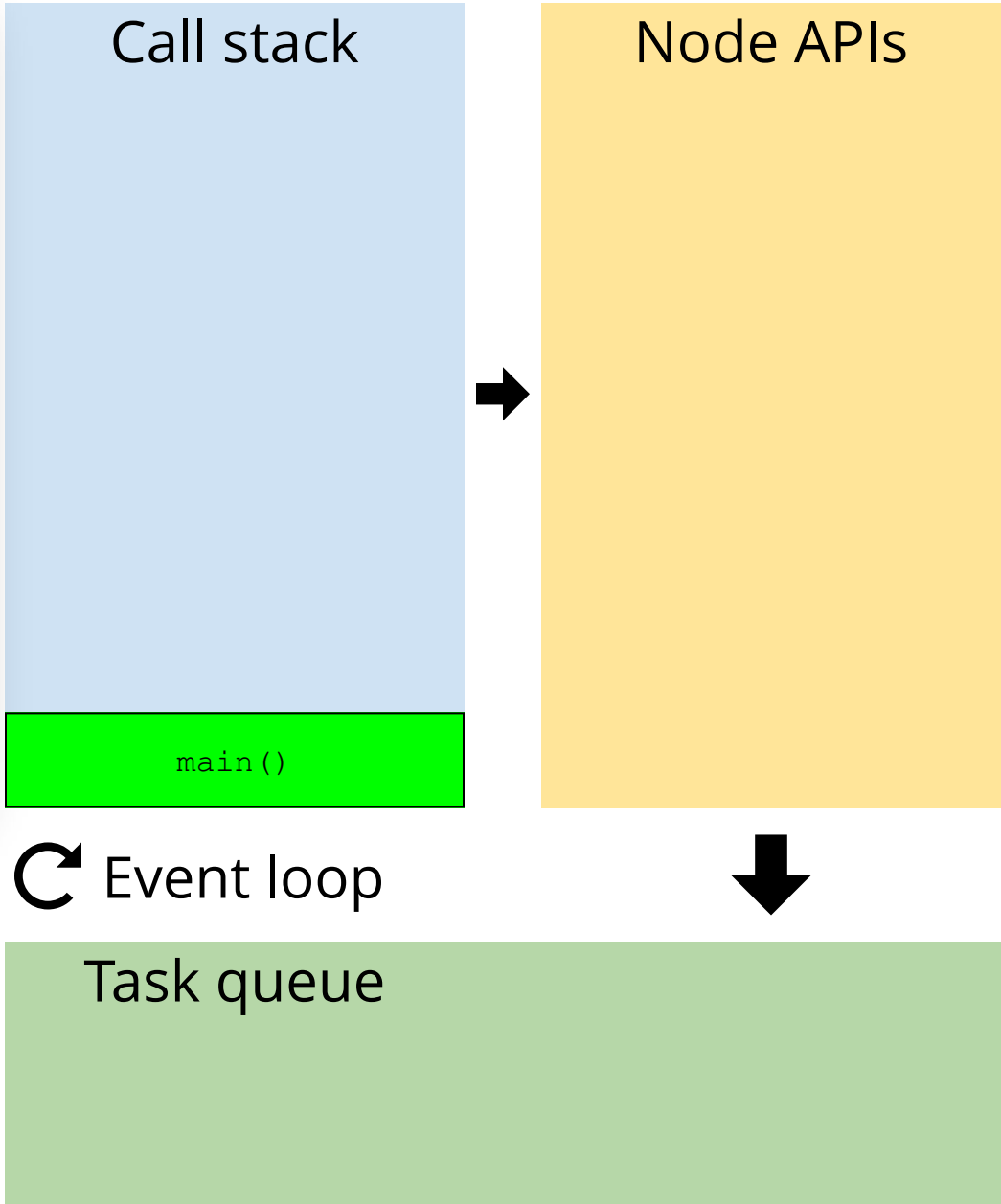
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



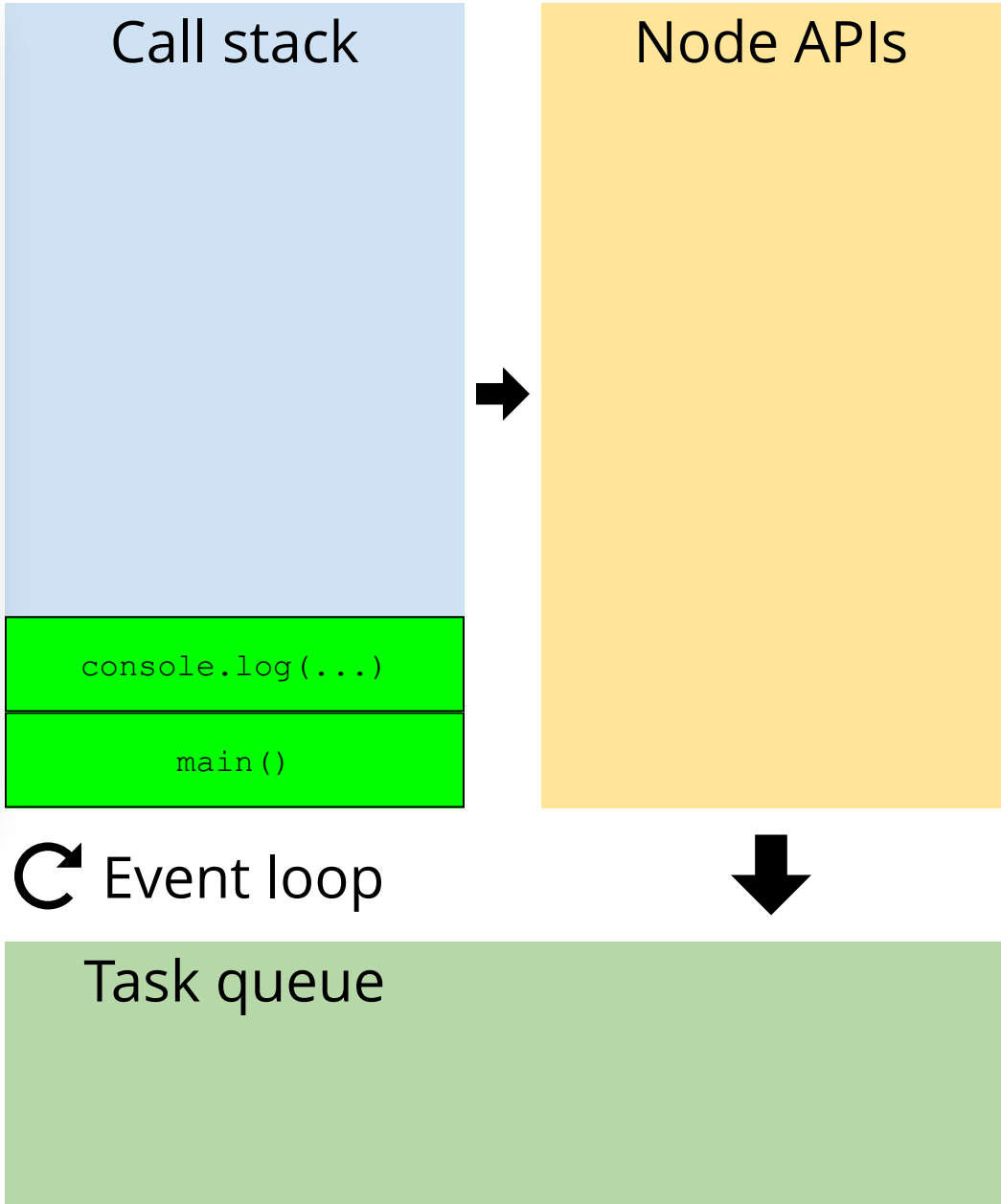
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



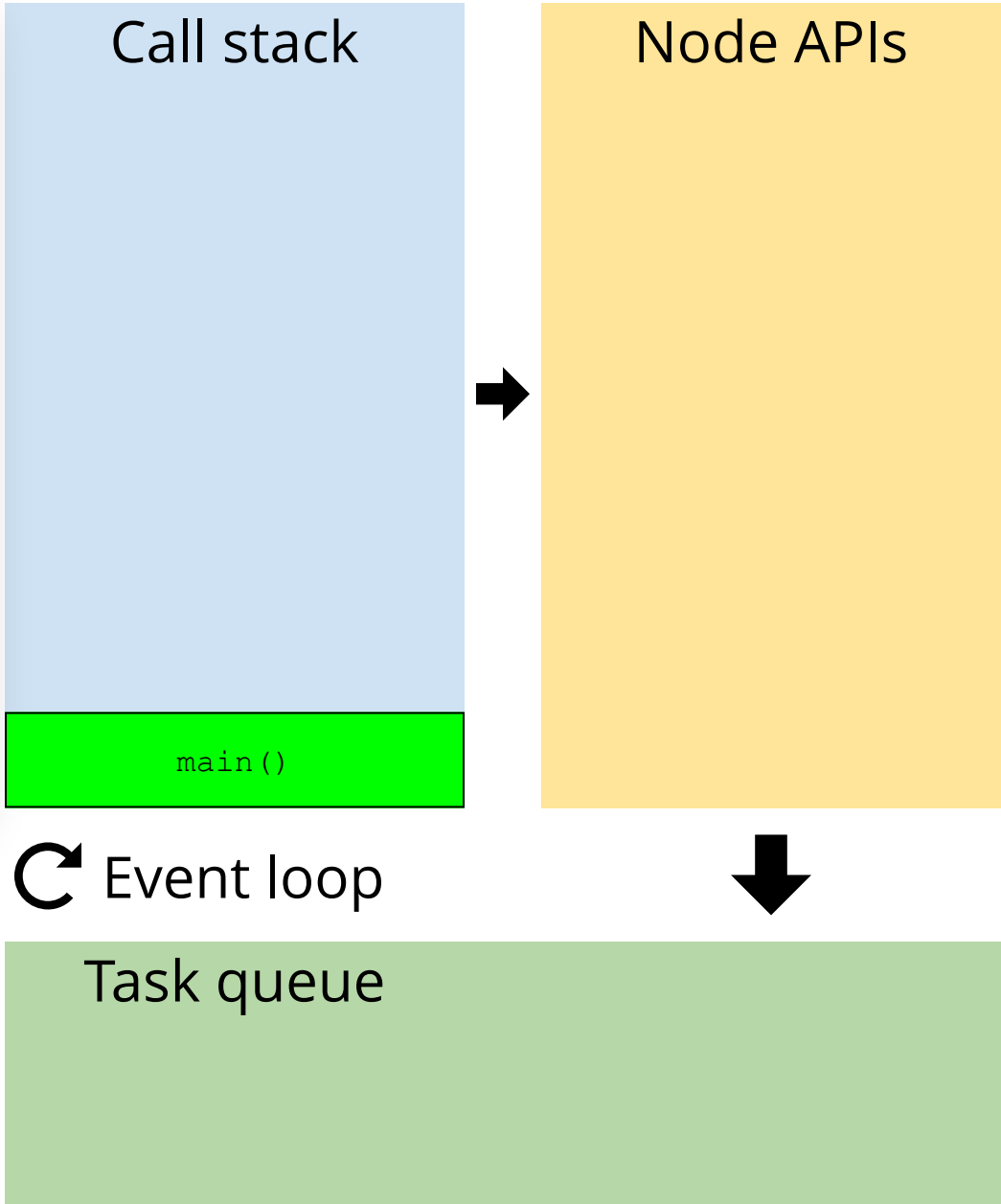
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



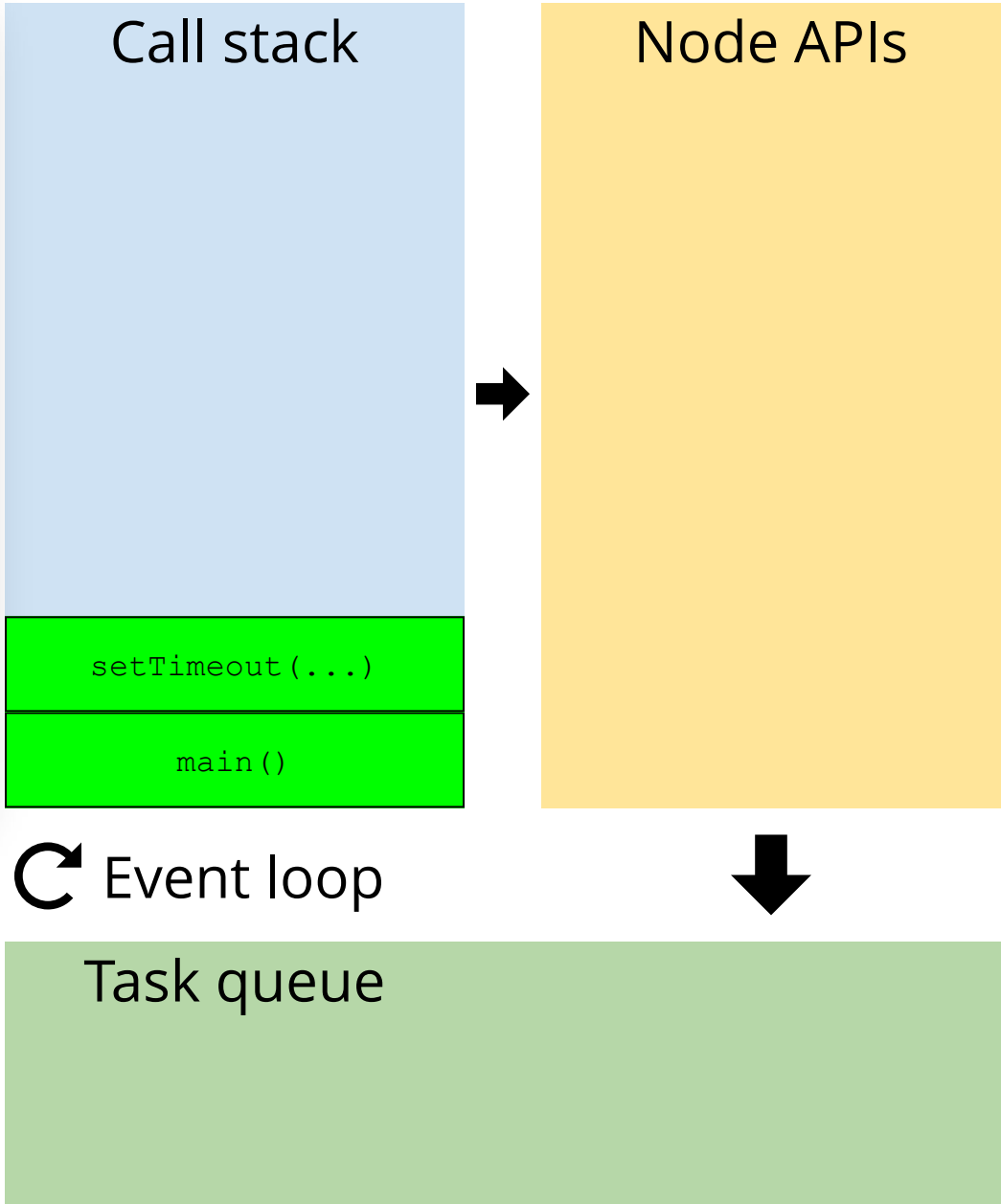
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



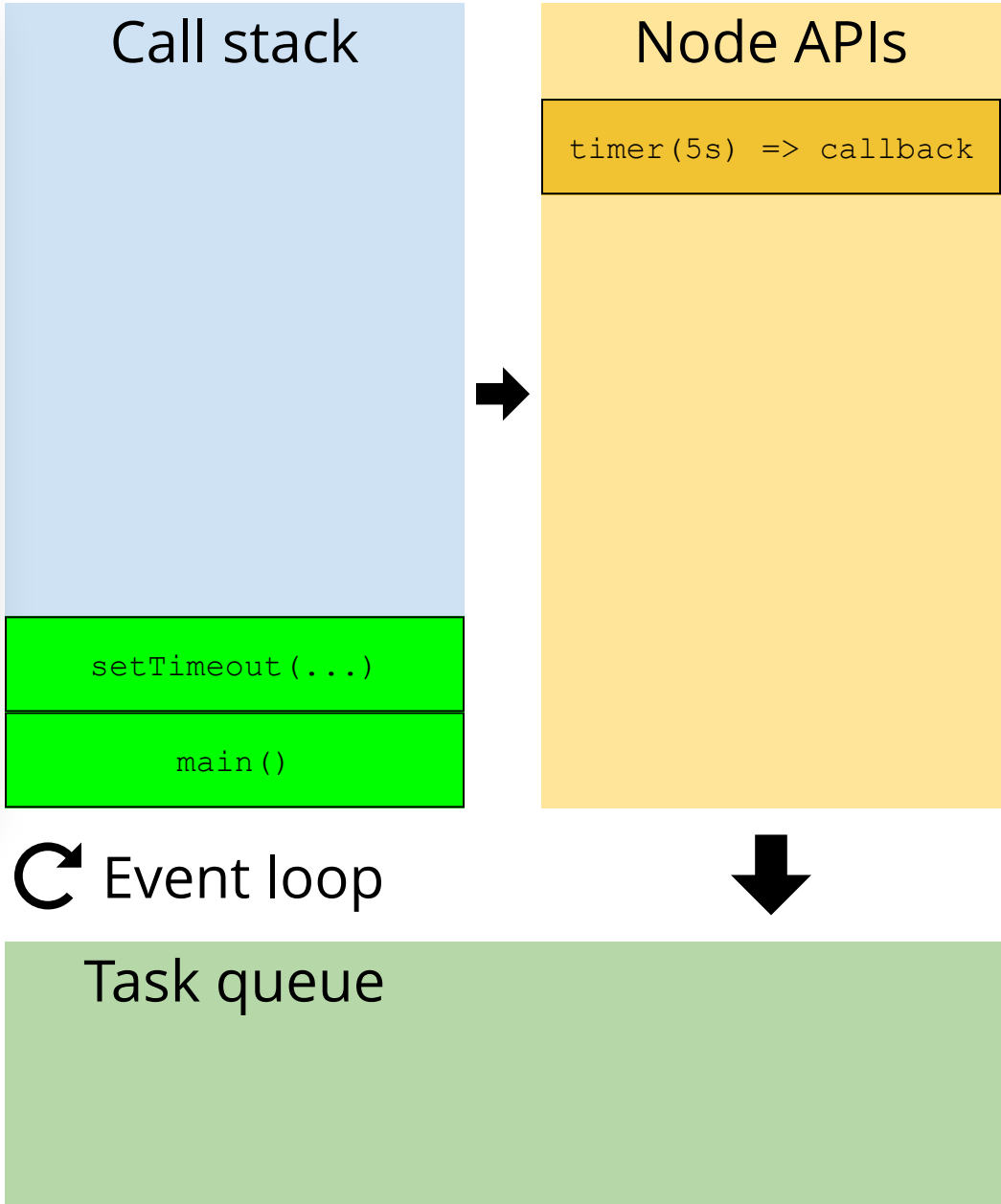
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



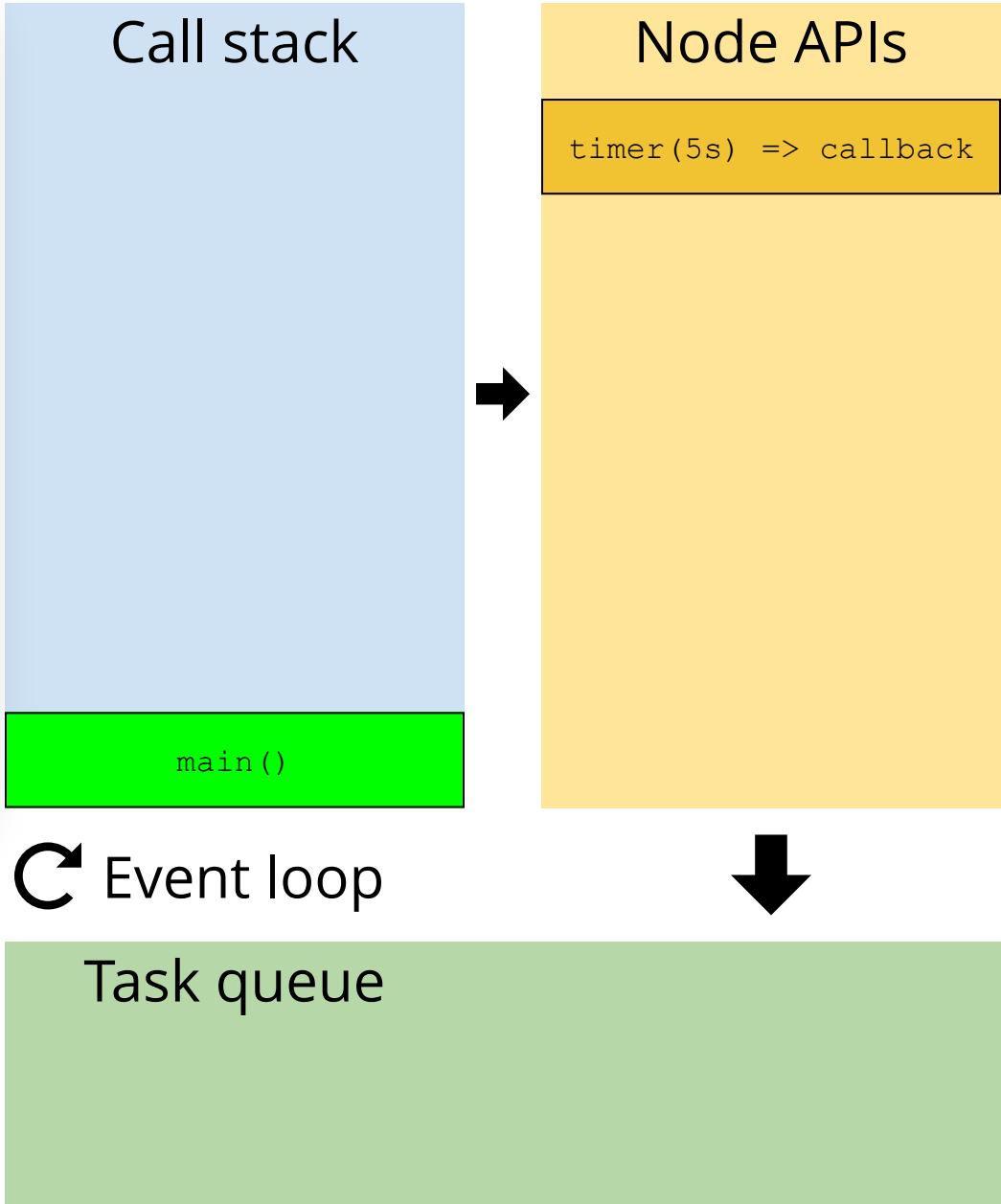
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



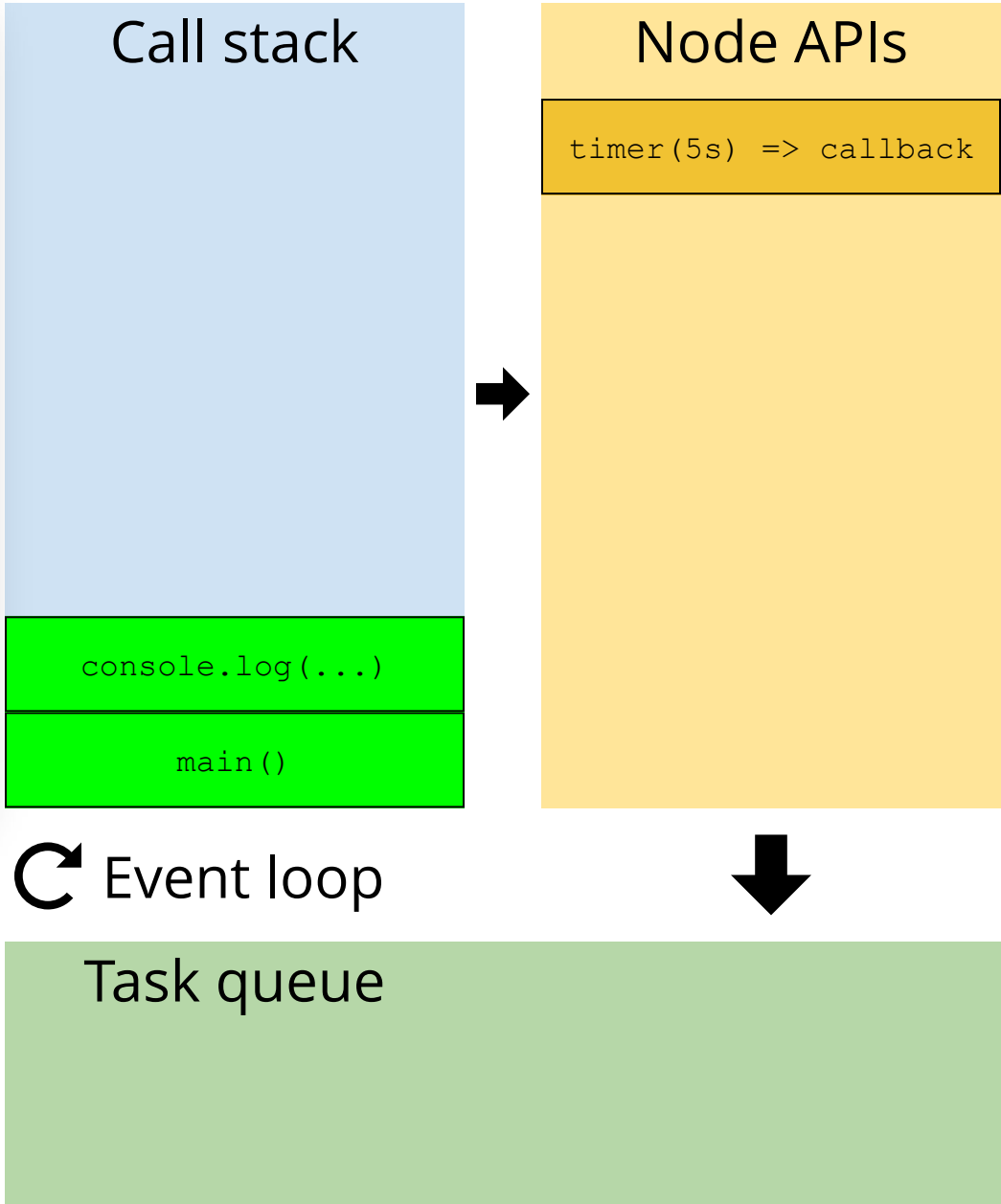
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



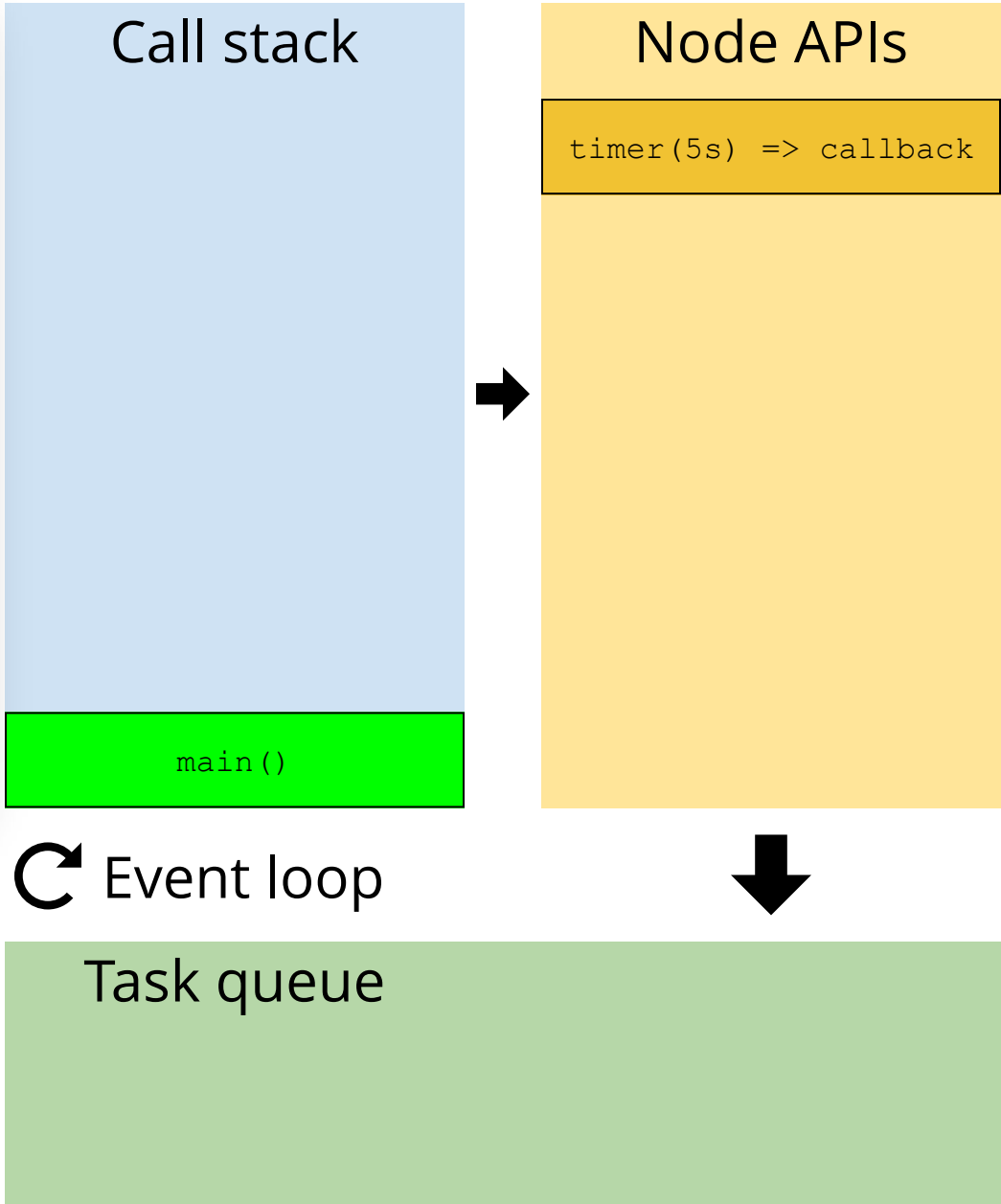
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



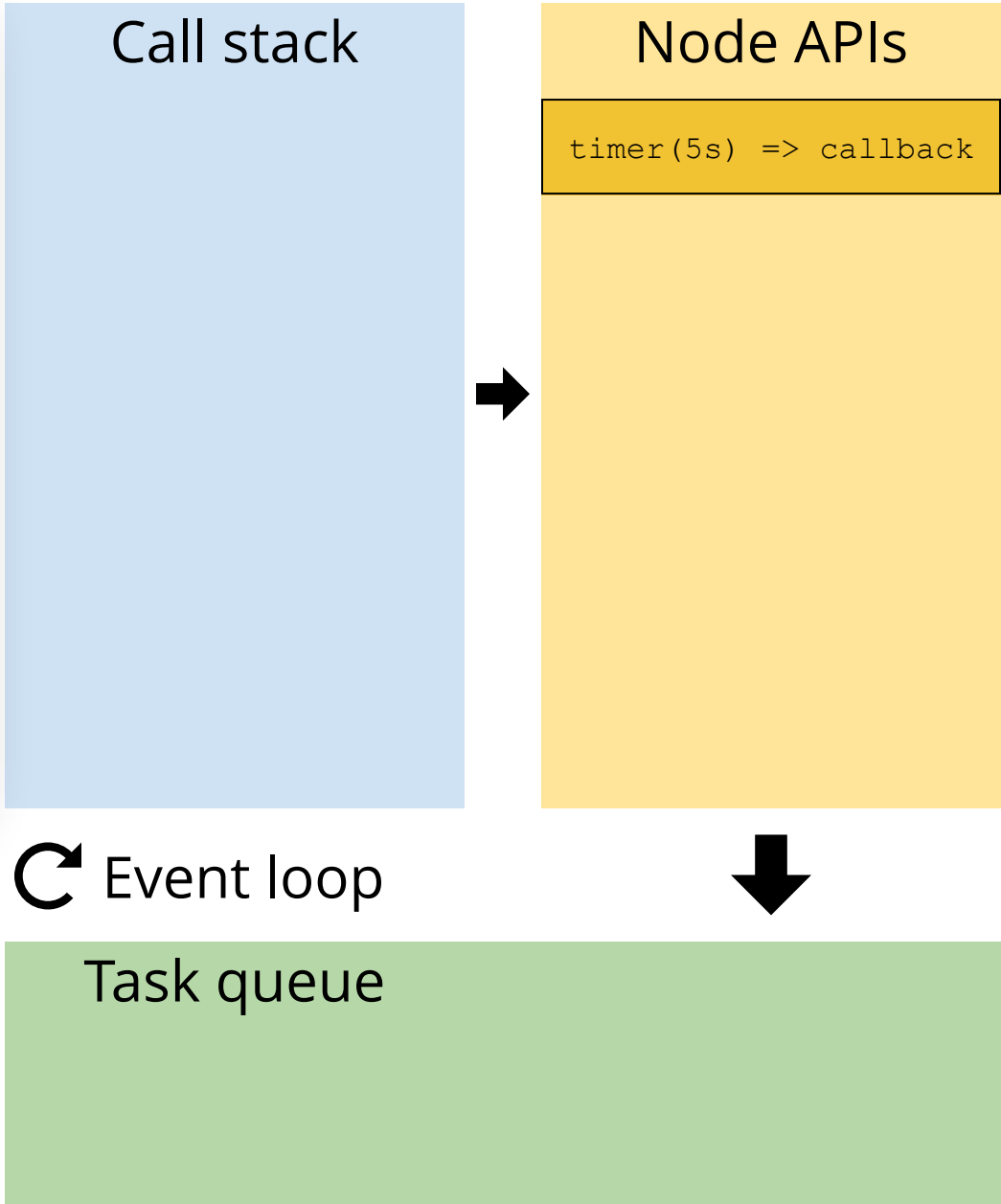
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



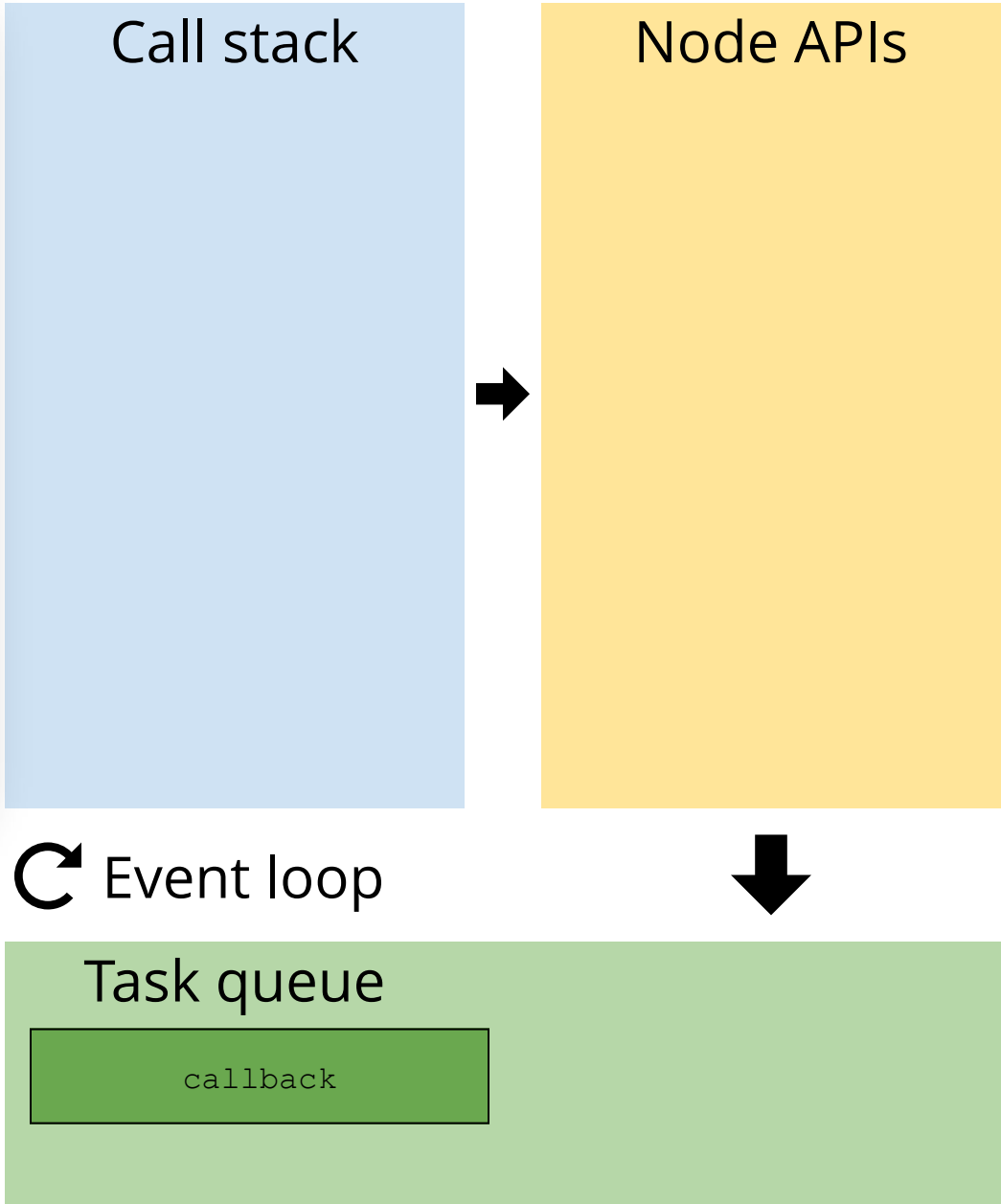
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



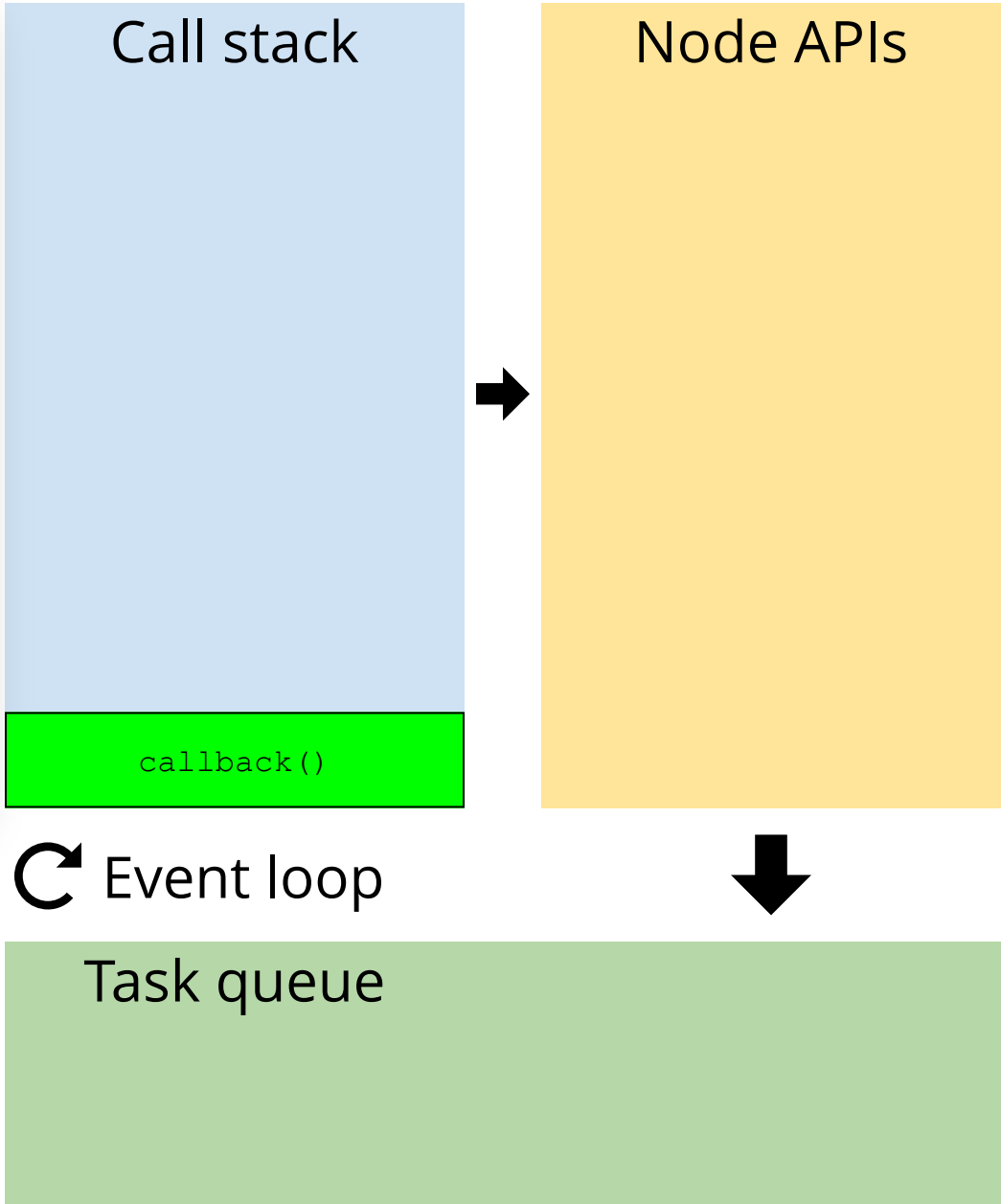
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



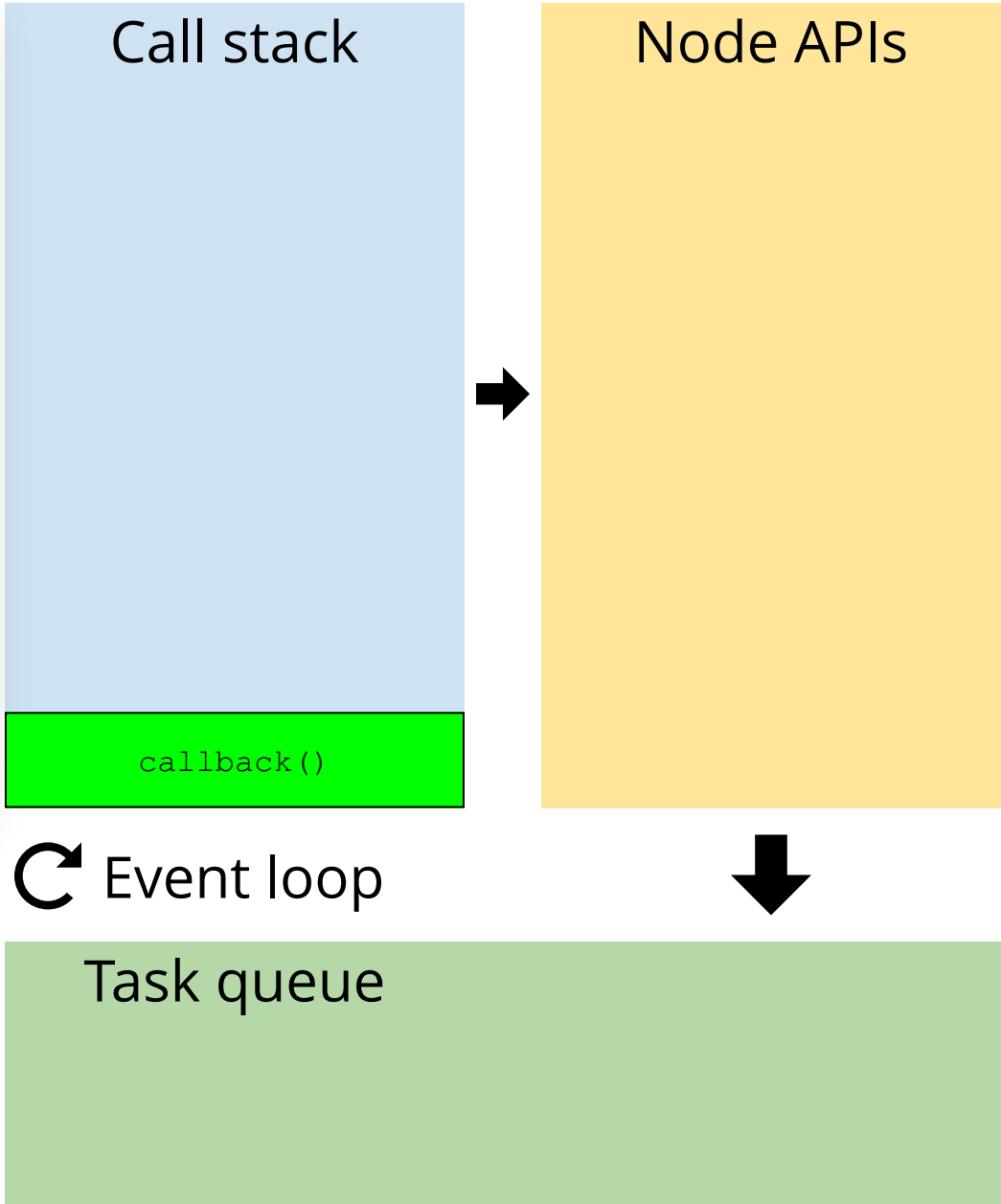
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



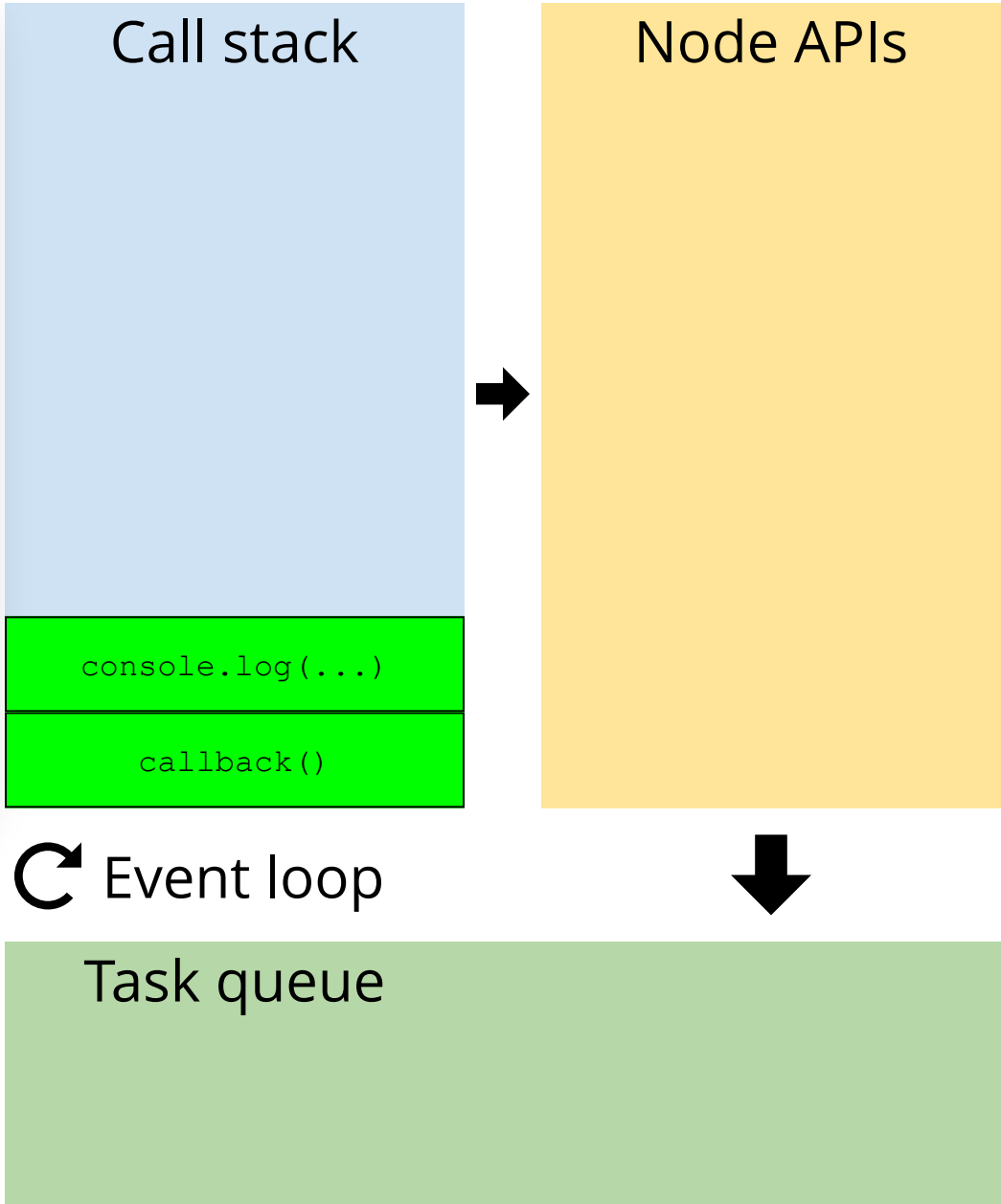
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



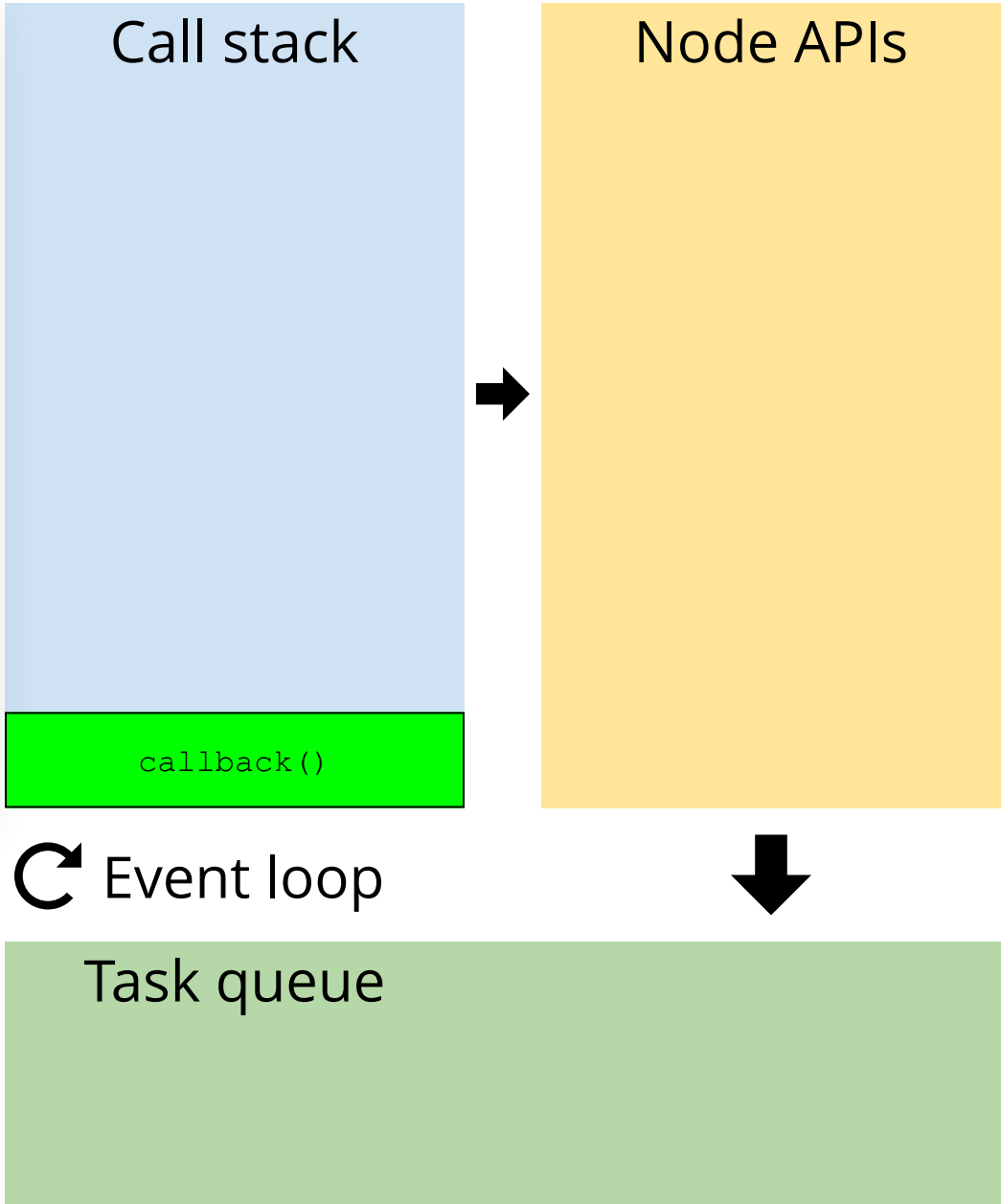
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



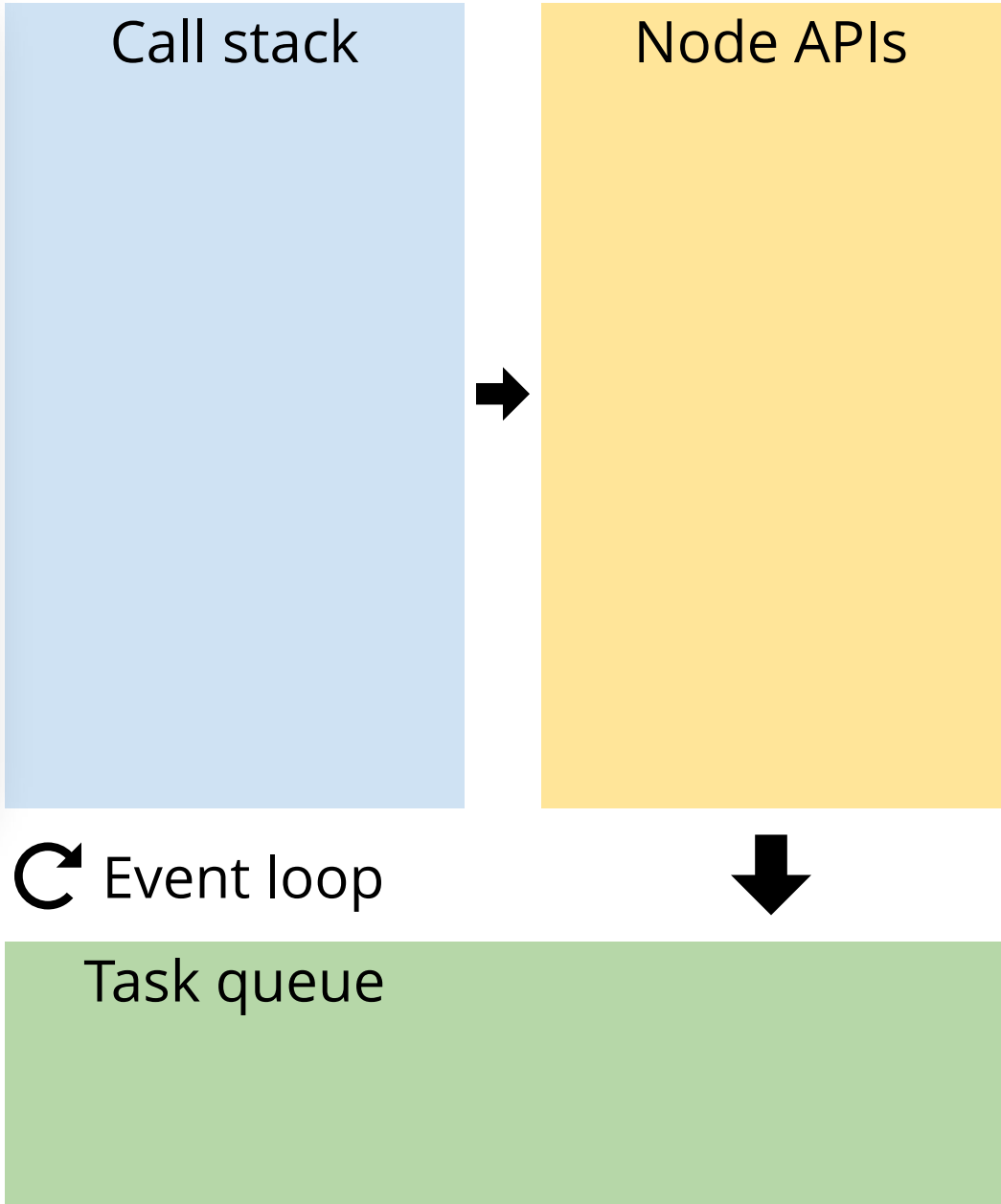
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



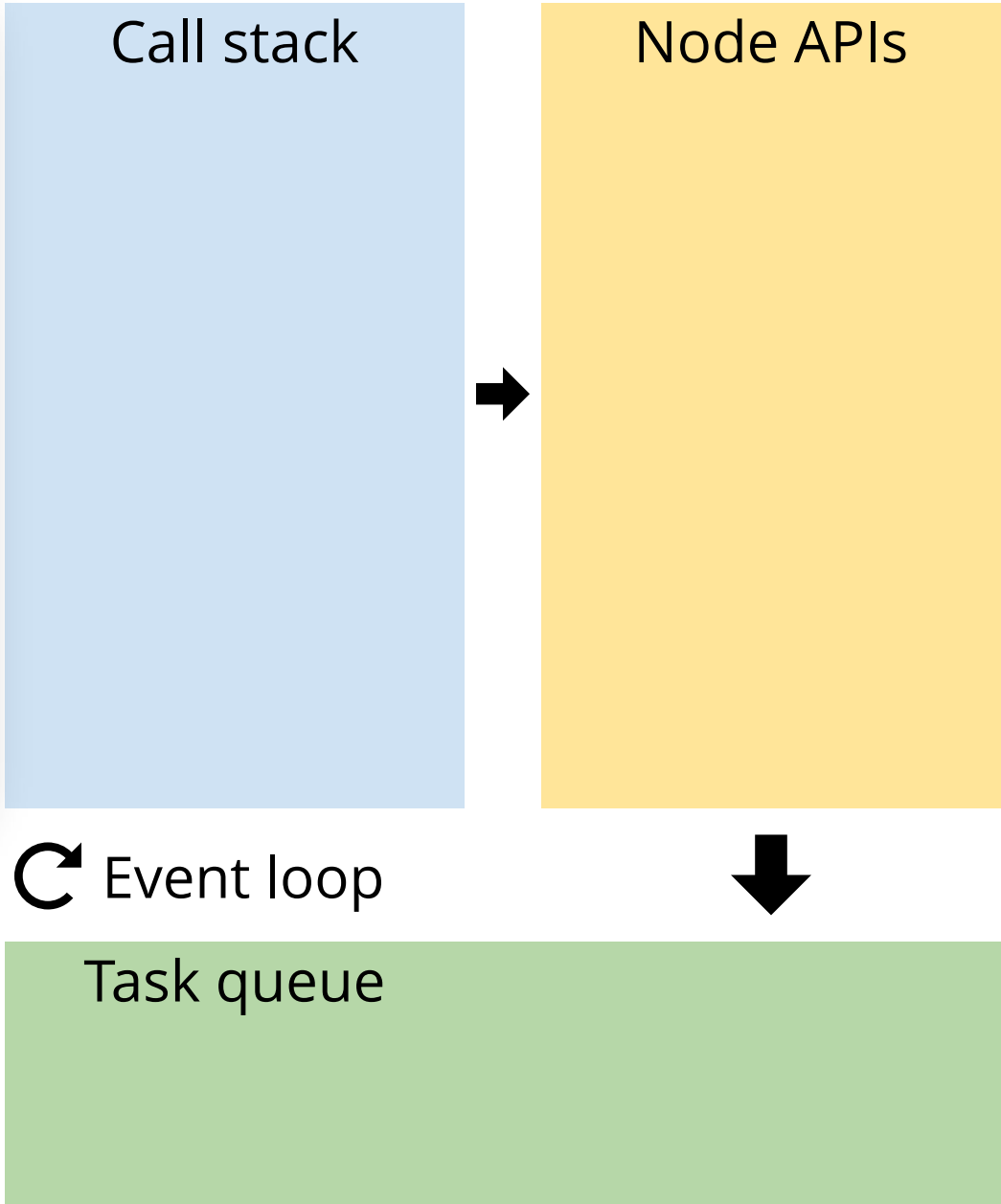
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



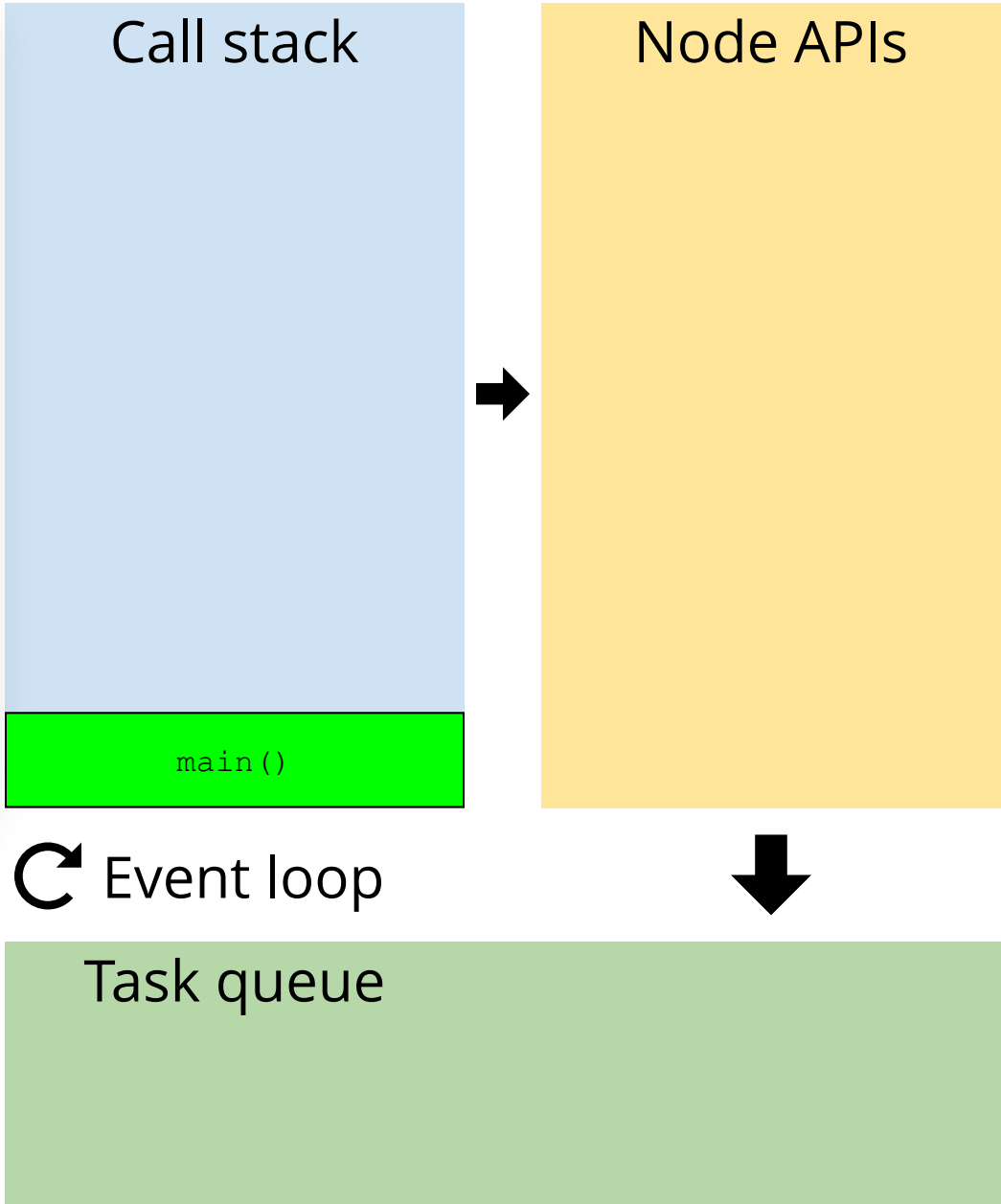
```
1 console.log('starting app');
2
3 setTimeout(function callback() {
4
5   console.log('my function callback');
6
7 }, 5000);
8
9  console.log('end app');
```



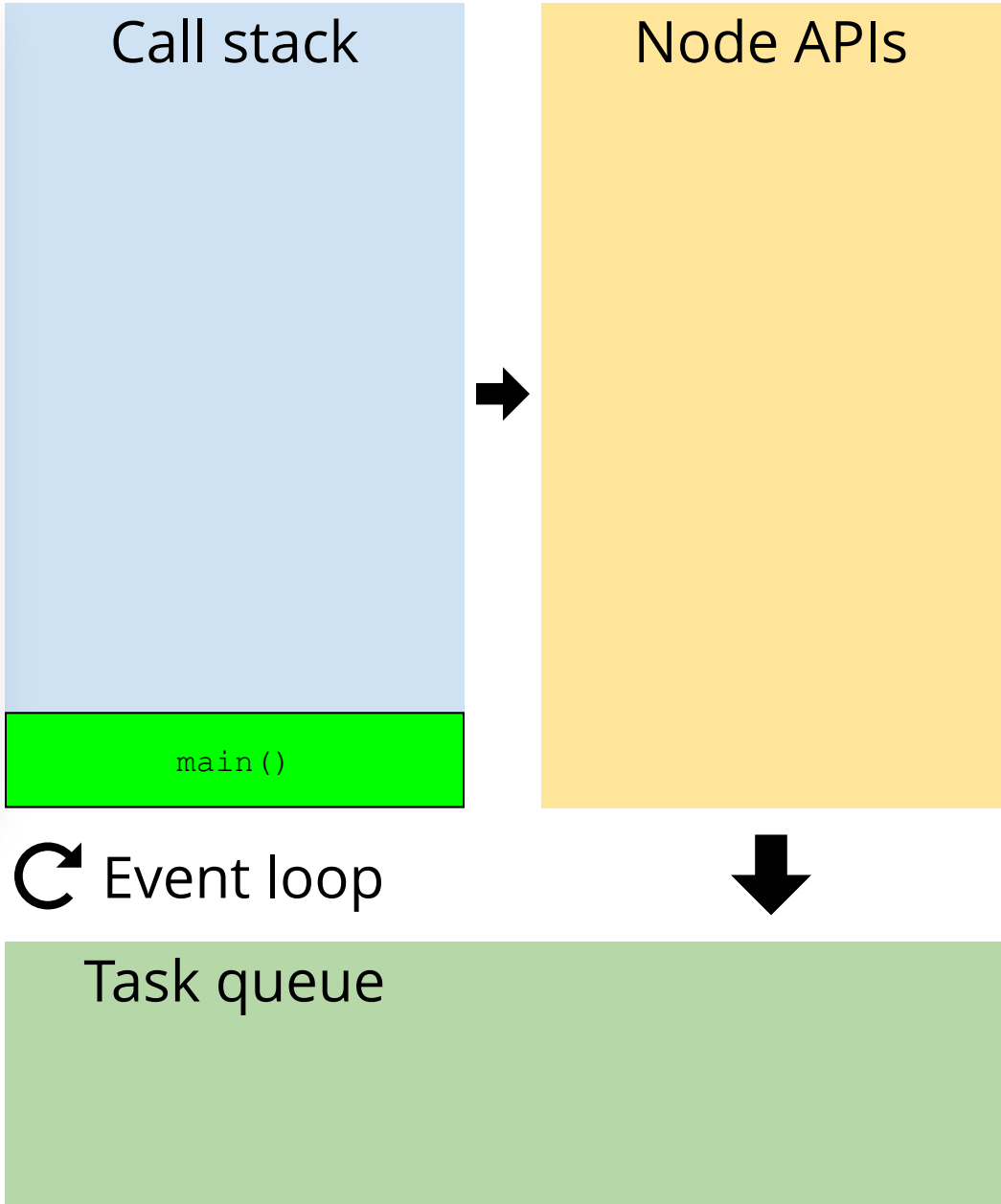
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



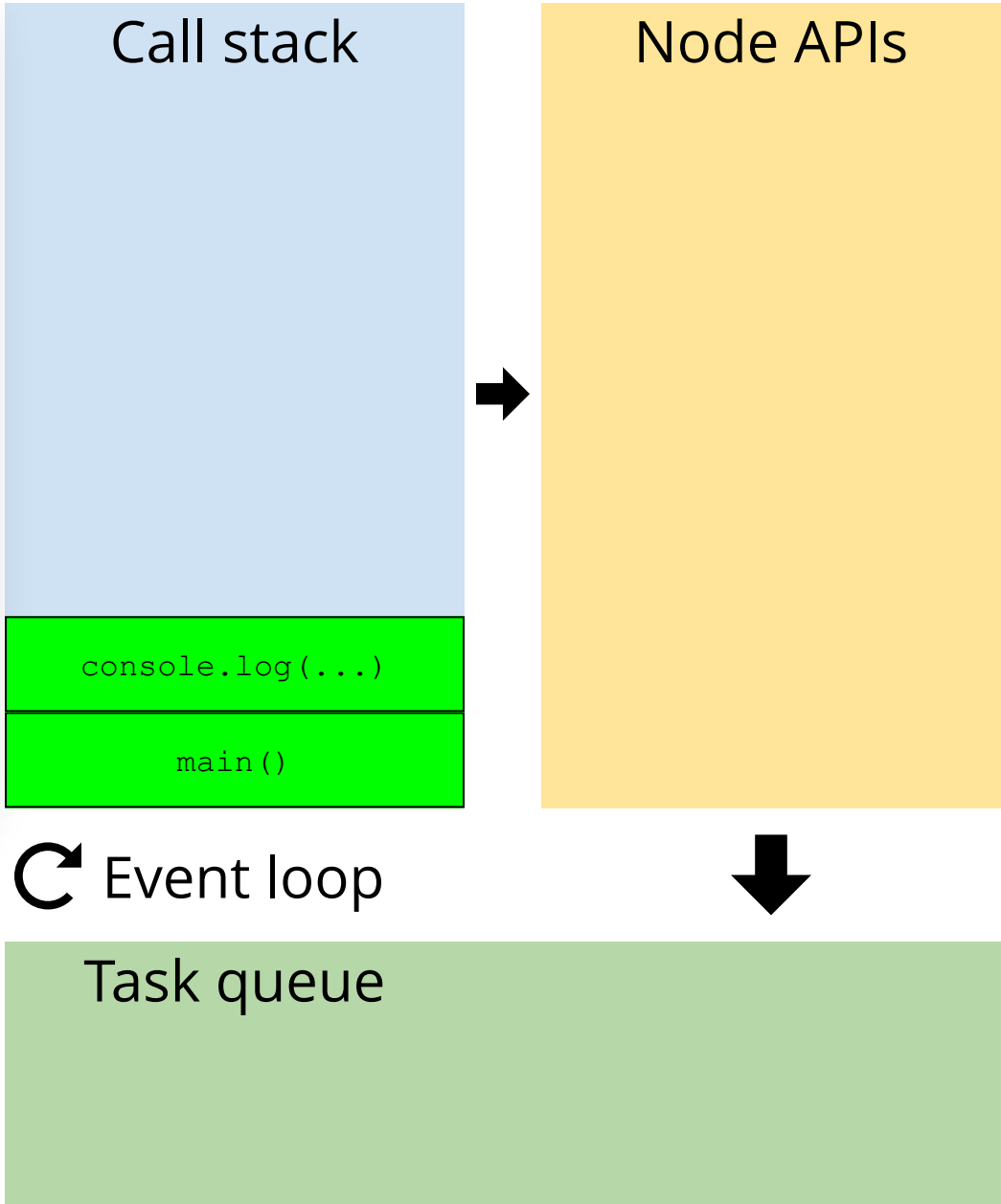
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



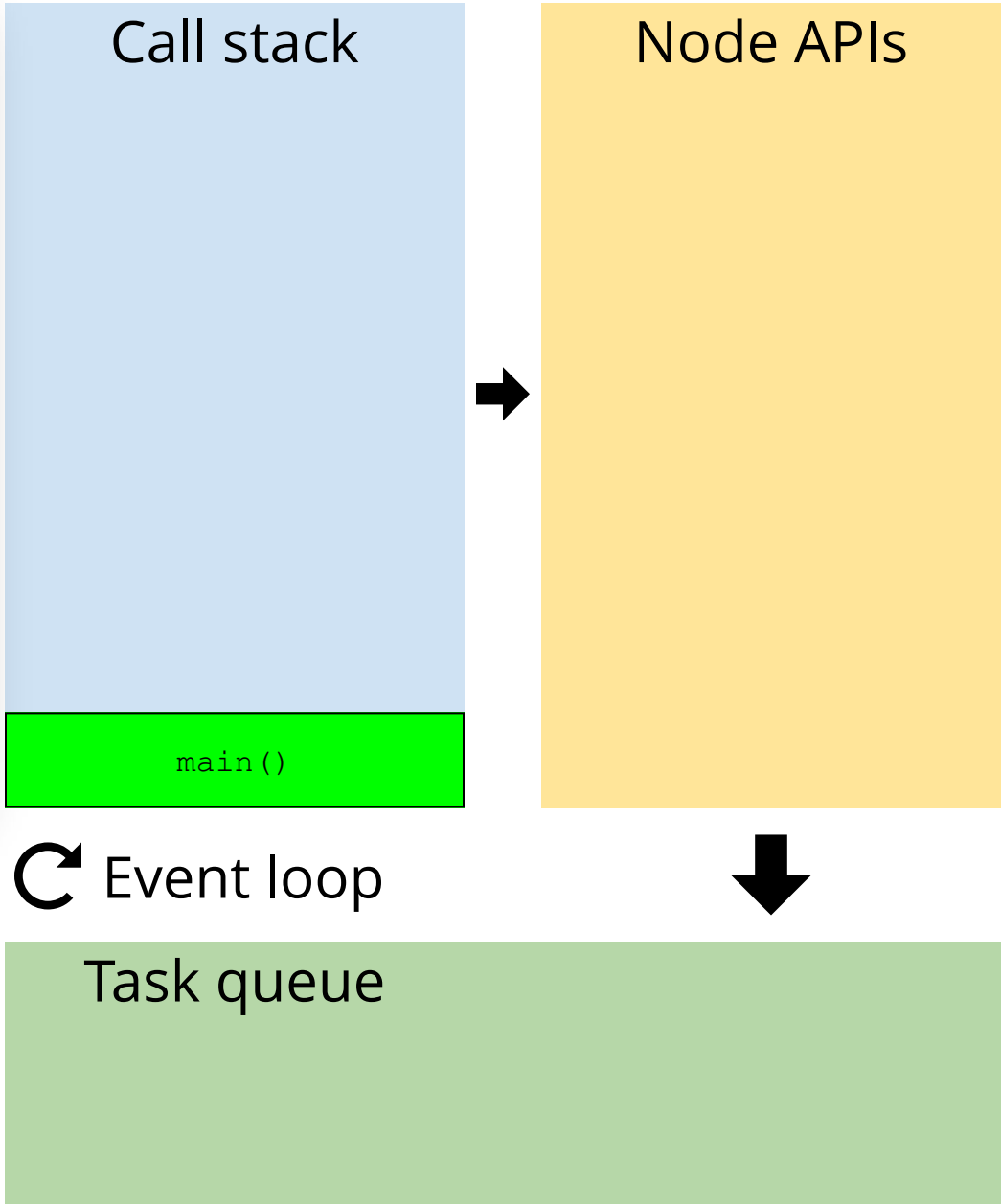
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



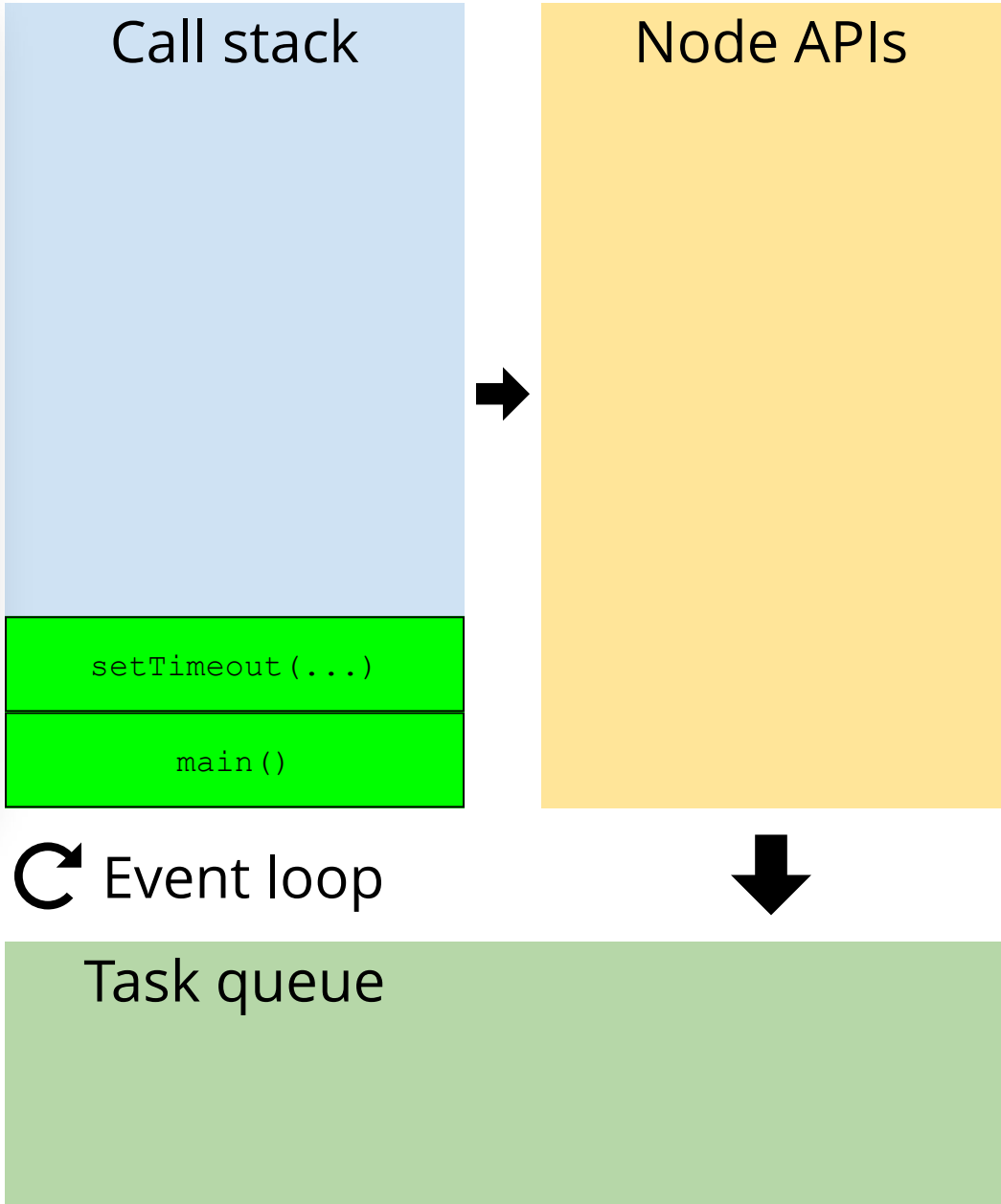
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



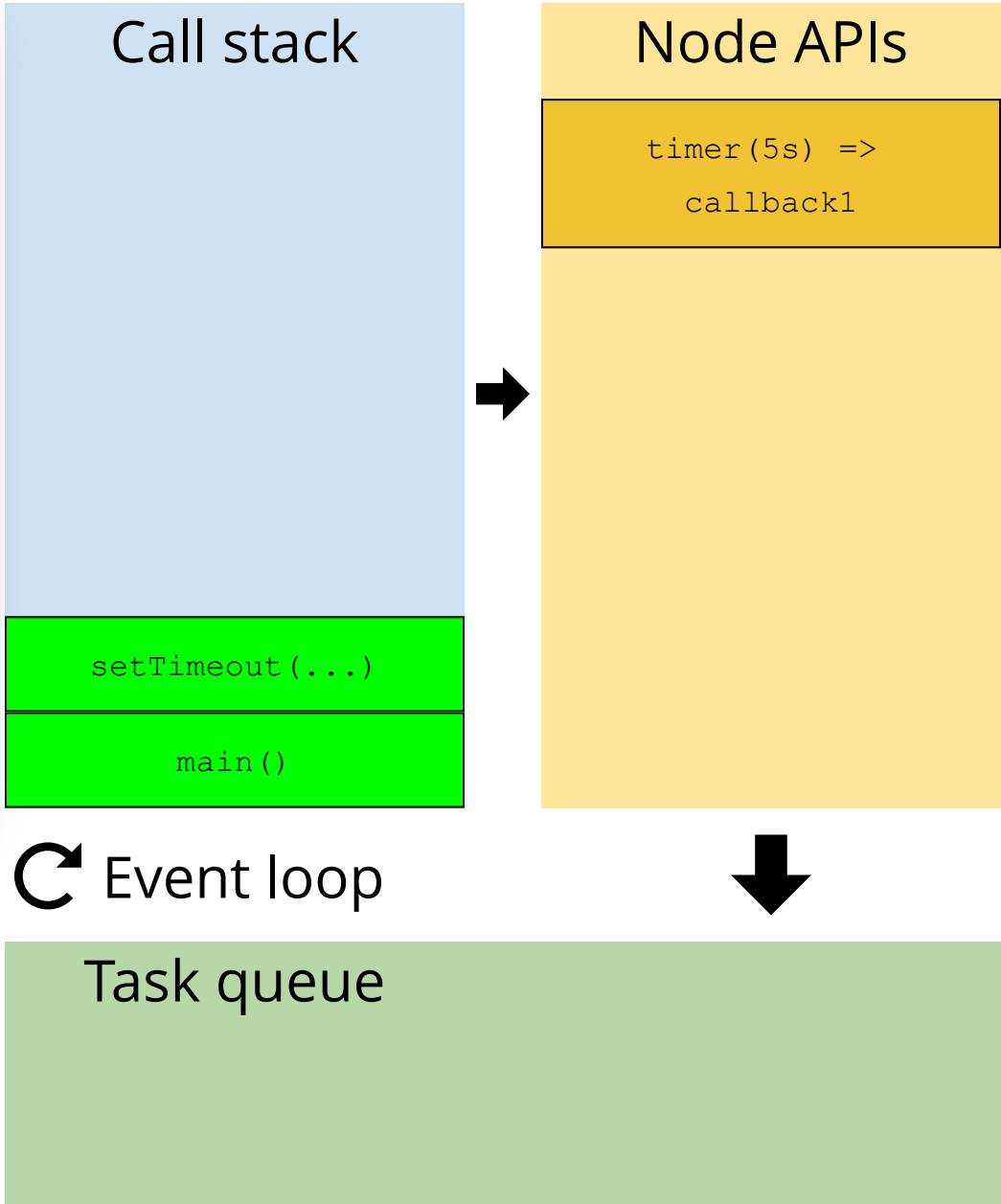
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



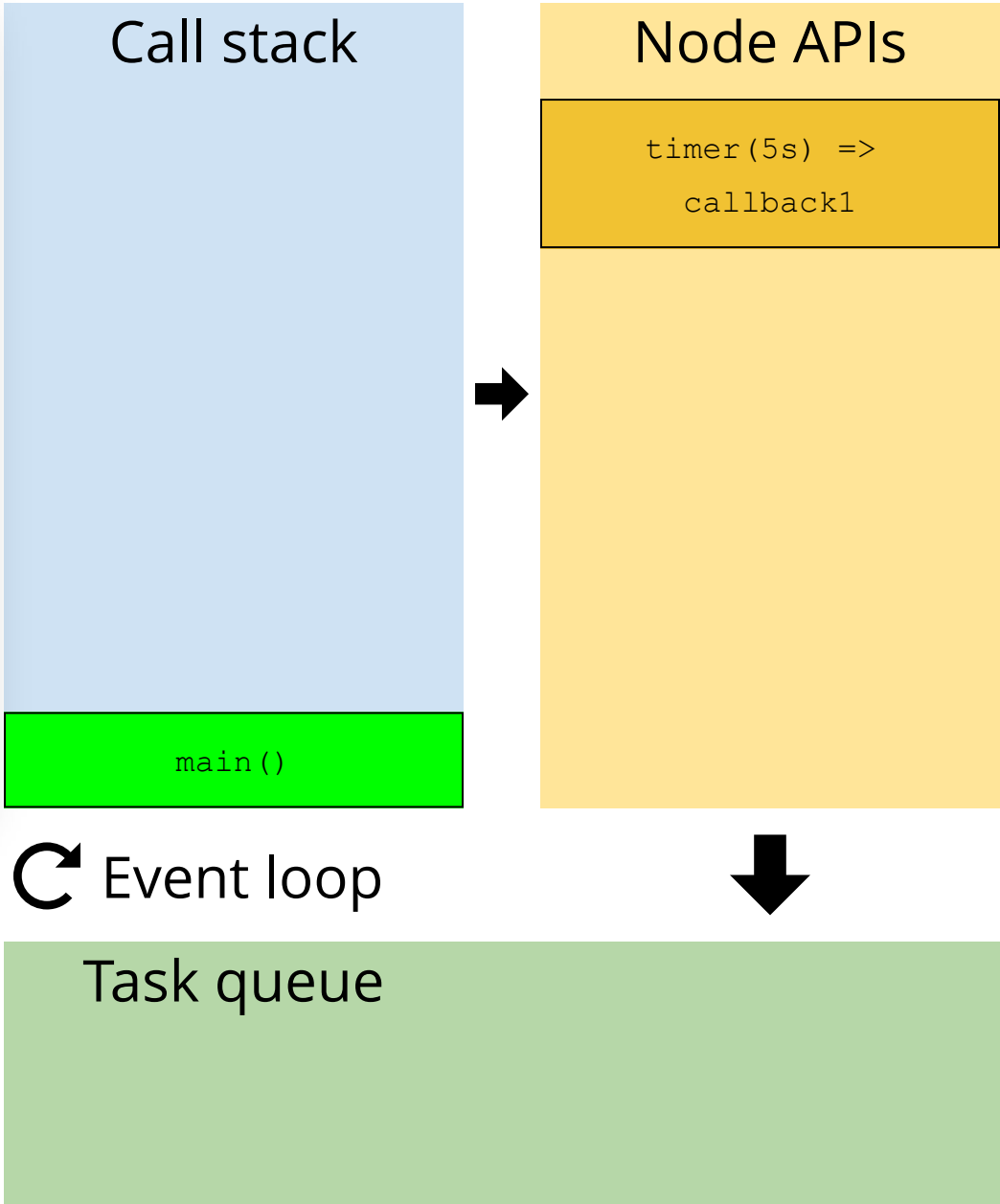
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



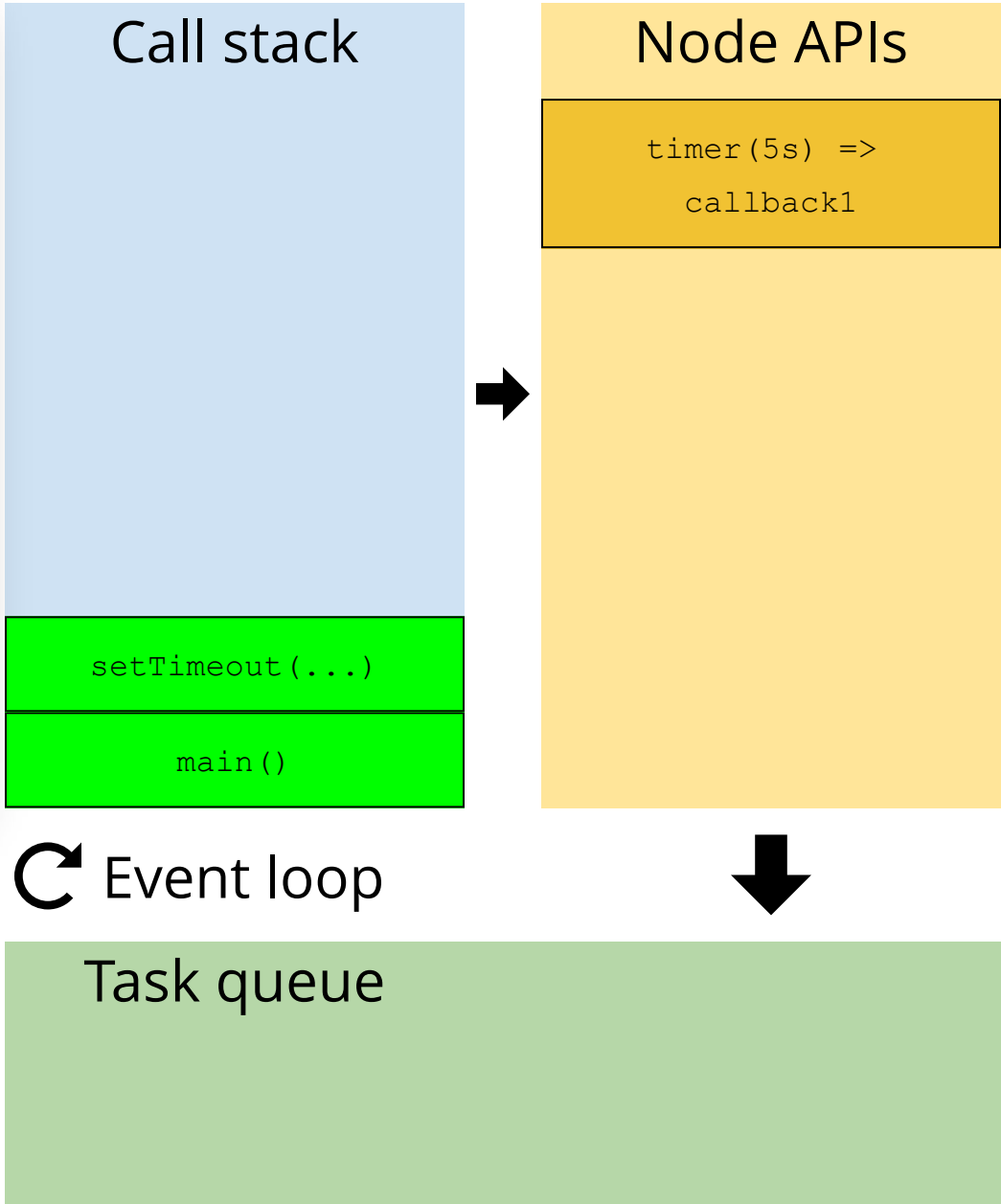
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



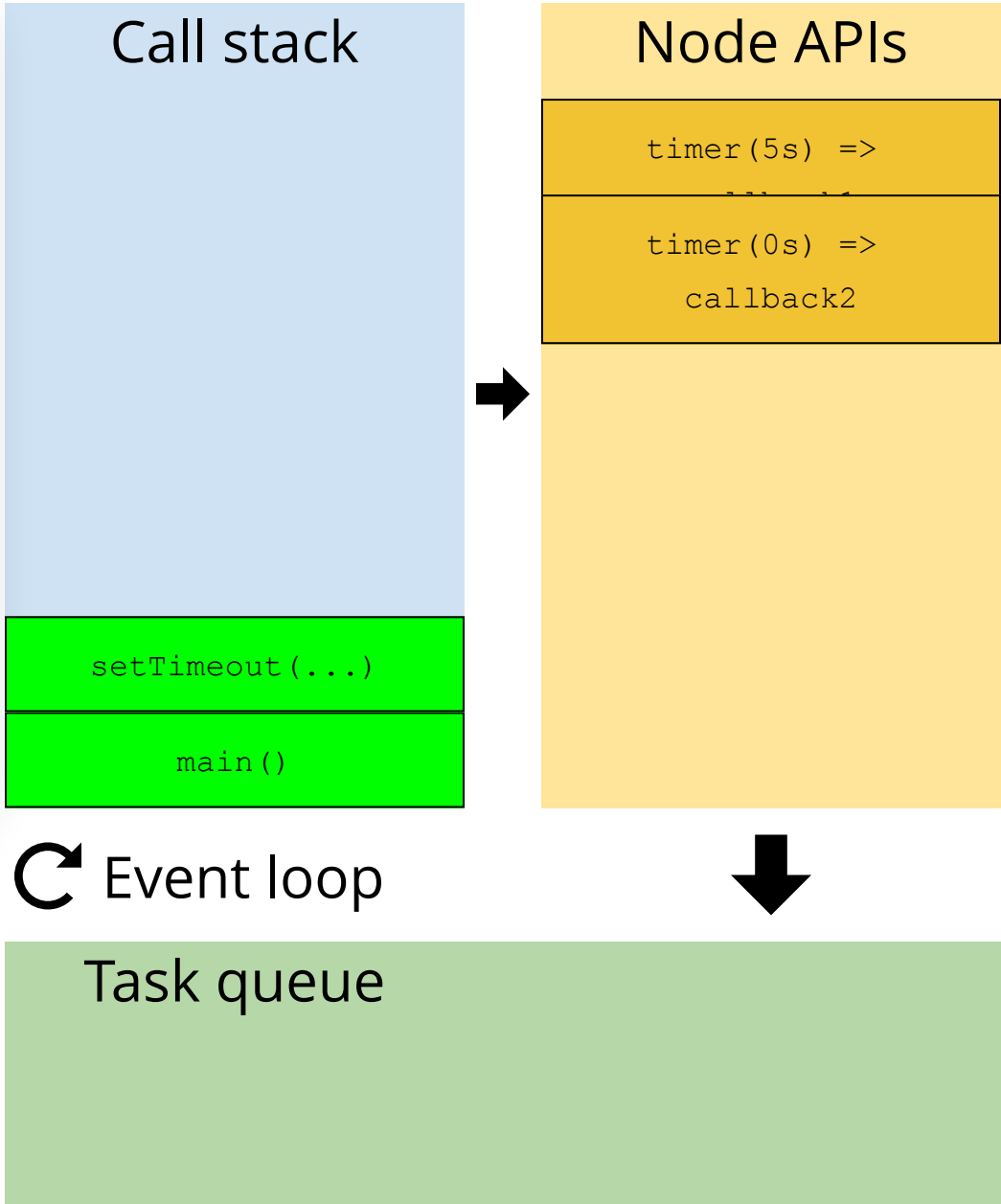
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



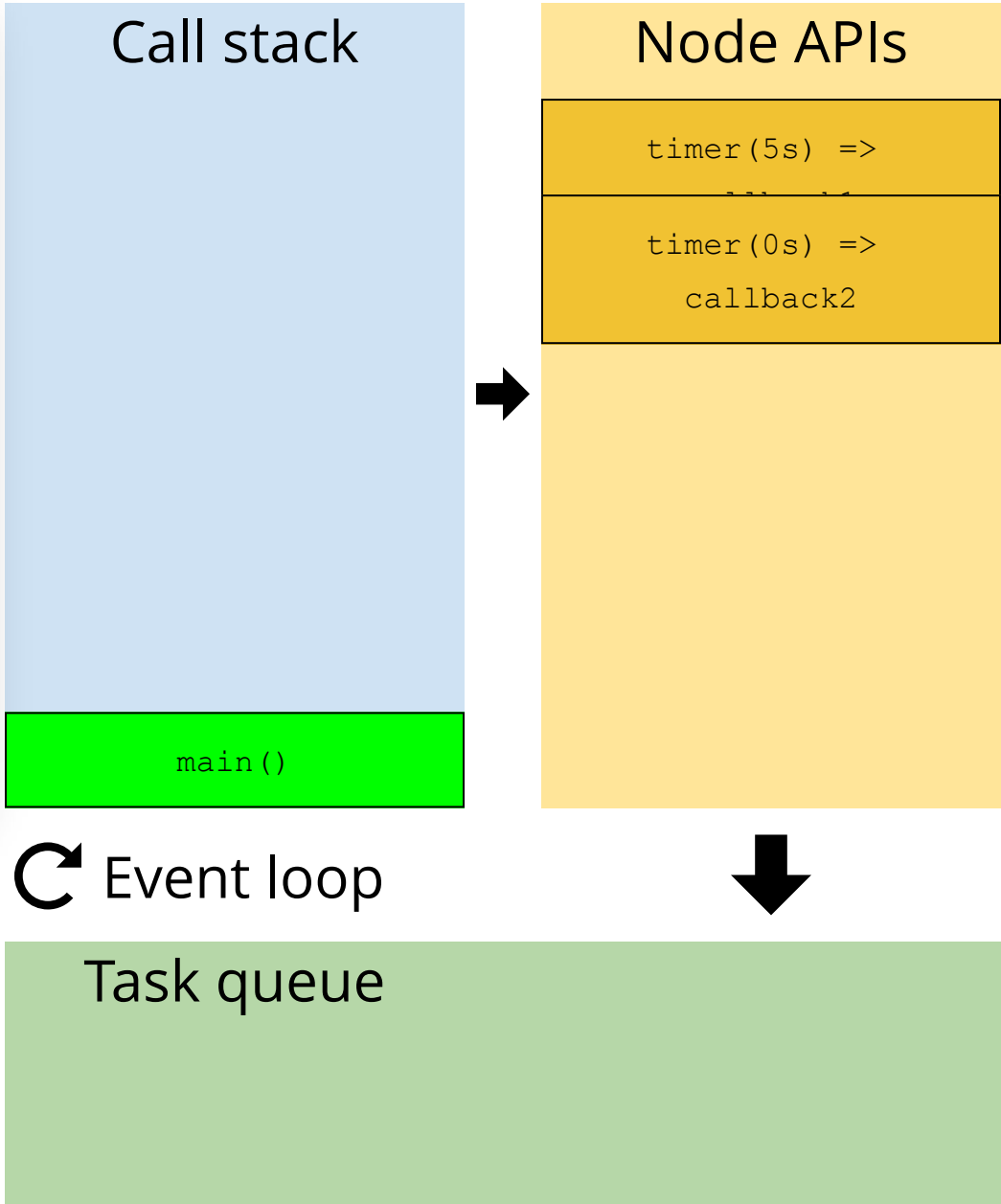
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



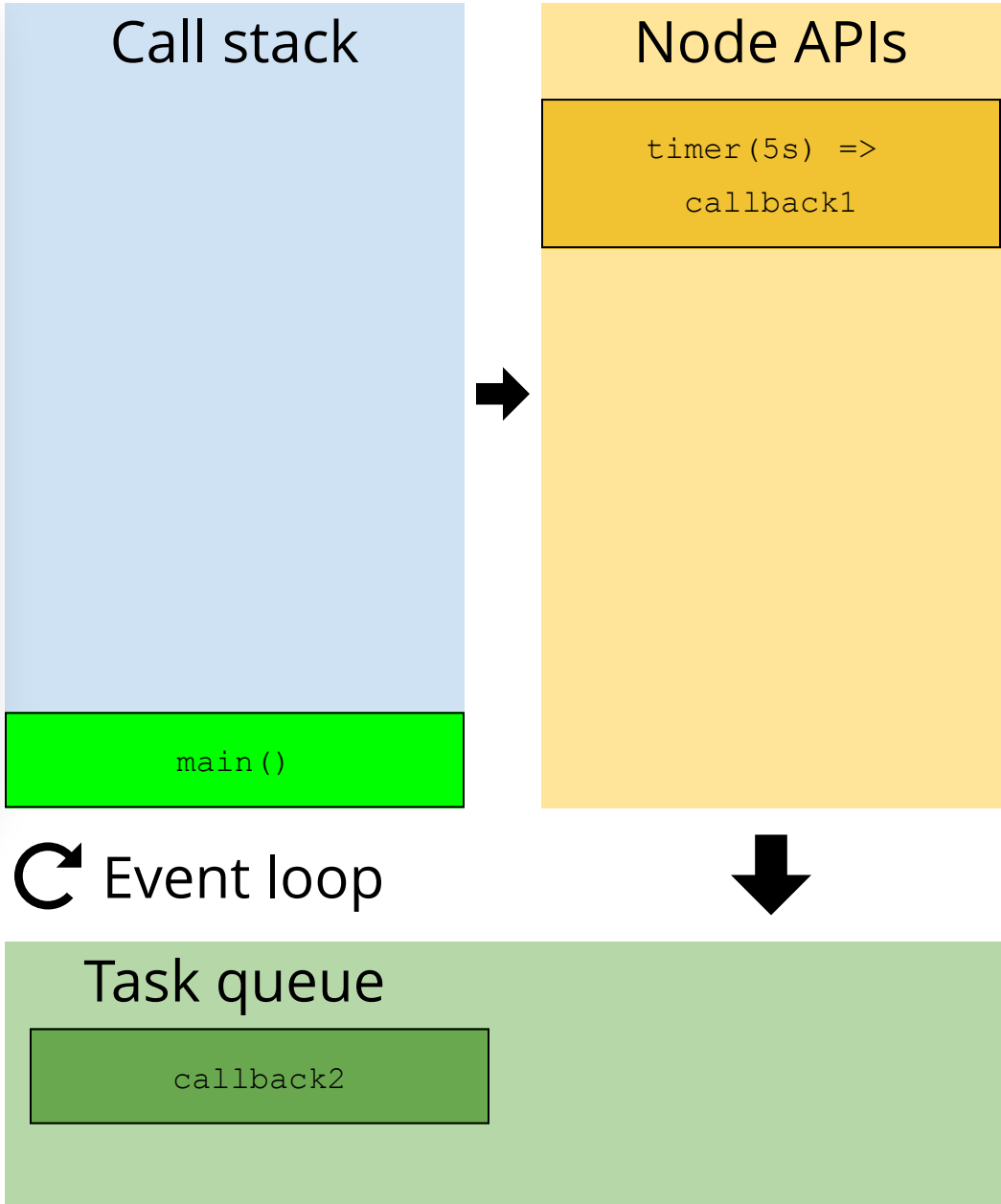
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



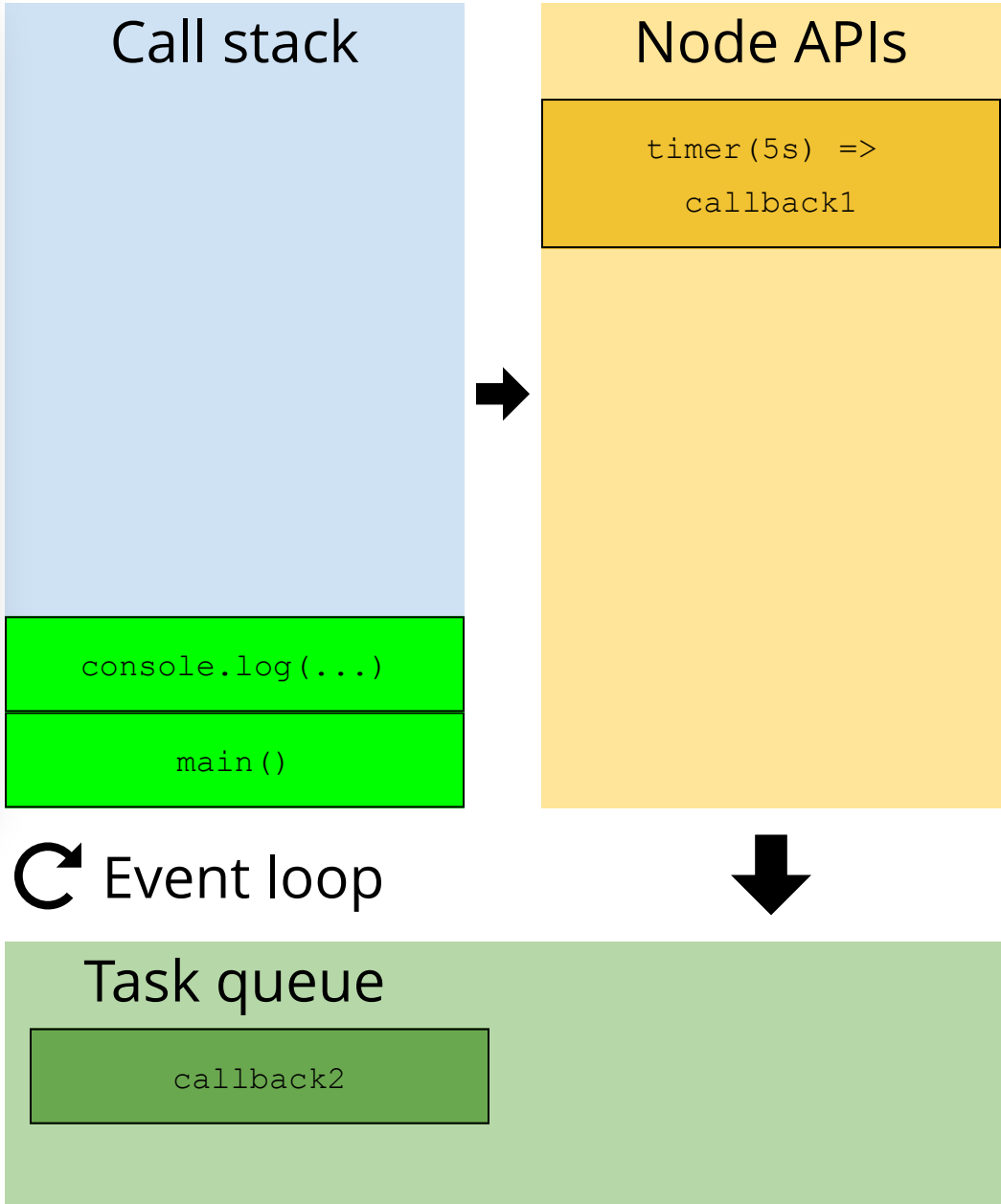
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



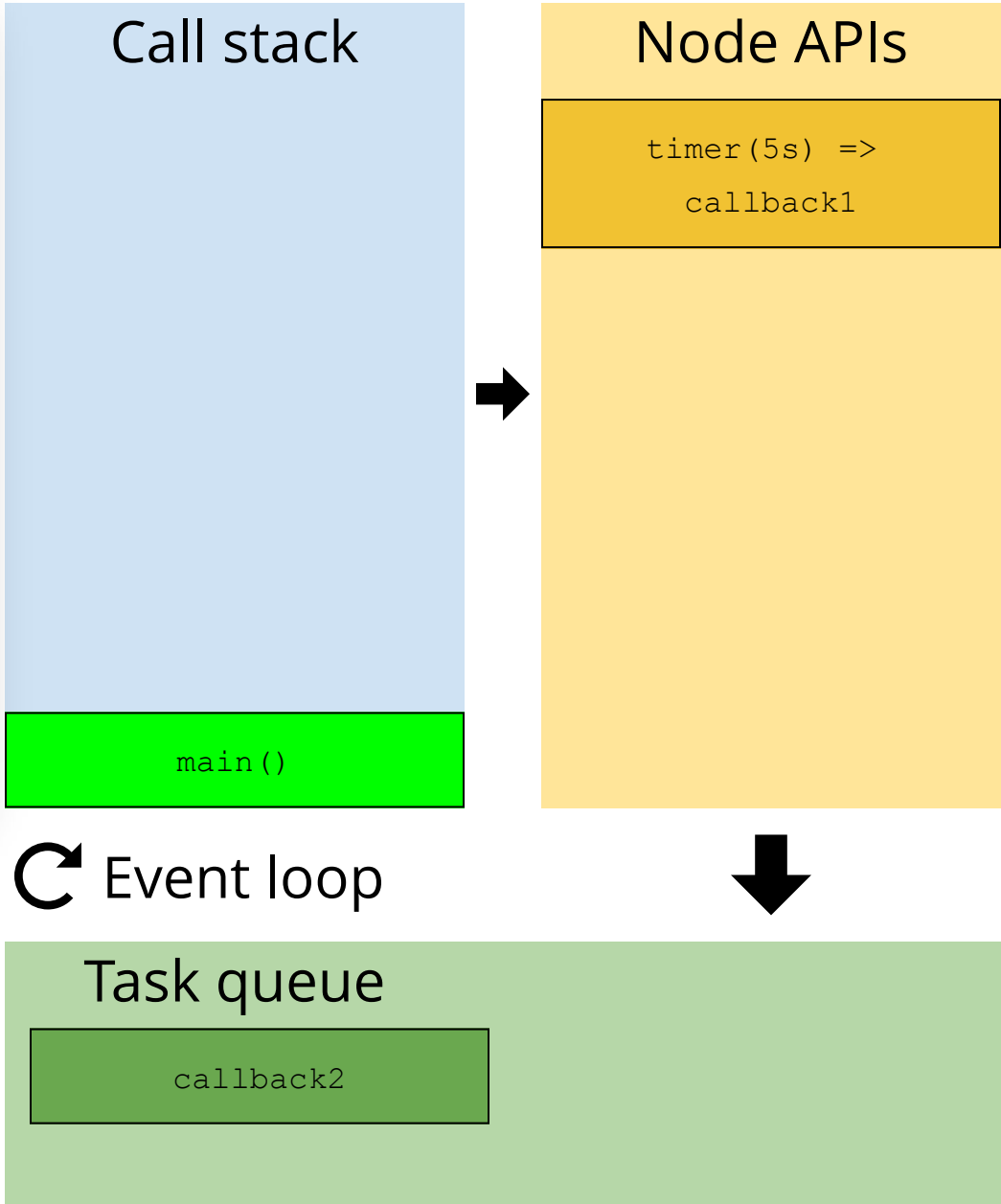
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



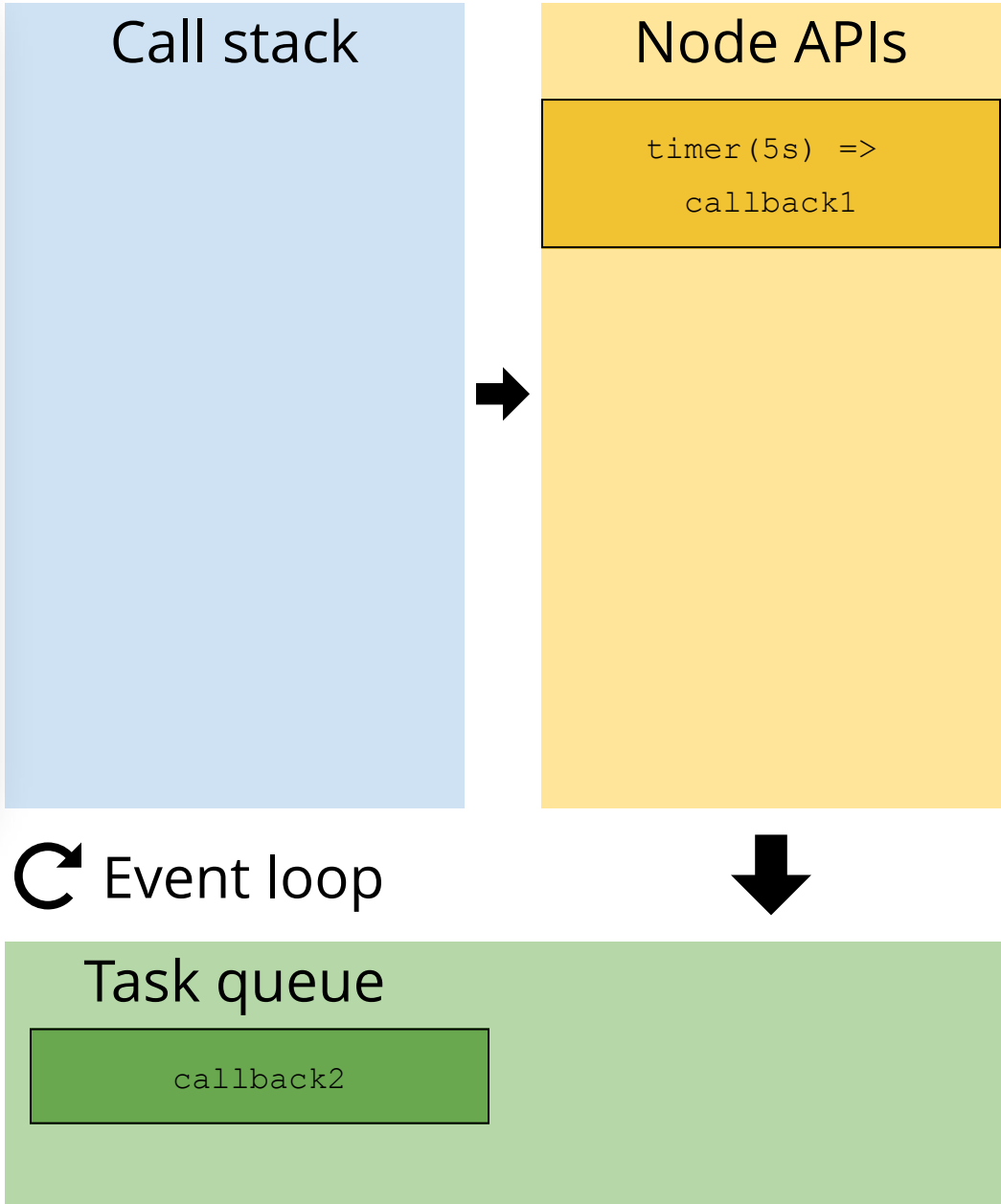
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



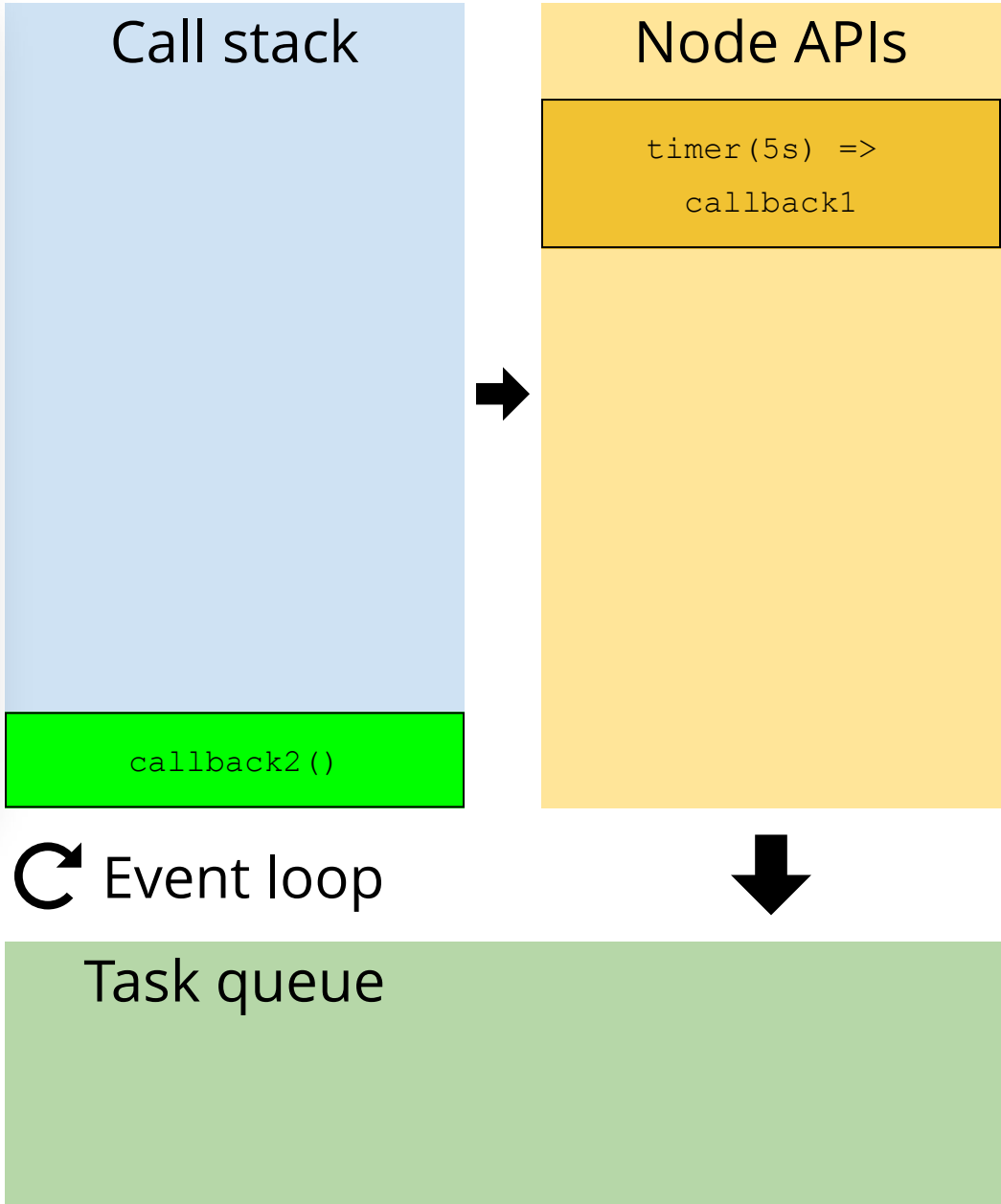
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



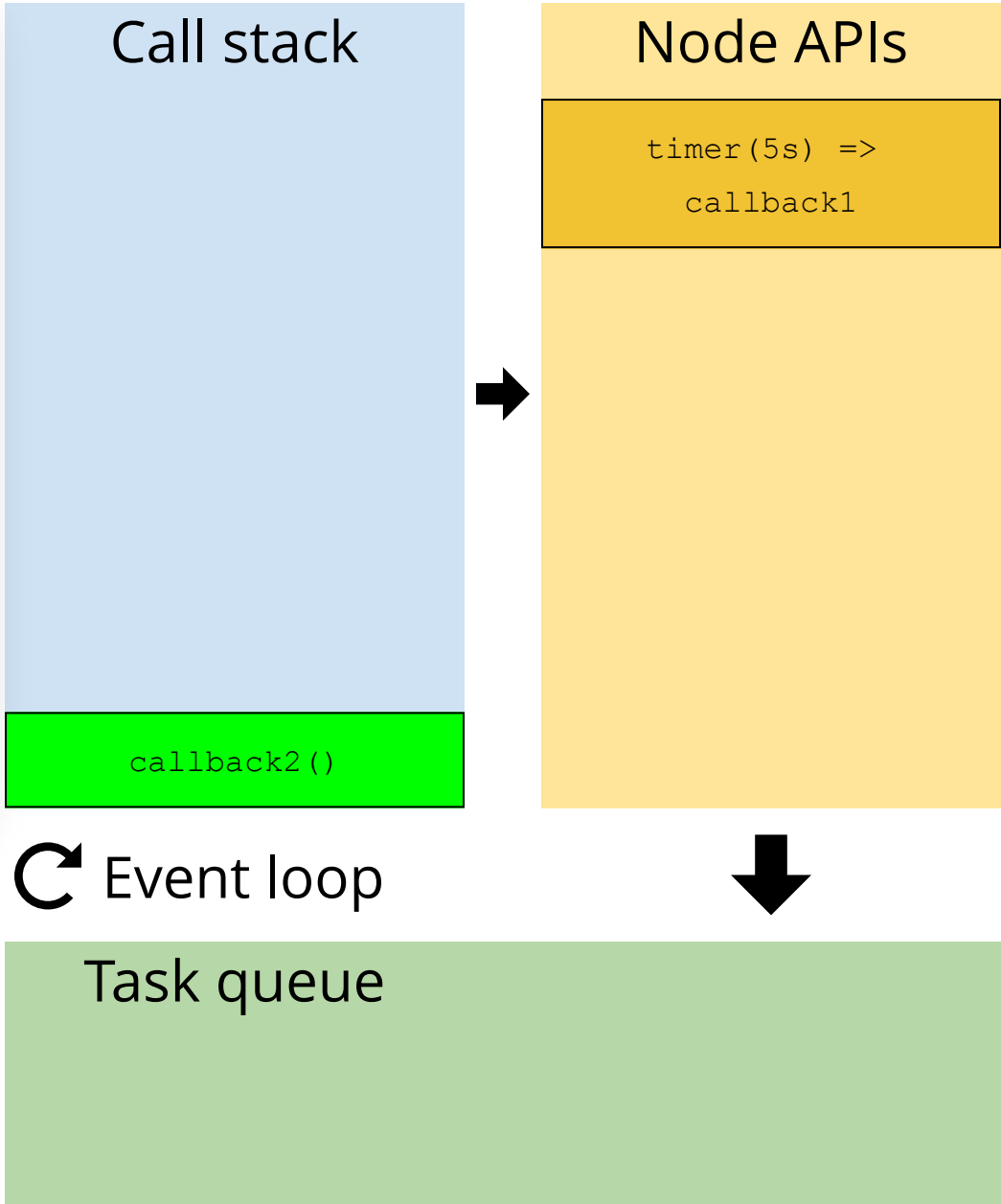
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



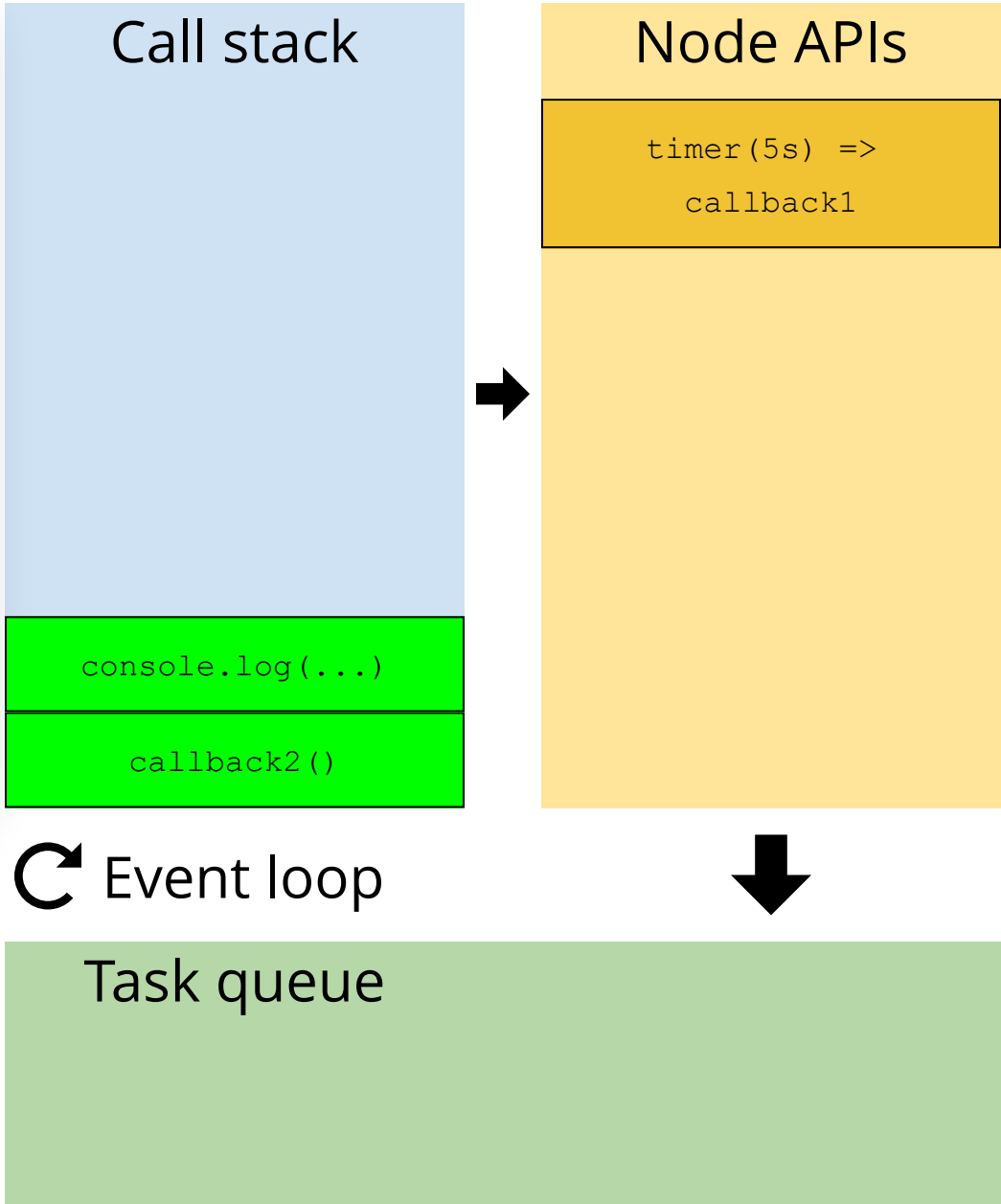
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



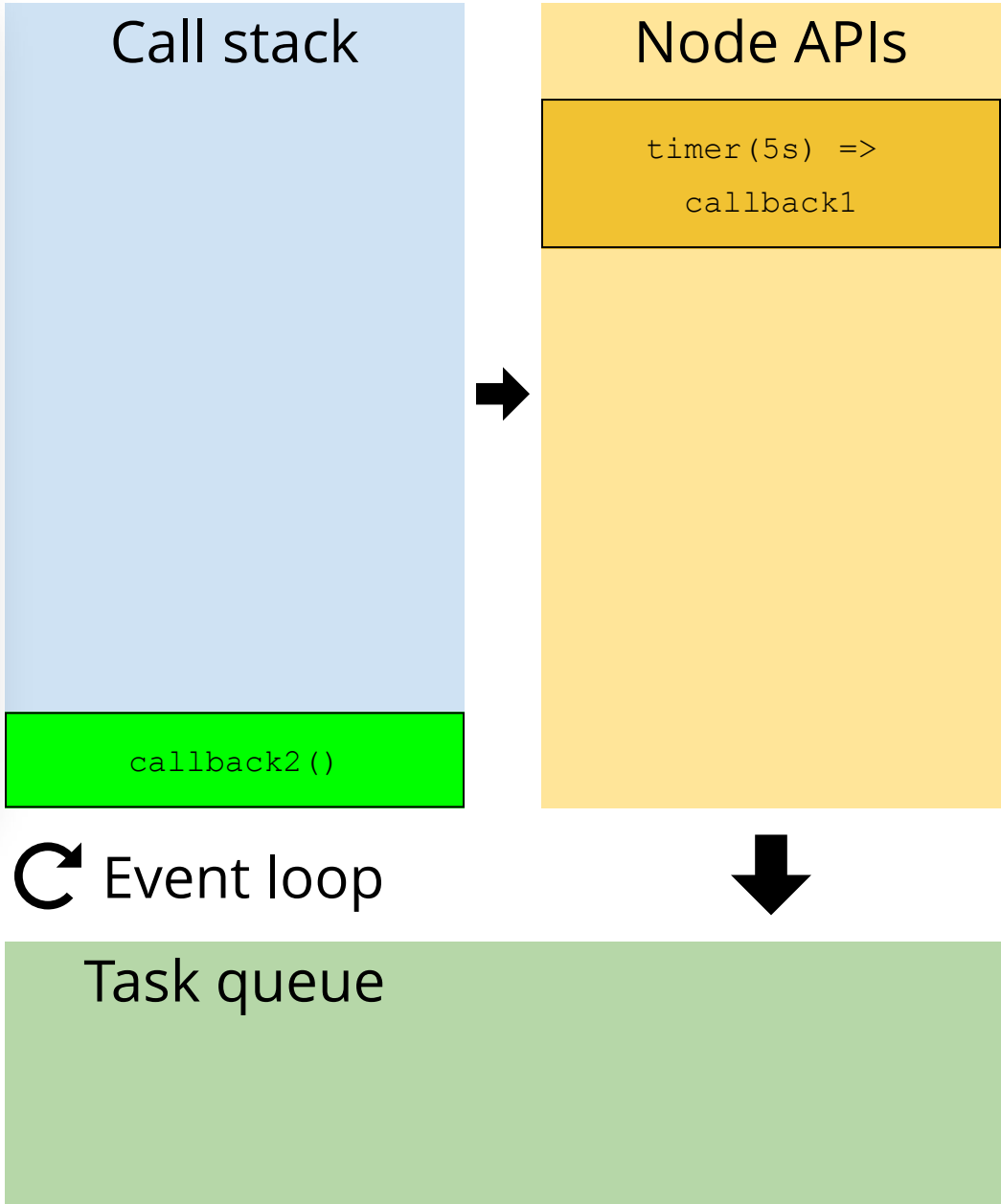
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



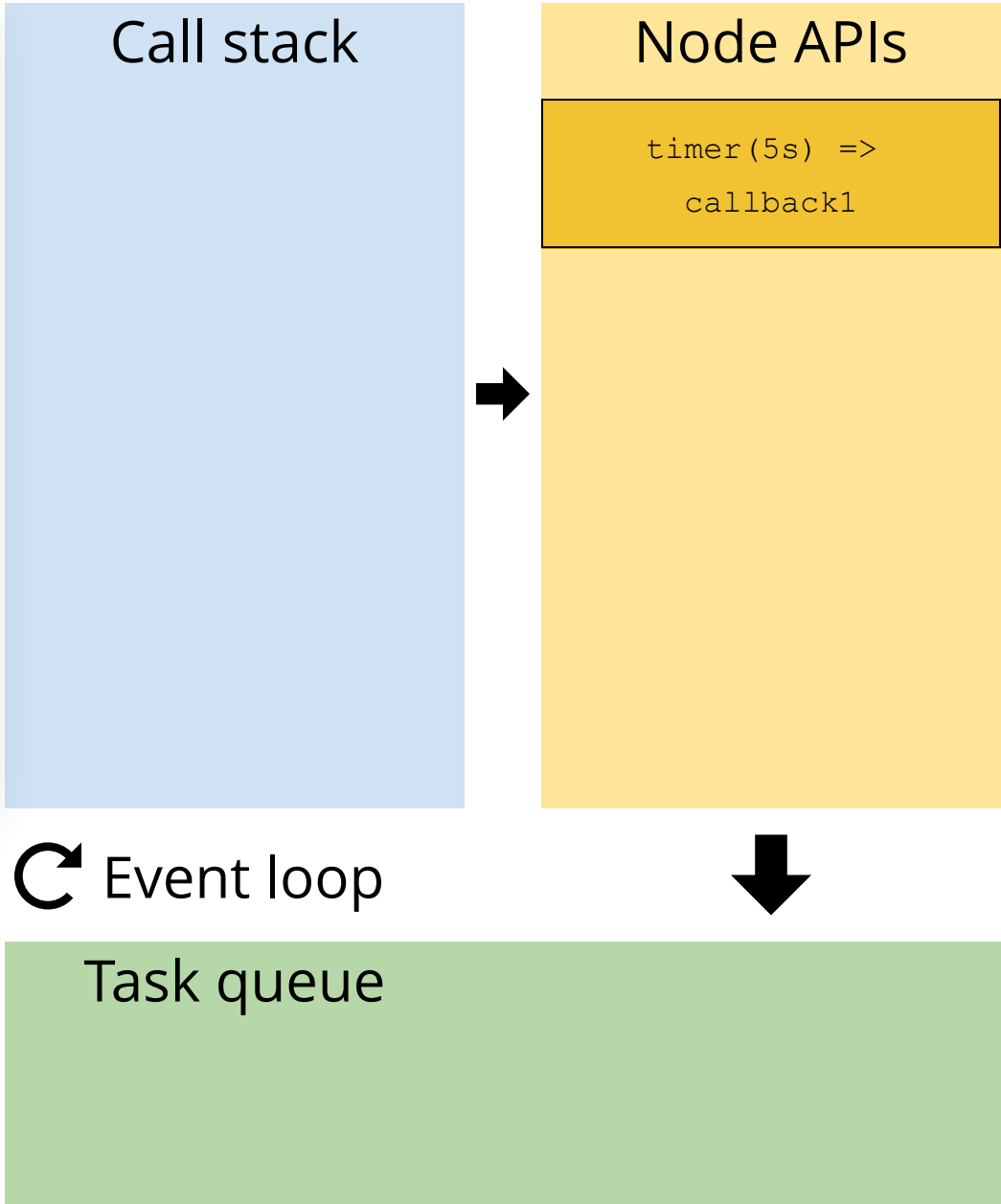
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



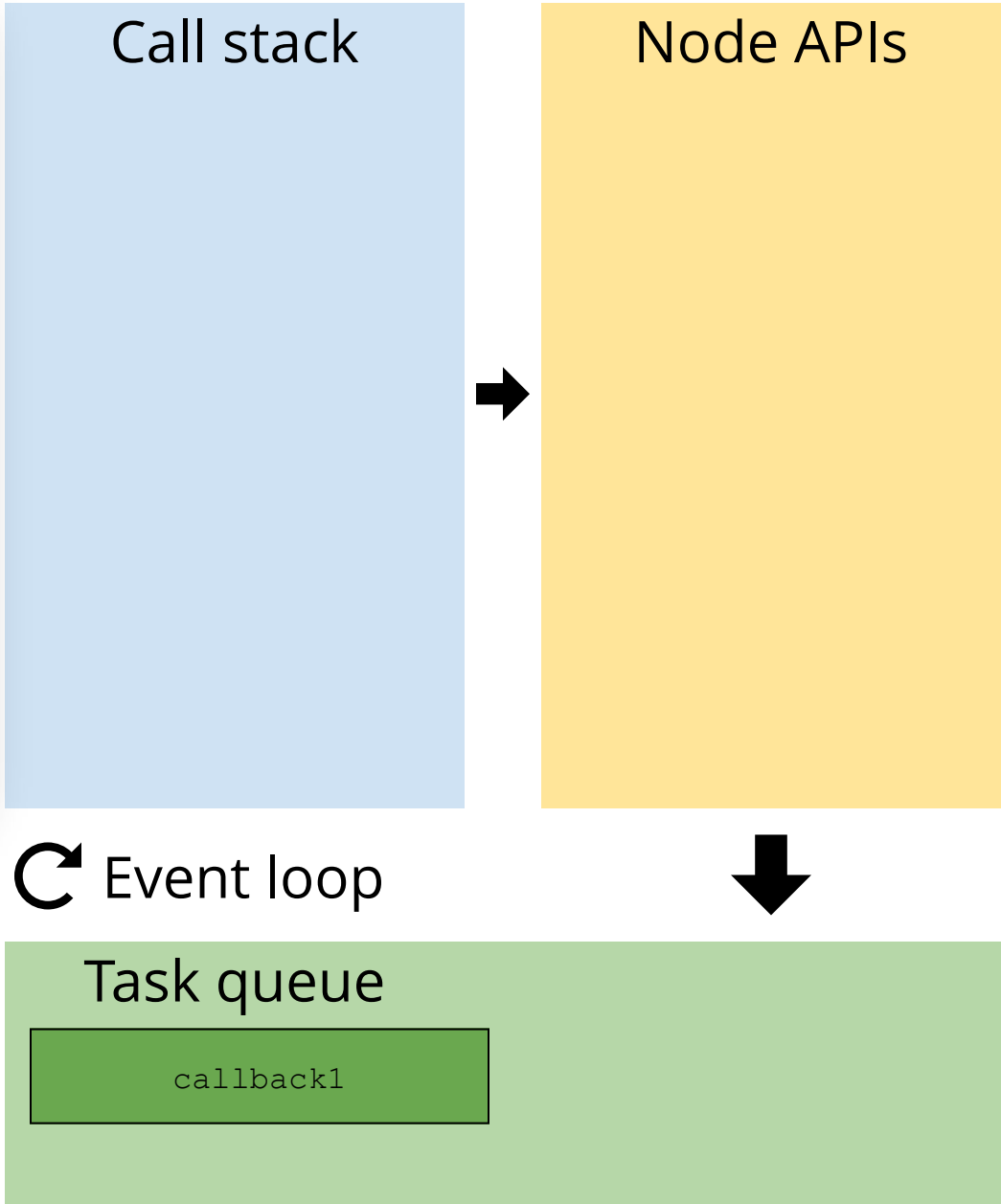
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



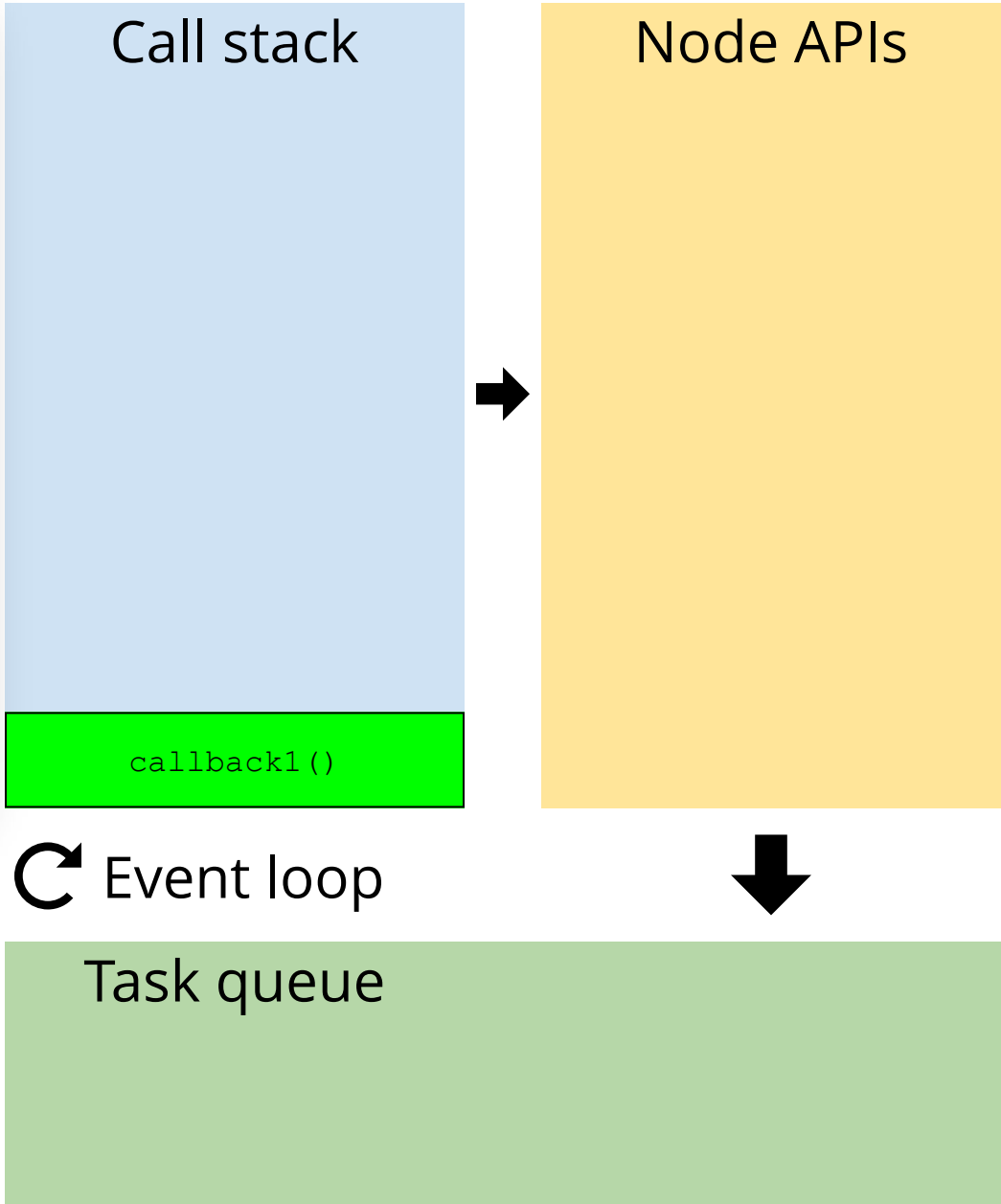
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



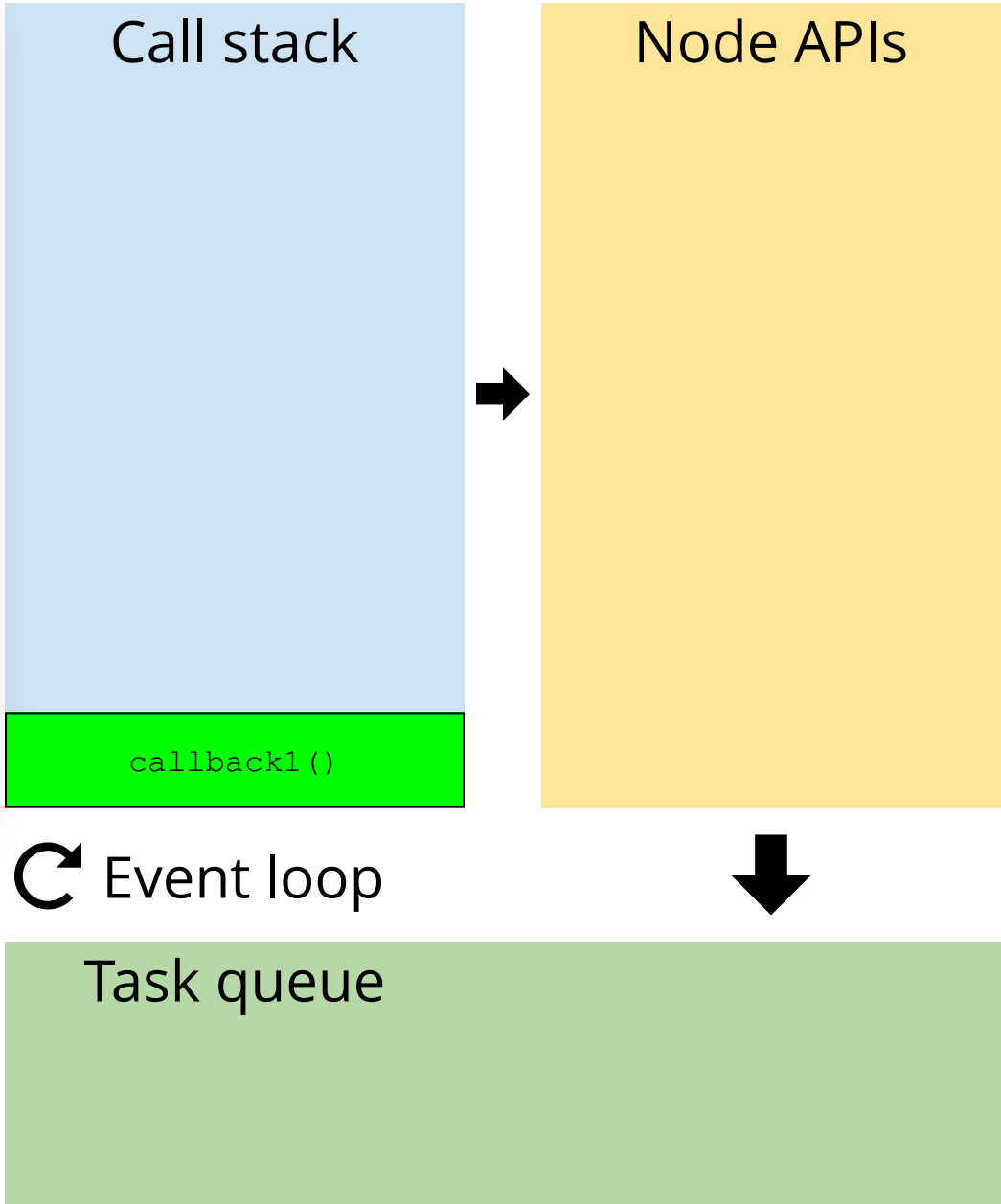
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



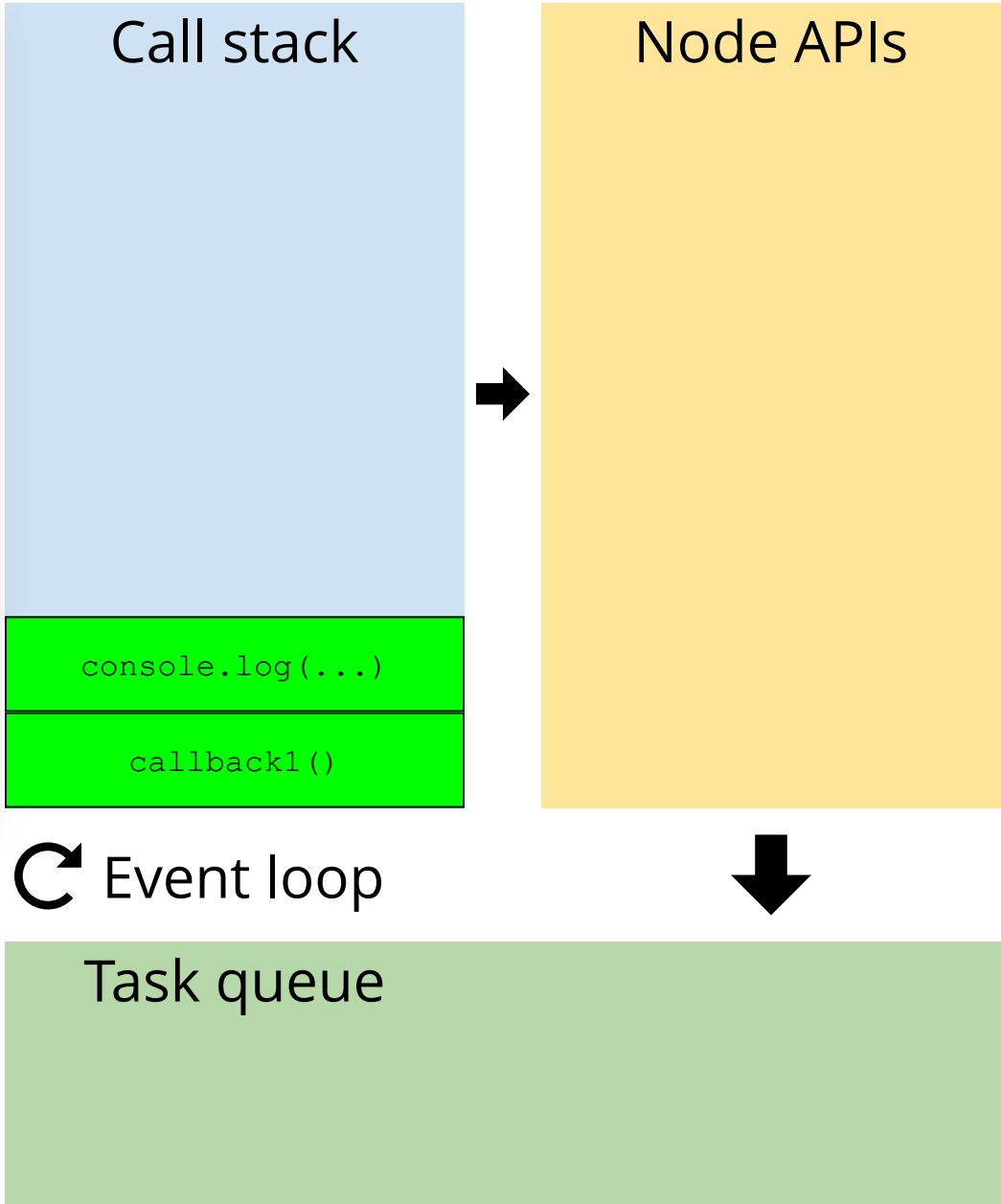
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



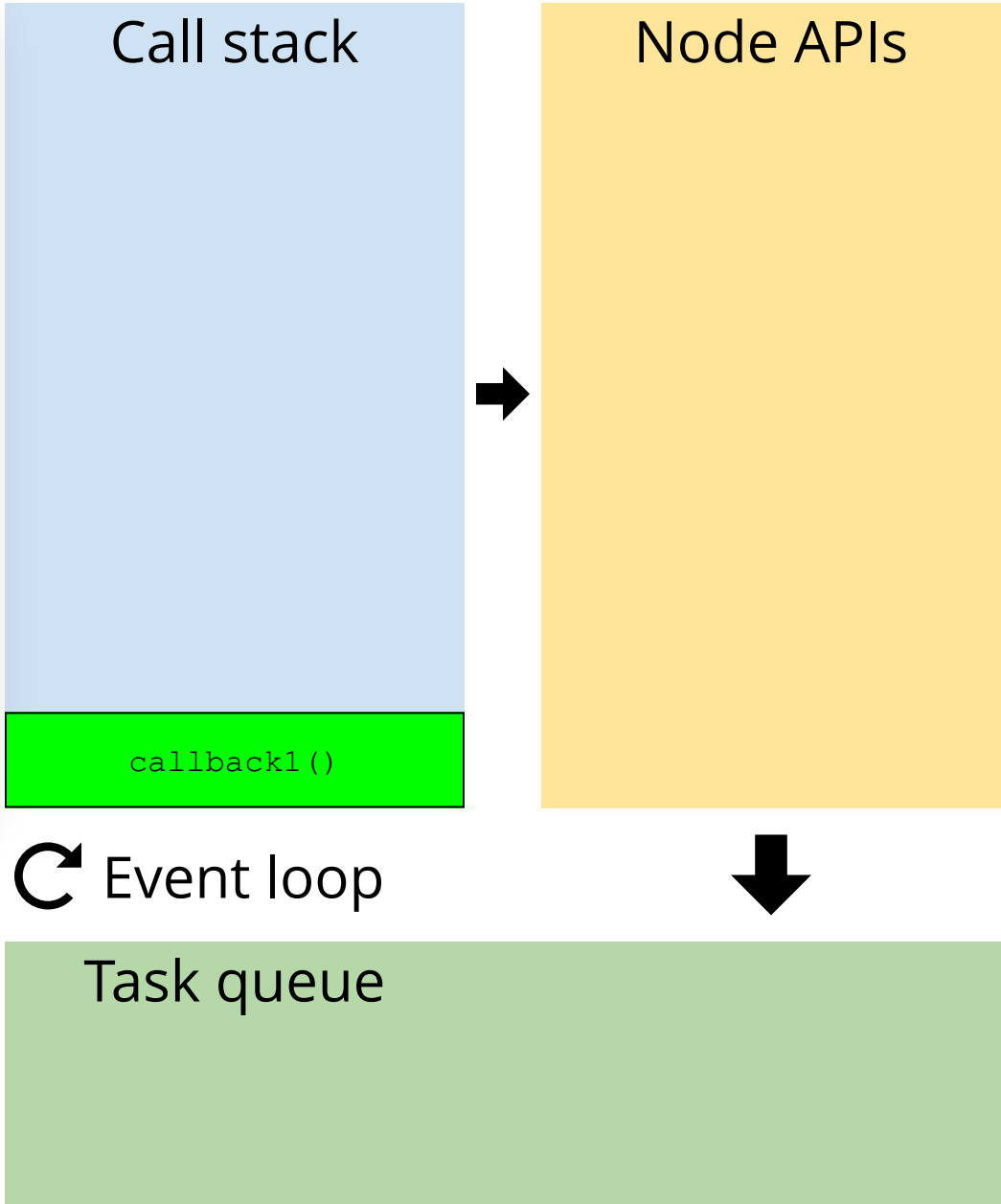
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



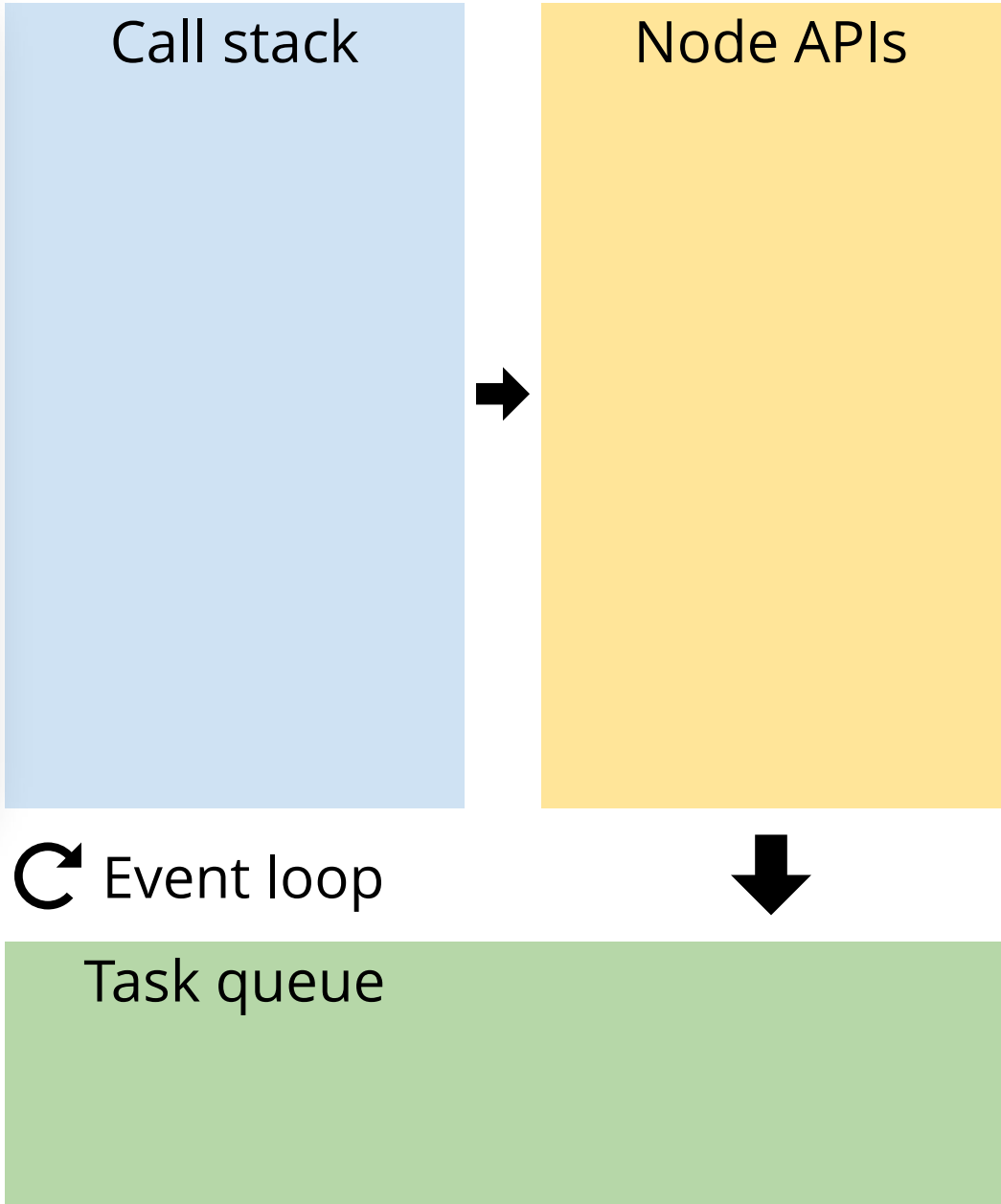
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



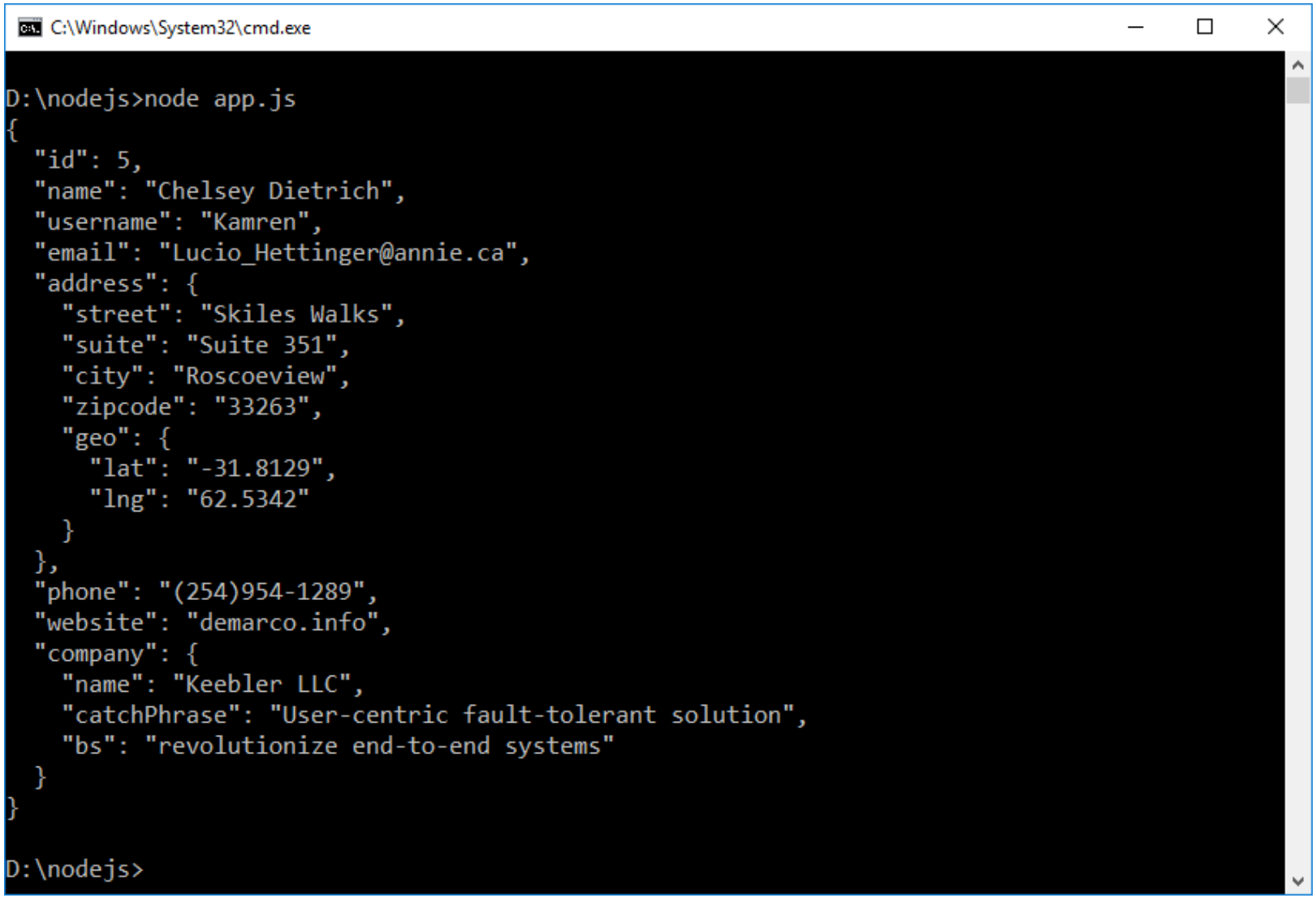
```
1  console.log('starting app');
2
3  setTimeout(function callback1() {
4
5    console.log('my function callback1')
6
7  }, 5000);
8
9  setTimeout(function callback2() {
10
11   console.log('my function callback2')
12
13 }, 0);
14
15  console.log('end app');
```



CALLBACK

```
1  const request = require('request');
2
3  function getUser(id, callback) {
4
5    const url = 'https://jsonplaceholder.typicode.com/users/' + id;
6
7    request(url, function (error, response, body) {
8      callback(body);
9    });
10
11 }
12
13
14 getUser(5, function (userInfo) {
15
16   console.log(userInfo);
17
18 });
```

https://www.npmjs.com/package/request


C:\Windows\System32\cmd.exe
D:\nodejs>node app.js
{
"id": 5,
"name": "Chelsey Dietrich",
"username": "Kamren",
"email": "Lucio_Hettinger@annie.ca",
"address": {
"street": "Skiles Walks",
"suite": "Suite 351",
"city": "Roscoeview",
"zipcode": "33263",
"geo": {
"lat": "-31.8129",
"lng": "62.5342"
}
},
"phone": "(254)954-1289",
"website": "demarco.info",
"company": {
"name": "Keebler LLC",
"catchPhrase": "User-centric fault-tolerant solution",
"bs": "revolutionize end-to-end systems"
}
}
D:\nodejs>

JSON

JavaScript Object Notation - lekki format wymiany danych komputerowych. JSON jest formatem tekstowym, bazującym na podzbiorze języka JavaScript.

Przykład formatu JSON

```json
{
    "id": 12,
    "name": "Jan",
    "age": 35,
    "fullName": "Jan Nowak"
}
```

JSON.PARSE

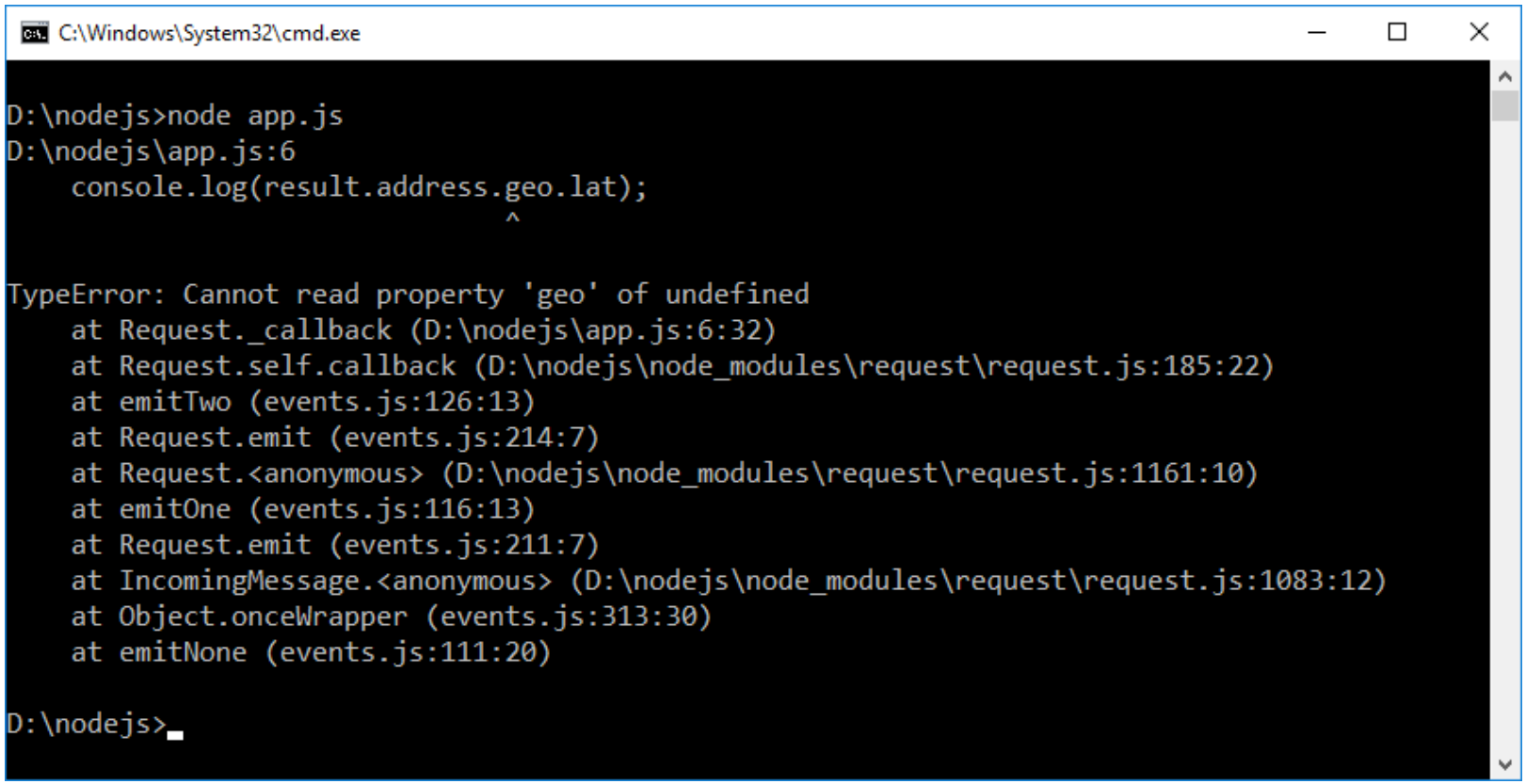
```
1 const userString = '{ "id": 12, "name": "Jan", "age": 35 }';
2 const user = JSON.parse(userString);
3
4 console.log(userString);
5 console.log(userString.id);
6 console.log(user);
7 console.log(user.id)
```

JSON.STRINGIFY

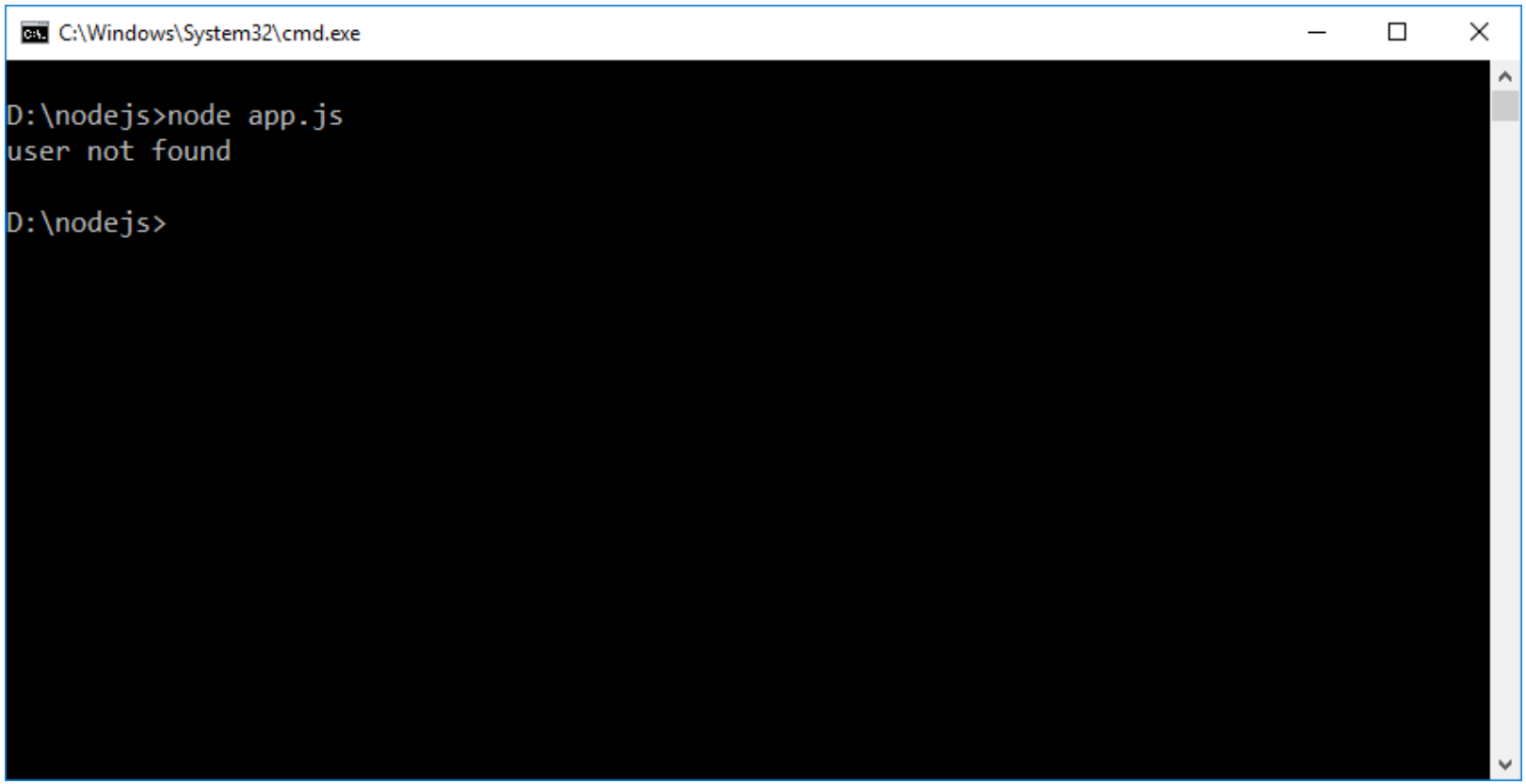
```
1  const user = {
2      id: 12,
3      name: 'Jan',
4      age: 25,
5  };
6  const userString = JSON.stringify(user);
7
8  console.log(user);
9  console.log(user.id);
10 console.log(userString);
11 console.log(userString.id);
```

CALLBACK ERRORS

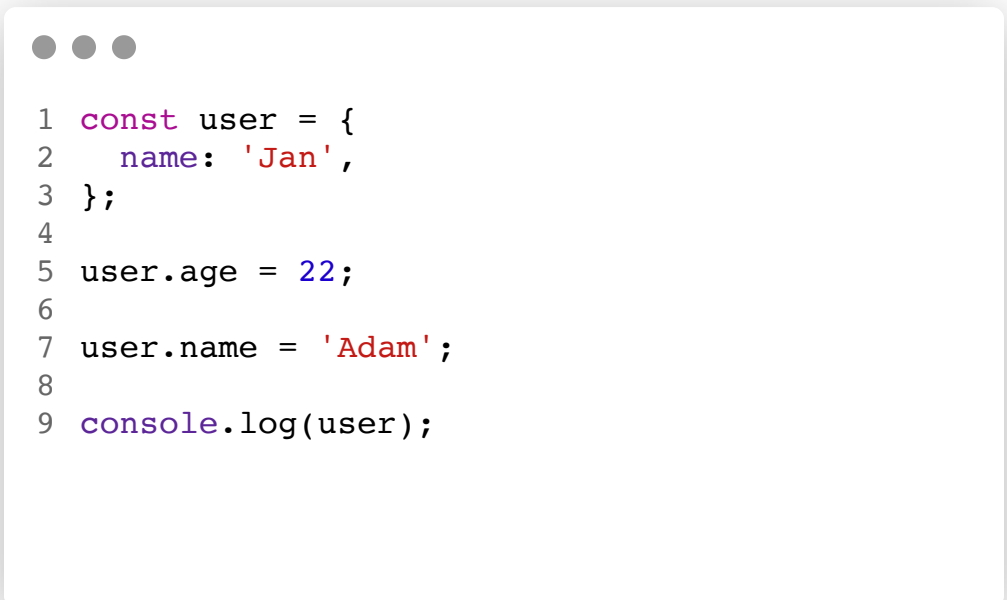
```
const request = require('request');

const url = 'https://jsonplaceholder.typicode.com/users/123';


request(url, function (error, response, body) {

  const result = JSON.parse(body);

  console.log(result.address.geo.lat);

  console.log(result.address.geo.lng);

});
```


C:\Windows\System32\cmd.exe
D:\nodejs>node app.js
D:\nodejs\app.js:6
console.log(result.address.geo.lat);
^
TypeError: Cannot read property 'geo' of undefined
at Request._callback (D:\nodejs\app.js:6:32)
at Request.self.callback (D:\nodejs\node_modules\request\request.js:185:22)
at emitTwo (events.js:126:13)
at Request.emit (events.js:214:7)
at Request.<anonymous> (D:\nodejs\node_modules\request\request.js:1161:10)
at emitOne (events.js:116:13)
at Request.emit (events.js:211:7)
at IncomingMessage.<anonymous> (D:\nodejs\node_modules\request\request.js:1083:12)
at Object.onceWrapper (events.js:313:30)
at emitNone (events.js:111:20)
D:\nodejs>

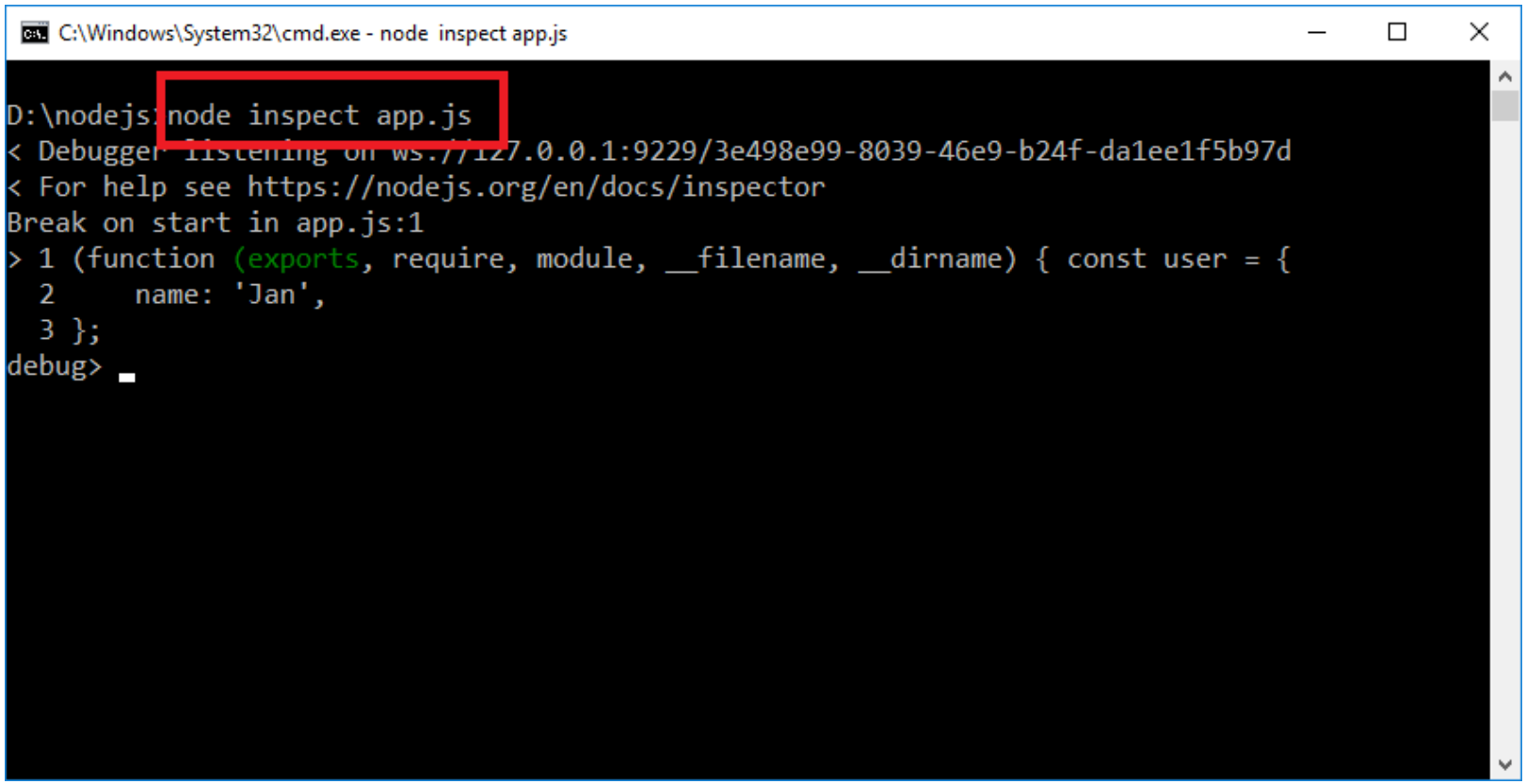
```
const request = require('request');

const url = 'https://jsonplaceholder.typicode.com/users/123';

request(url, function (error, response, body) {

    if (error) {

        console.log('problem with api');

    } else if (response.statusCode === 200) {

        const result = JSON.parse(body);
        console.log(result.address.geo.lat);
        console.log(result.address.geo.lng);

    } else {

        console.log('user not found');

    }
});
```

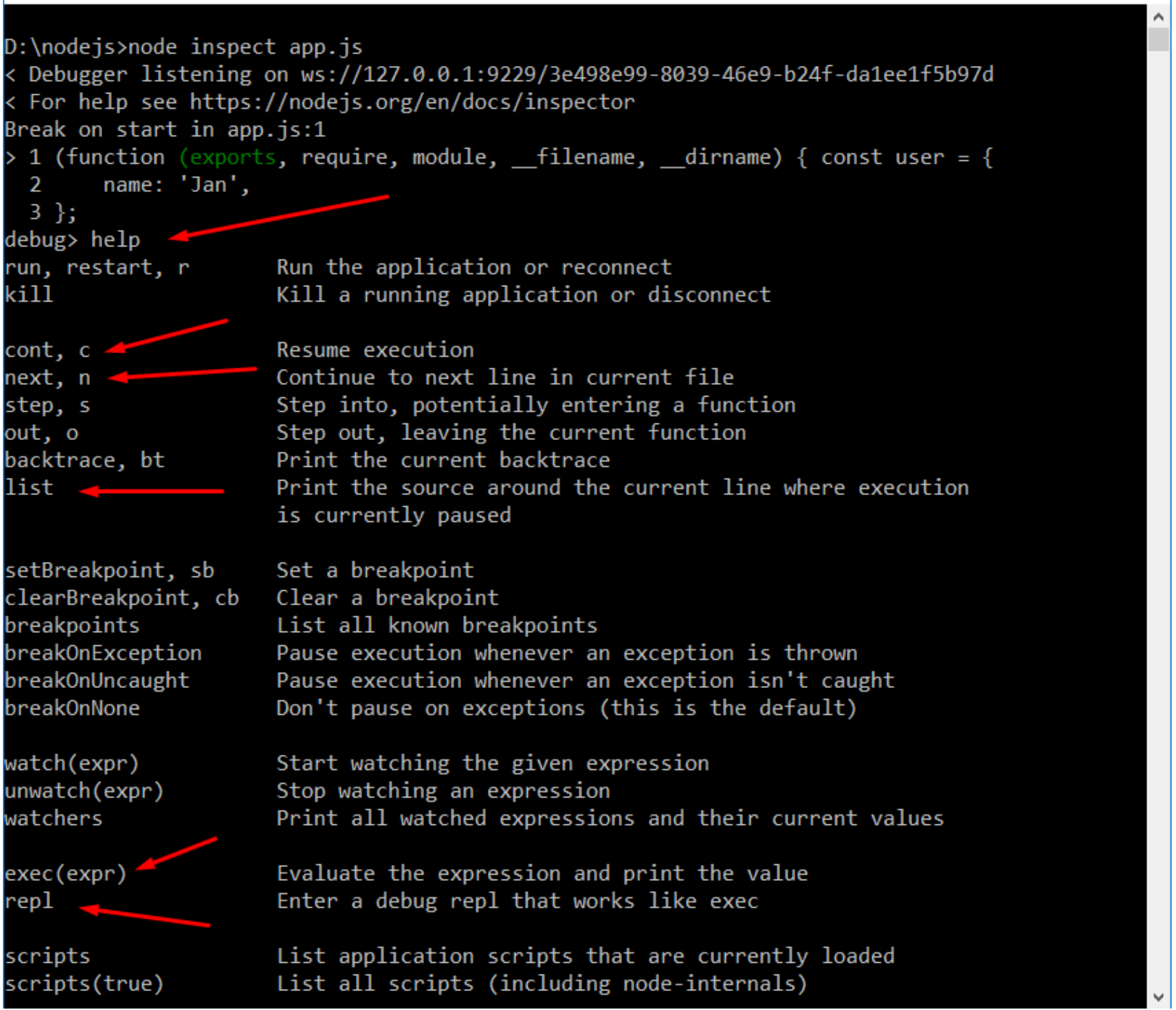

C:\Windows\System32\cmd.exe
D:\nodejs>node app.js
user not found
D:\nodejs>

DEBUGGING NODEJS



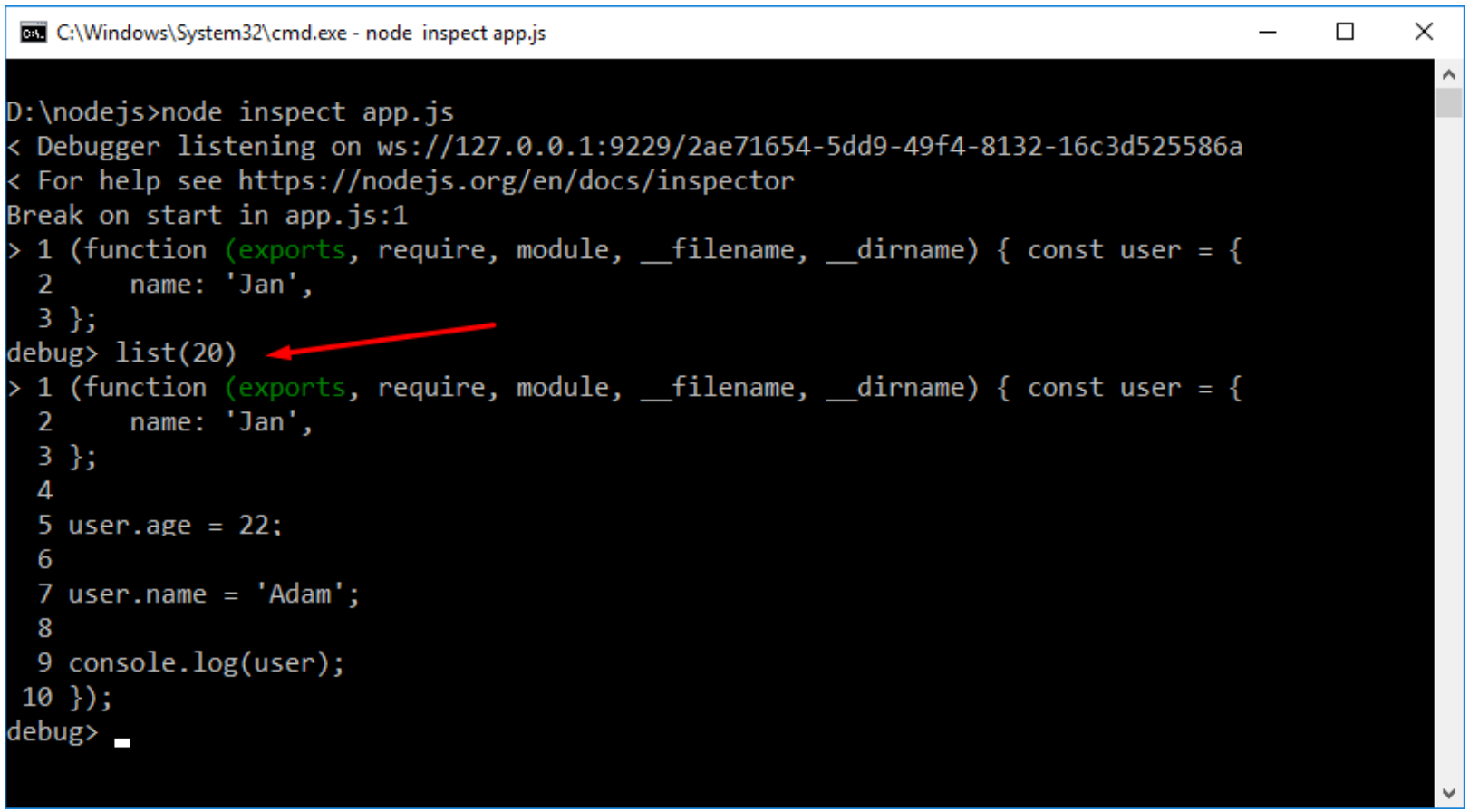
```
const user = {
  name: 'Jan',
};

user.age = 22;

user.name = 'Adam';

console.log(user);
```

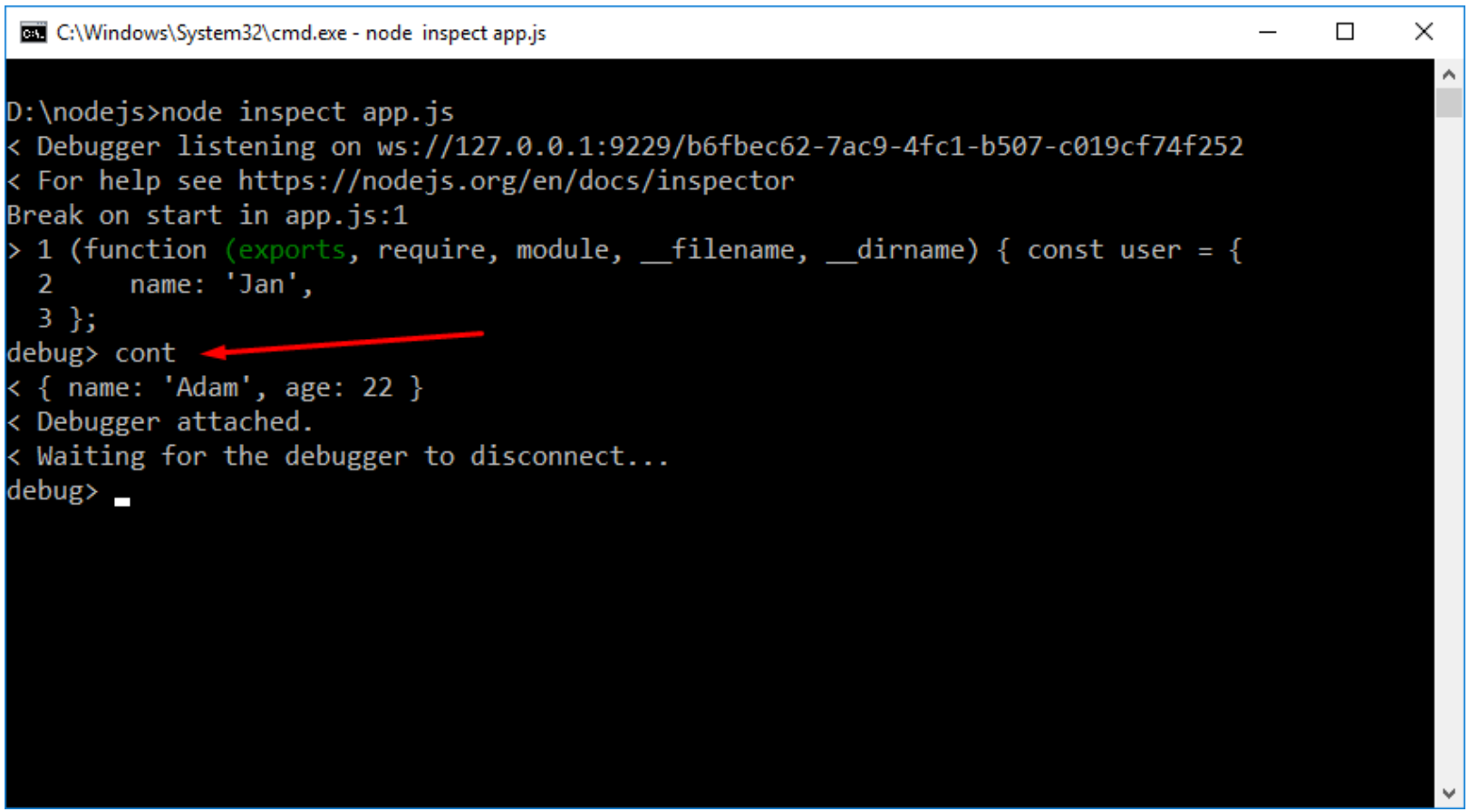

C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/3e498e99-8039-46e9-b24f-da1ee1f5b97d
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug>

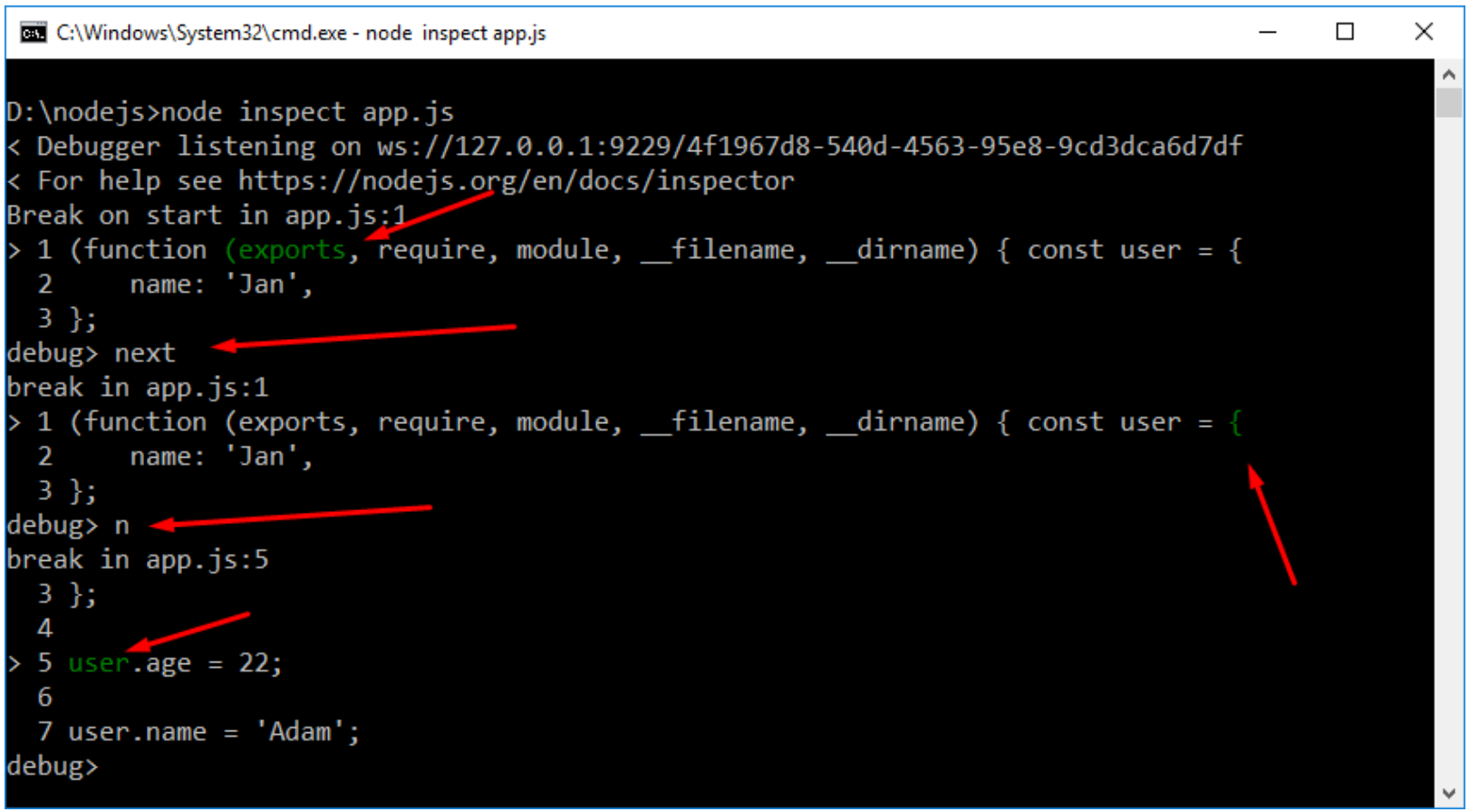

D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/3e498e99-8039-46e9-b24f-da1ee1f5b97d
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> help
run, restart, r Run the application or reconnect
kill Kill a running application or disconnect
cont, c Resume execution
next, n Continue to next line in current file
step, s Step into, potentially entering a function
out, o Step out, leaving the current function
backtrace, bt Print the current backtrace
list Print the source around the current line where execution
is currently paused
setBreakpoint, sb Set a breakpoint
clearBreakpoint, cb Clear a breakpoint
breakpoints List all known breakpoints
breakOnException Pause execution whenever an exception is thrown
breakOnUncaught Pause execution whenever an exception isn't caught
breakOnNone Don't pause on exceptions (this is the default)
watch(expr) Start watching the given expression
unwatch(expr) Stop watching an expression
watchers Print all watched expressions and their current values
exec(expr) Evaluate the expression and print the value
repl Enter a debug repl that works like exec
scripts List application scripts that are currently loaded
scripts(true) List all scripts (including node-internals)

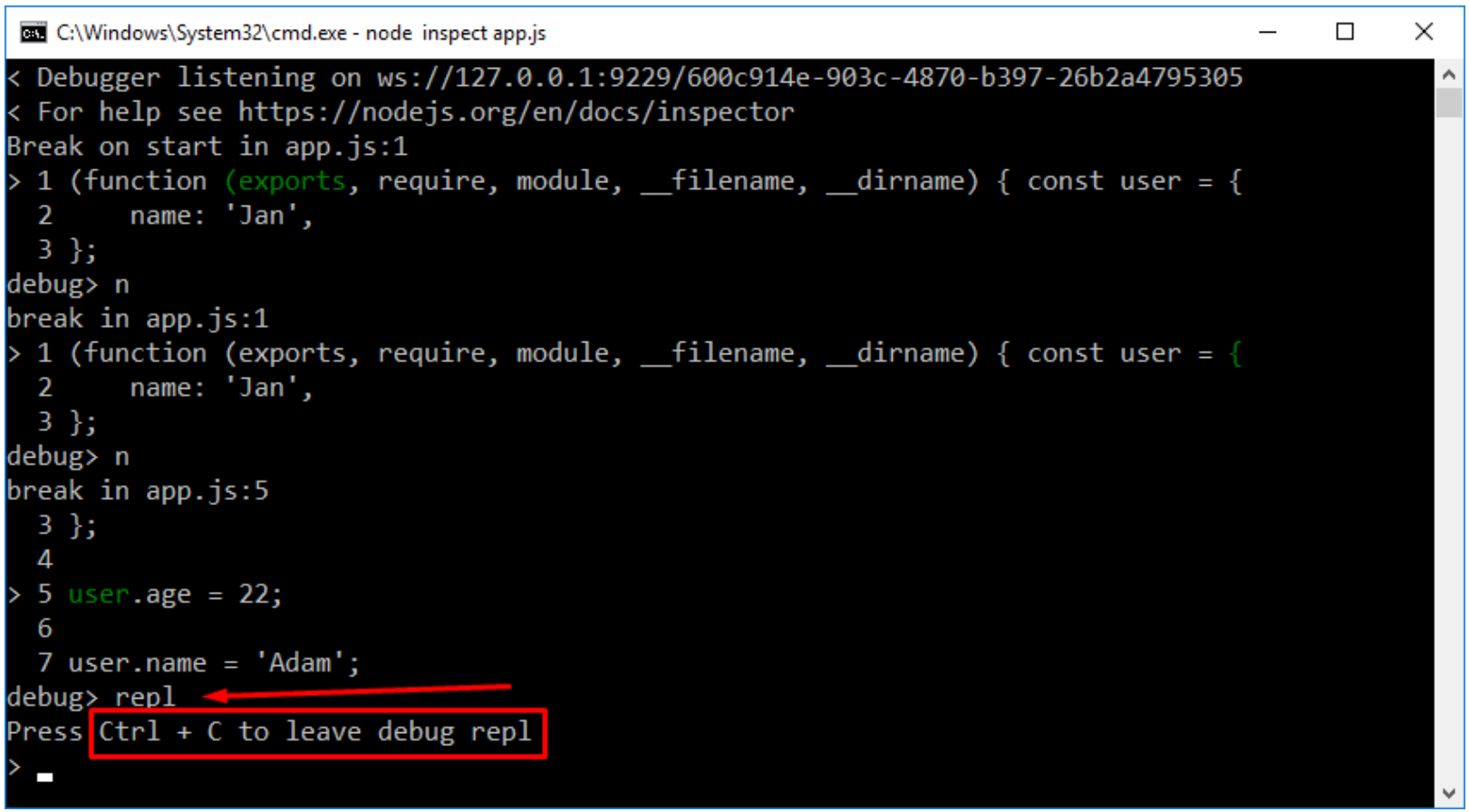
REPL

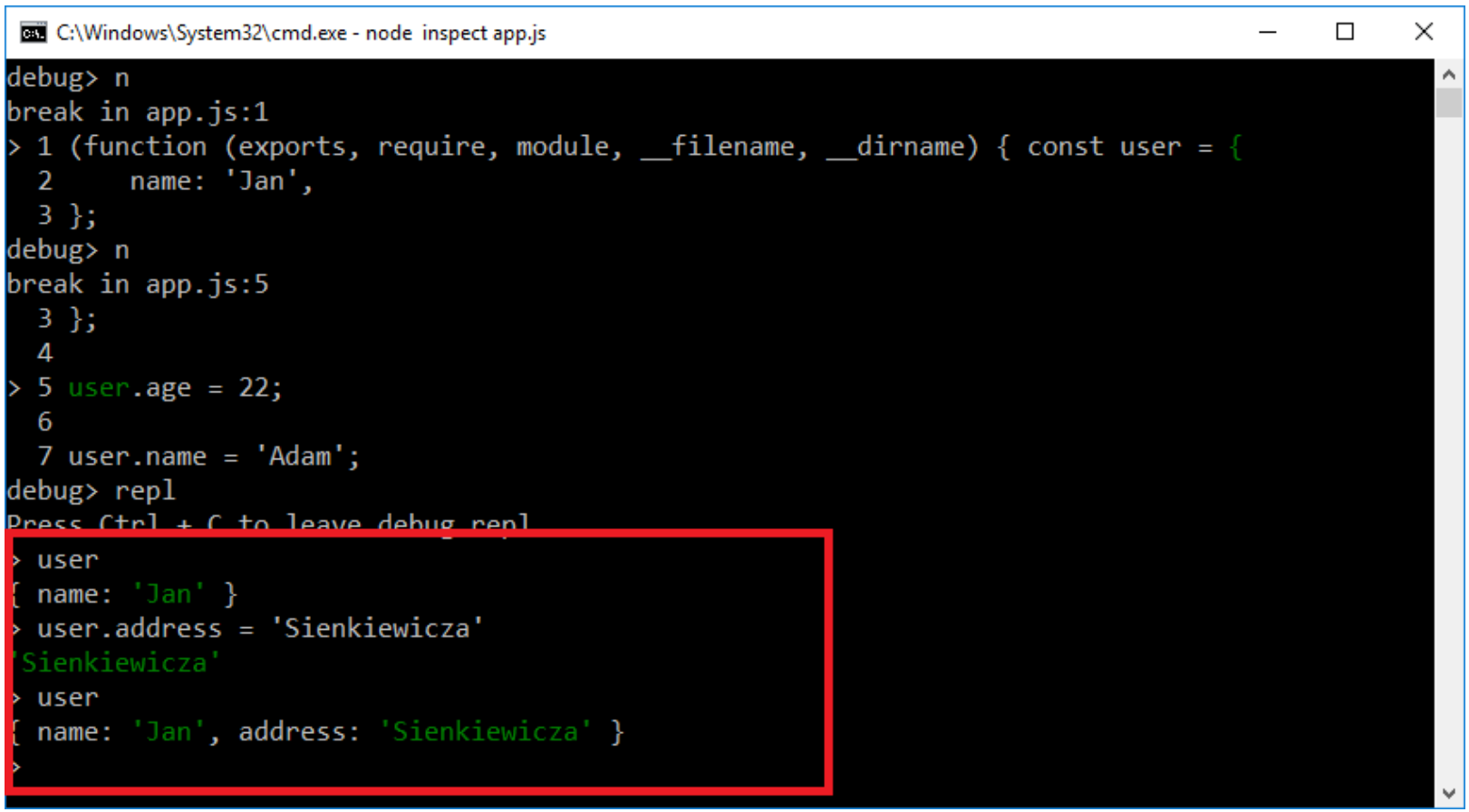
REPL (ang. read-eval-print loop) - proste, interaktywne środowisko programowania. Praktycznie każdy język uruchamiany jako interpreter posiada REPL. Dzięki REPL użytkownik może wprowadzać polecenia, które zostaną wykonane a ich wynik wypisany na ekran.


C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/2ae71654-5dd9-49f4-8132-16c3d525586a
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> list(20)
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
4
5 user.age = 22;
6
7 user.name = 'Adam';
8
9 console.log(user);
10 });
debug>


C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/b6fbec62-7ac9-4fc1-b507-c019cf74f252
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> cont
< { name: 'Adam', age: 22 }
< Debugger attached.
< Waiting for the debugger to disconnect...
debug>


C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/4f1967d8-540d-4563-95e8-9cd3dca6d7df
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> next
break in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> n
break in app.js:5
3 };
4
> 5 user.age = 22;
6
7 user.name = 'Adam';
debug>

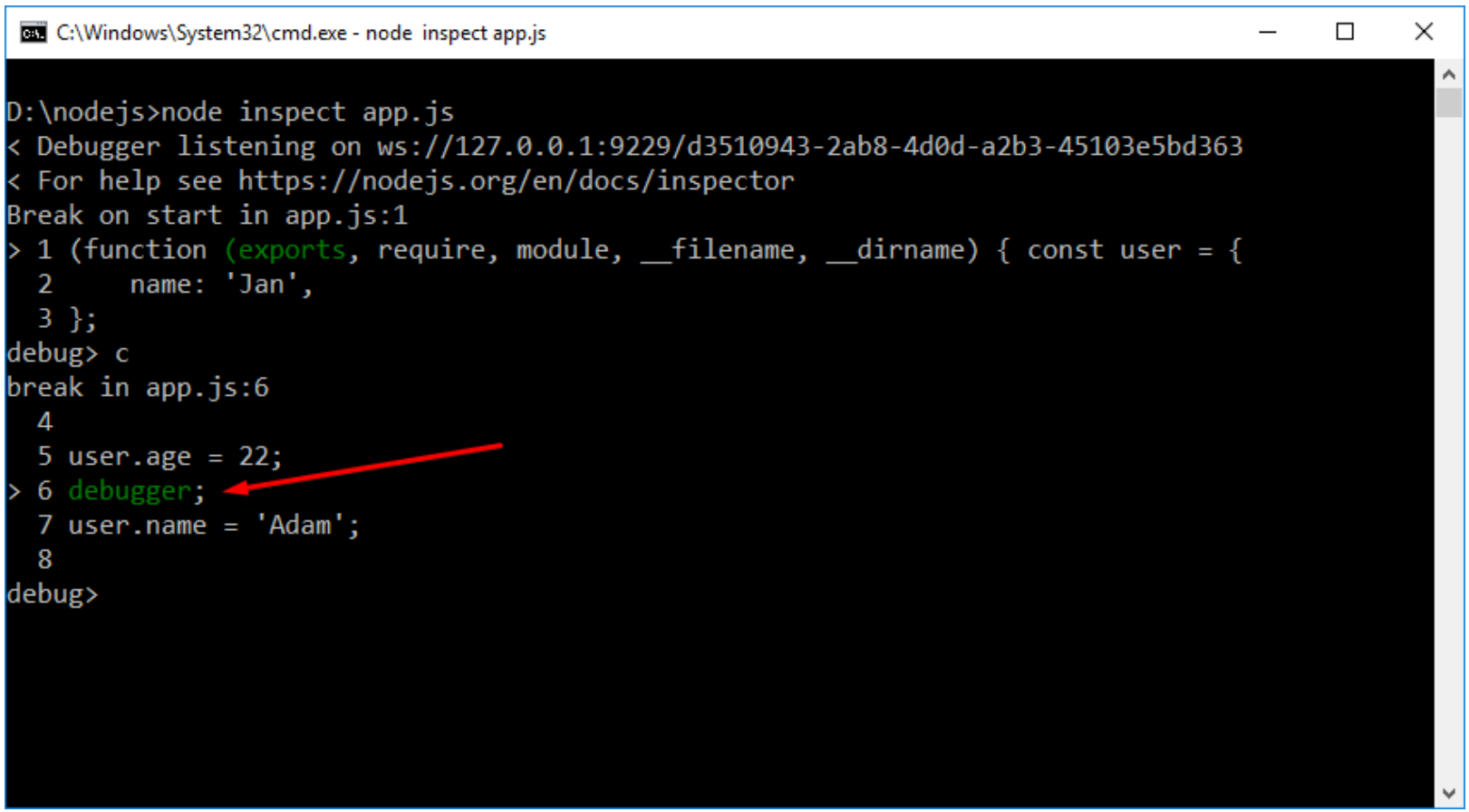

C:\Windows\System32\cmd.exe - node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/600c914e-903c-4870-b397-26b2a4795305
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> n
break in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> n
break in app.js:5
3 };
4
> 5 user.age = 22;
6
7 user.name = 'Adam';
debug> repl
Press Ctrl + C to leave debug repl
>

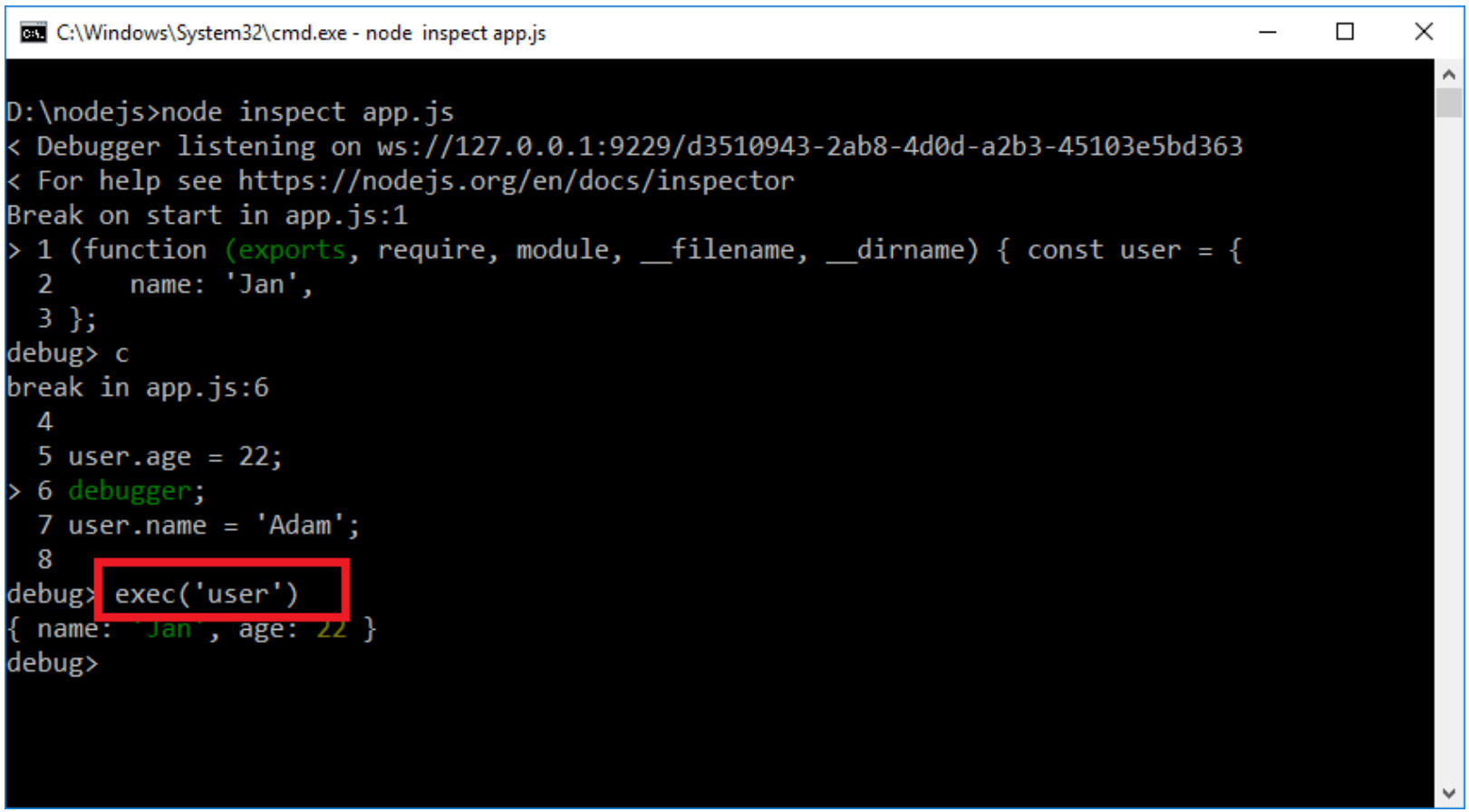

C:\Windows\System32\cmd.exe - node inspect app.js
debug> n
break in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> n
break in app.js:5
3 };
4
> 5 user.age = 22;
6
7 user.name = 'Adam';
debug> repl
Press Ctrl + C to leave debug repl
> user
{ name: 'Jan' }
> user.address = 'Sienkiewicza'
'Sienkiewicza'
> user
{ name: 'Jan', address: 'Sienkiewicza' }
>


C:\Windows\System32\cmd.exe - node inspect app.js
'Sienkiewicza'
> user
{ name: 'Jan', address: 'Sienkiewicza' }
debug> n
break in app.js:7
5 user.age = 22;
6
> 7 user.name = 'Adam';
8
9 console.log(user);
debug> n
break in app.js:9
7 user.name = 'Adam';
8
> 9 console.log(user);
10 });
debug> c
< { name: 'Adam', address: 'Sienkiewicza', age: 22 }
< Debugger attached.
< Waiting for the debugger to disconnect...
debug>

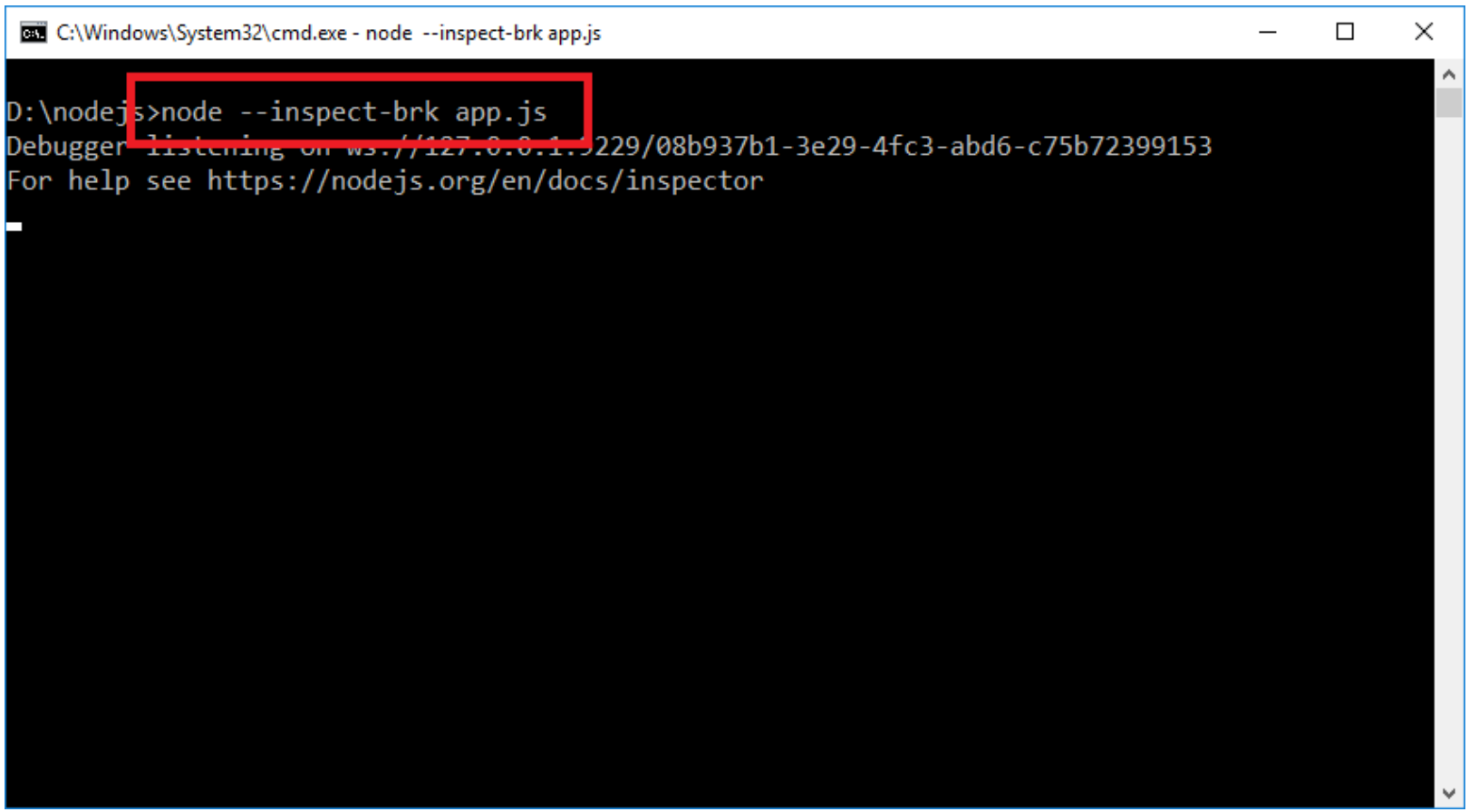
DEBUGGER;

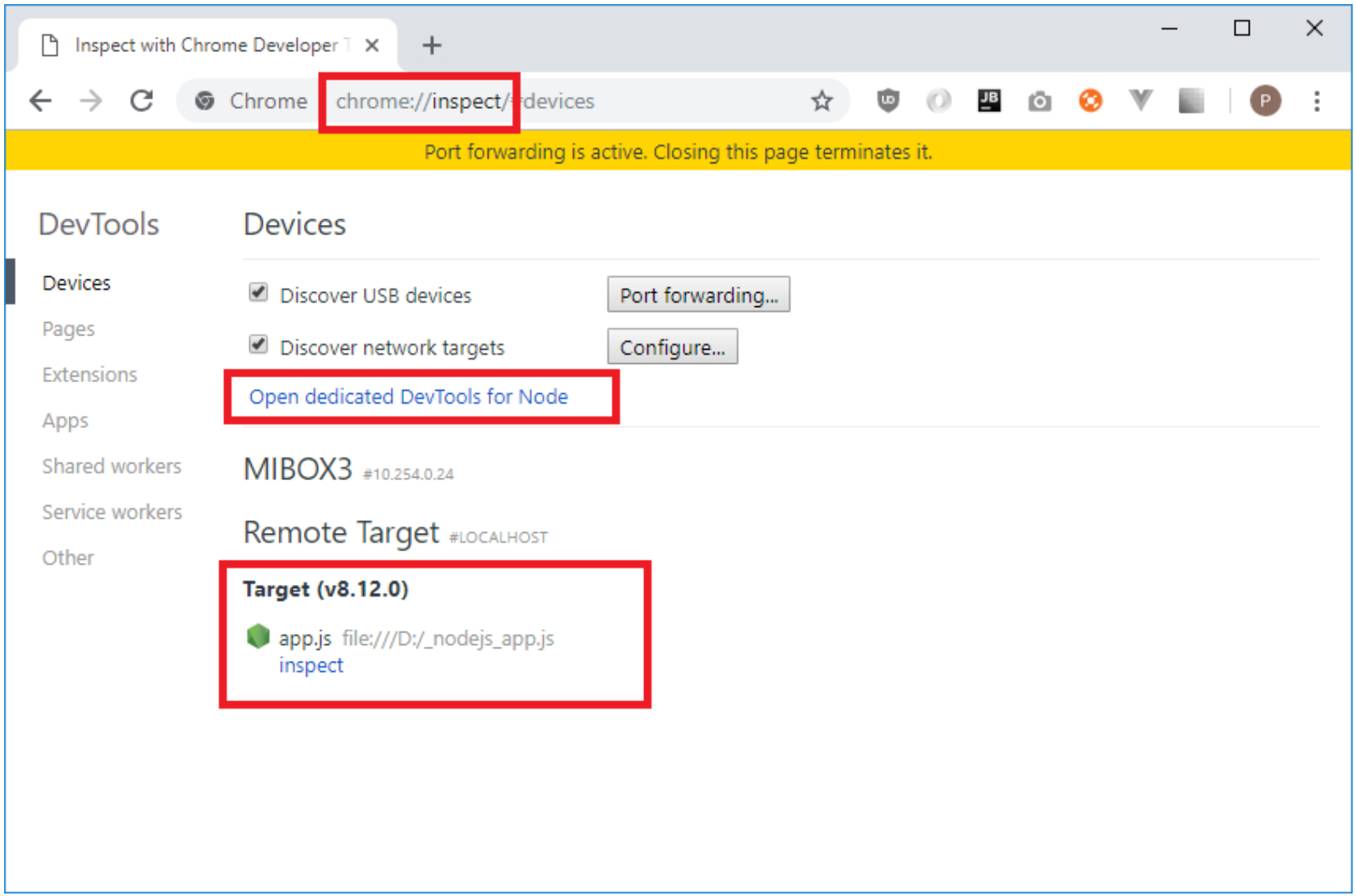
```
1  const user = {
2    name: 'Jan',
3  };
4
5  user.age = 22;
6
7  debugger;
8
9  user.name = 'Adam';
10
11 console.log(user);
```

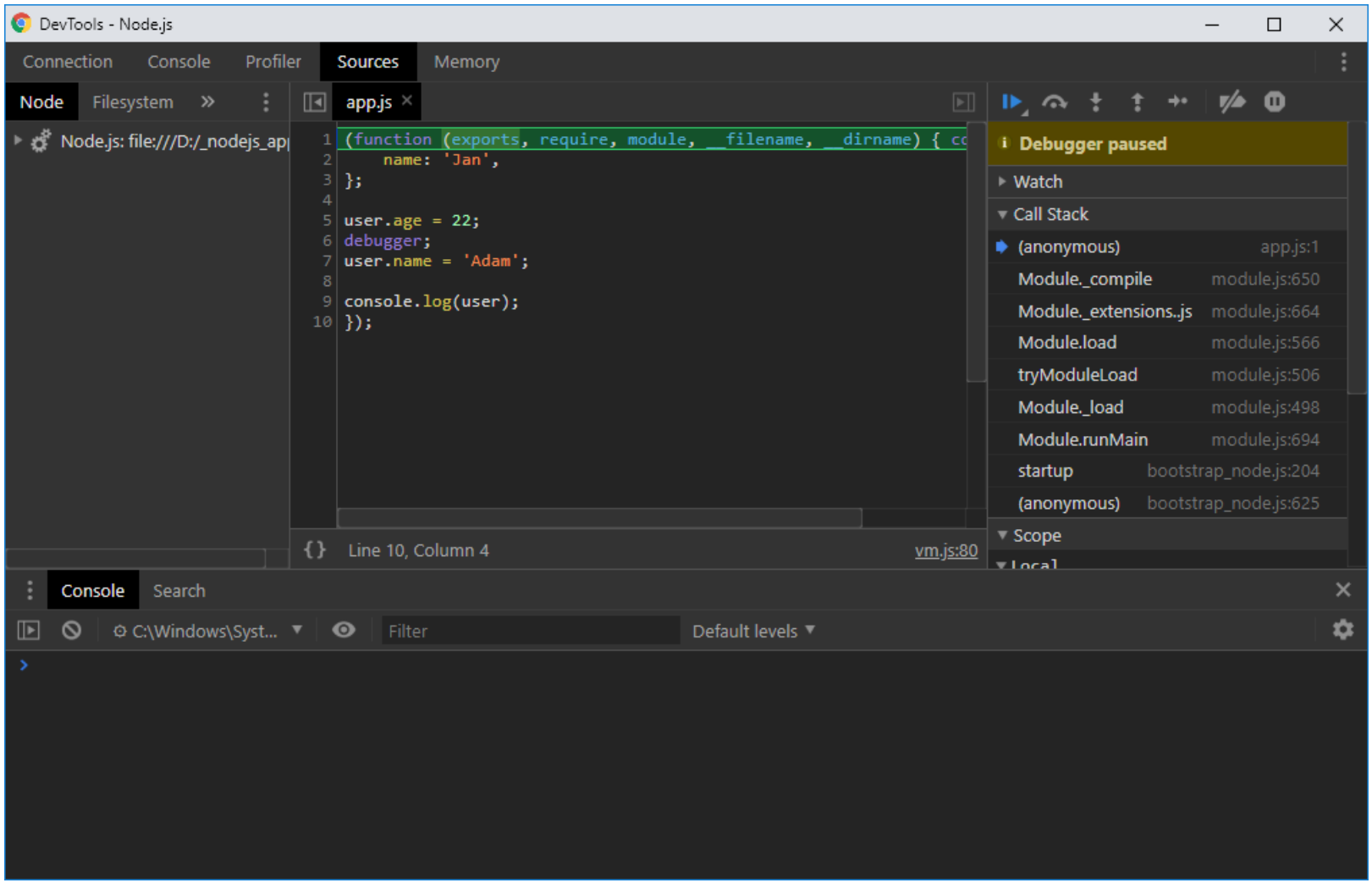

C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/d3510943-2ab8-4d0d-a2b3-45103e5bd363
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> c
break in app.js:6
4
5 user.age = 22;
> 6 debugger;
7 user.name = 'Adam';
8
debug>

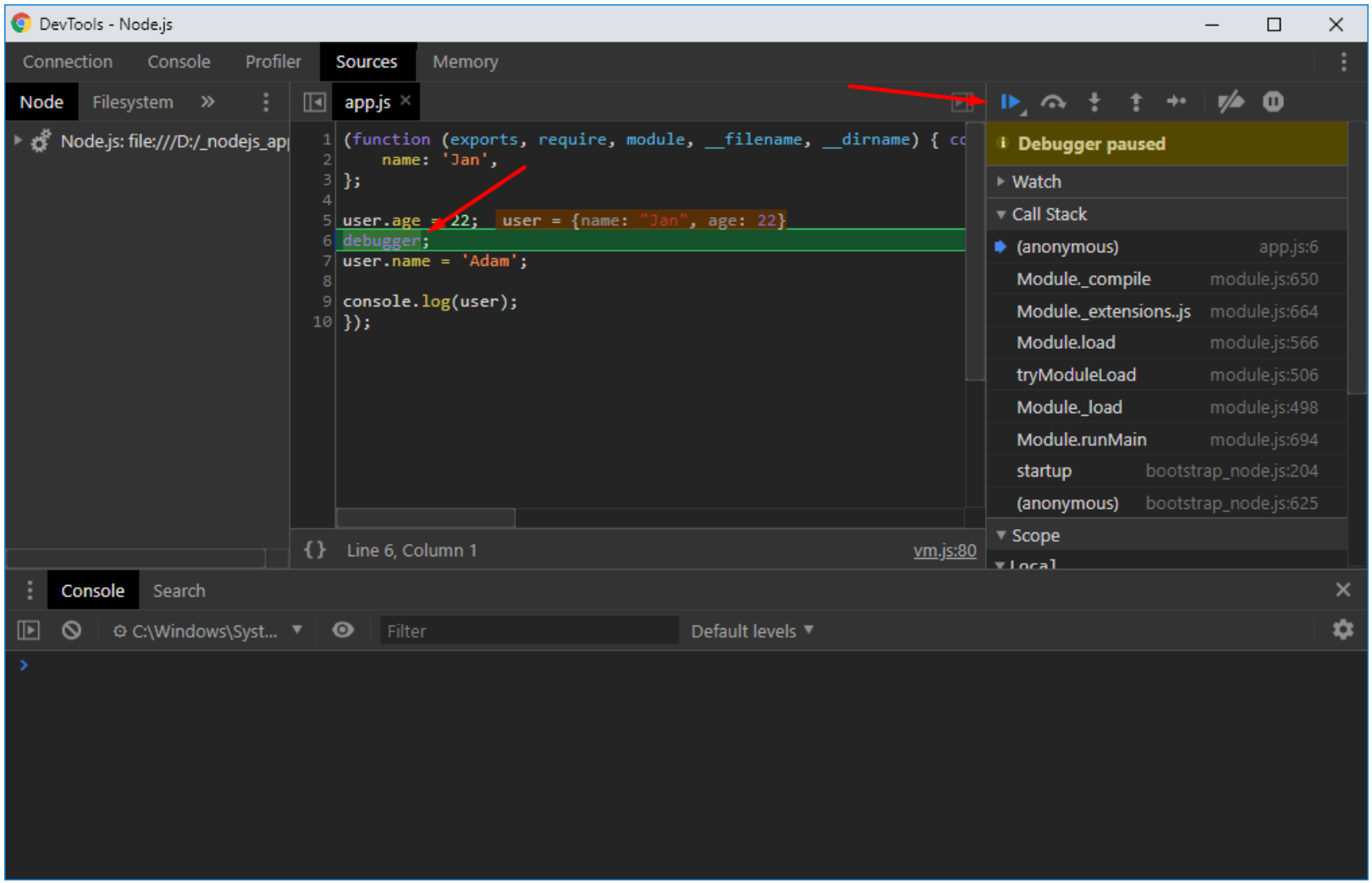

C:\Windows\System32\cmd.exe - node inspect app.js
D:\nodejs>node inspect app.js
< Debugger listening on ws://127.0.0.1:9229/d3510943-2ab8-4d0d-a2b3-45103e5bd363
< For help see https://nodejs.org/en/docs/inspector
Break on start in app.js:1
> 1 (function (exports, require, module, __filename, __dirname) { const user = {
2 name: 'Jan',
3 };
debug> c
break in app.js:6
4
5 user.age = 22;
> 6 debugger;
7 user.name = 'Adam';
8
debug> exec('user')
{ name: 'Jan', age: 22 }
debug>

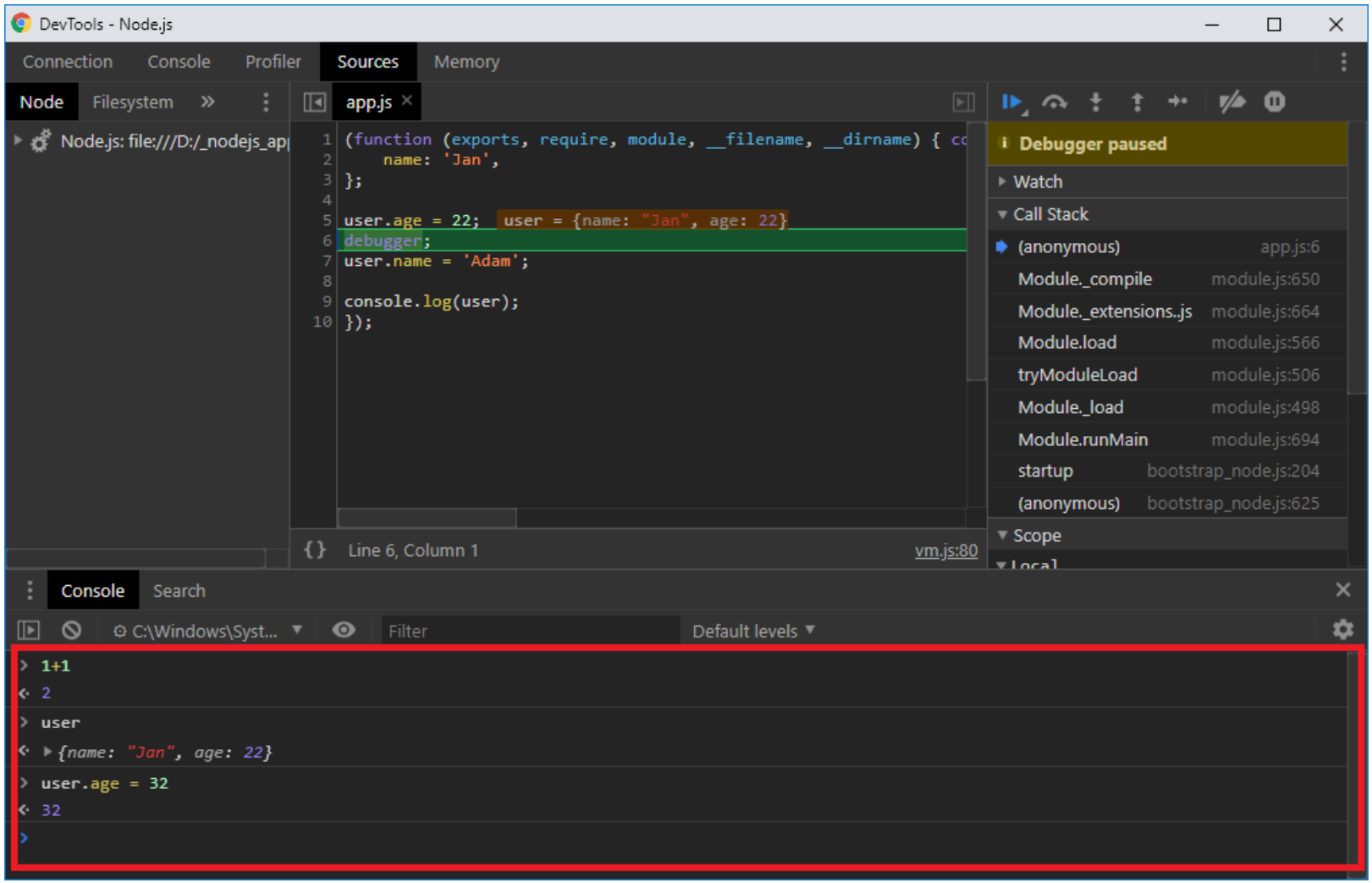
DEBUGGING IN BROWSER


C:\Windows\System32\cmd.exe - node --inspect-brk app.js
D:\nodejs>node --inspect-brk app.js
Debugger listening on ws://127.0.0.1:9229/08b937b1-3e29-4fc3-abd6-c75b72399153
For help see https://nodejs.org/en/docs/inspector

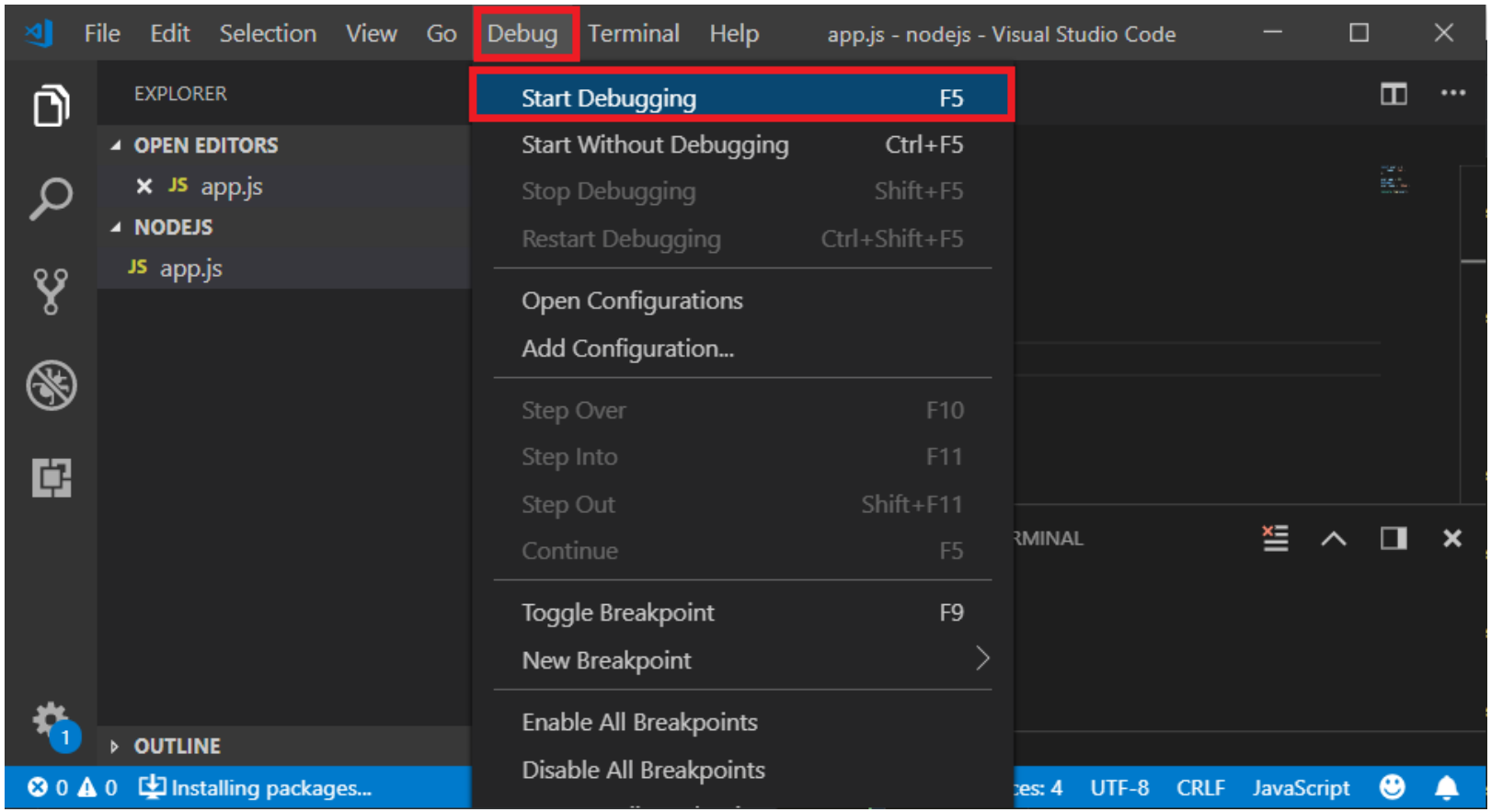

Inspect with Chrome Developer T
Chrome
chrome://inspect/#devices
Port forwarding is active. Closing this page terminates it.
DevTools
Devices
Pages
Extensions
Apps
Shared workers
Service workers
Other
Devices
Discover USB devices
Port forwarding...
Discover network targets
Configure...
Open dedicated DevTools for Node
MIBOX3 #10.254.0.24
Remote Target #LOCALHOST
Target (v8.12.0)
app.js file:///D:/_nodejs_app.js
inspect

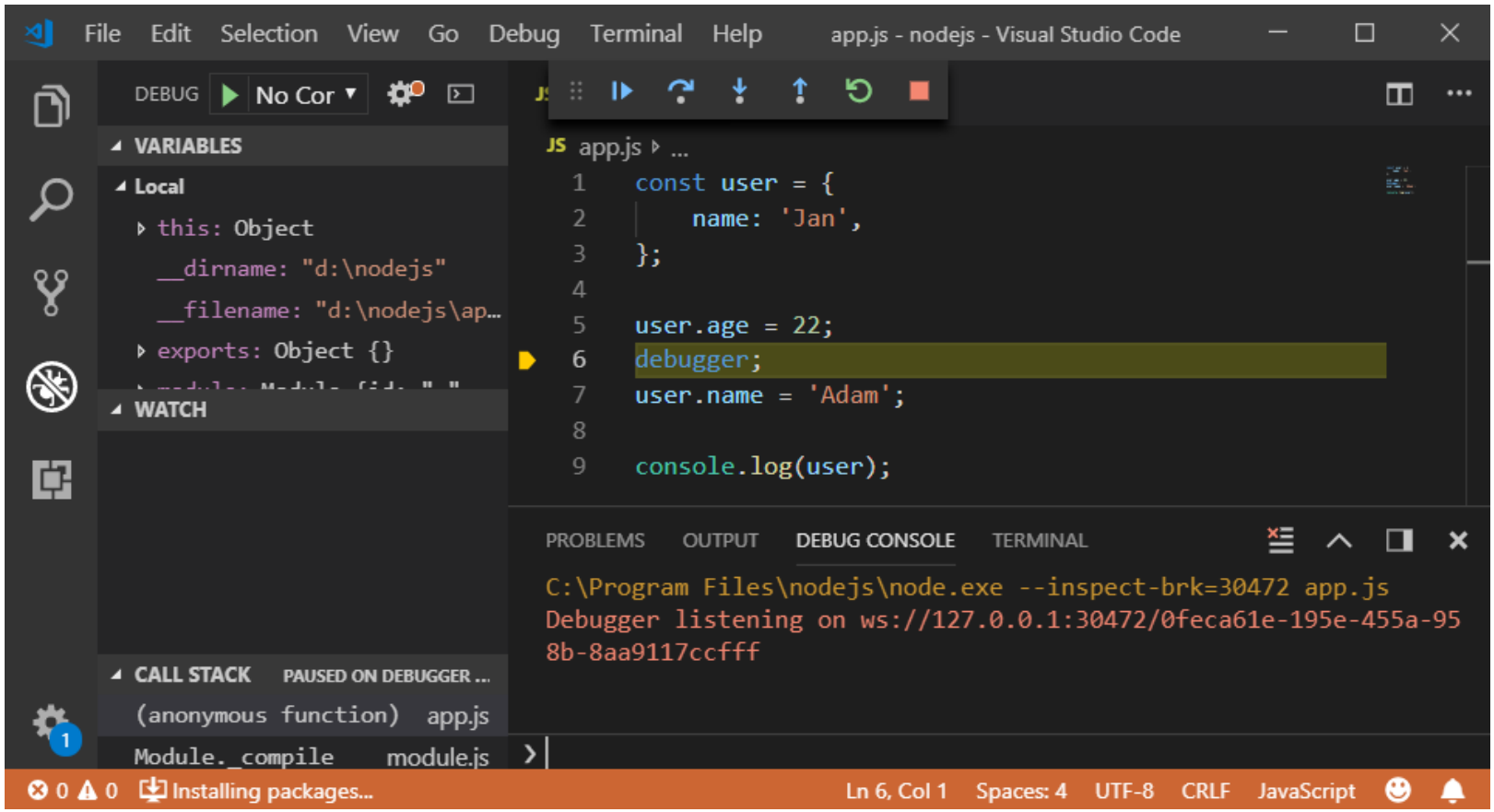

DevTools - Node.js
Connection
Console
Profiler
Sources
Memory
Node
Filesystem
Node.js: file:///D:/_nodejs_ap
app.js
(function (exports, require, module, __filename, __dirname) { cc
name: 'Jan',
};
user.age = 22;
debugger;
user.name = 'Adam';
console.log(user);
});
Line 10, Column 4
vm.js:80
Debugger paused
Watch
Call Stack
(anonymous) app.js:1
Module._compile module.js:650
Module._extensions..js module.js:664
Module.load module.js:566
tryModuleLoad module.js:506
Module._load module.js:498
Module.runMain module.js:694
startup bootstrap_node.js:204
(anonymous) bootstrap_node.js:625
Scope
Console
Search
C:\Windows\Syst...
Filter
Default levels


DevTools - Node.js
Connection
Console
Profiler
Sources
Memory
Node
Filesystem
Node.js: file:///D:/_nodejs_ap
app.js
(function (exports, require, module, __filename, __dirname) { c
name: 'Jan',
};
user.age = 22; user = {name: "Jan", age: 22}
debugger;
user.name = 'Adam';
console.log(user);
});
Line 6, Column 1
vm.js:80
Debugger paused
Watch
Call Stack
(anonymous) app.js:6
Module._compile module.js:650
Module._extensions..js module.js:664
Module.load module.js:566
tryModuleLoad module.js:506
Module._load module.js:498
Module.runMain module.js:694
startup bootstrap_node.js:204
(anonymous) bootstrap_node.js:625
Scope
Console
Search
C:\Windows\Syst...
Filter
Default levels


DevTools - Node.js
Connection
Console
Profiler
Sources
Memory
Node
Filesystem
Node.js: file:///D:/_nodejs_ap
app.js
(function (exports, require, module, __filename, __dirname) { c
name: 'Jan',
};
user.age = 22; user = {name: "Jan", age: 22}
debugger;
user.name = 'Adam';
console.log(user);
});
{}
Line 6, Column 1
vm.js:80
Debugger paused
Watch
Call Stack
(anonymous) app.js:6
Module._compile module.js:650
Module._extensions..js module.js:664
Module.load module.js:566
tryModuleLoad module.js:506
Module._load module.js:498
Module.runMain module.js:694
startup bootstrap_node.js:204
(anonymous) bootstrap_node.js:625
Scope
Console
Search
C:\Windows\Syst...
Filter
Default levels
1+1
2
user
{name: "Jan", age: 22}
user.age = 32
32

DEBUGGING IN VISUAL STUDIO CODE


File
Edit
Selection
View
Go
Debug
Terminal
Help
app.js - nodejs - Visual Studio Code
EXPLORER
OPEN EDITORS
app.js
NODEJS
app.js
OUTLINE
Start Debugging F5
Start Without Debugging Ctrl+F5
Stop Debugging Shift+F5
Restart Debugging Ctrl+Shift+F5
Open Configurations
Add Configuration...
Step Over F10
Step Into F11
Step Out Shift+F11
Continue F5
Toggle Breakpoint F9
New Breakpoint
Enable All Breakpoints
Disable All Breakpoints
0 0
Installing packages...
UTF-8
CRLF
JavaScript


File Edit Selection View Go Debug Terminal Help
app.js - nodejs - Visual Studio Code
DEBUG
No Cor
VARIABLES
Local
this: Object
__dirname: "d:\nodejs"
__filename: "d:\nodejs\ap...
exports: Object {}
WATCH
CALL STACK
PAUSED ON DEBUGGER ...
(anonymous function) app.js
Module._compile module.js
app.js ...
1 const user = {
2 name: 'Jan',
3 };
4
5 user.age = 22;
6 debugger;
7 user.name = 'Adam';
8
9 console.log(user);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
C:\Program Files\nodejs\node.exe --inspect-brk=30472 app.js
Debugger listening on ws://127.0.0.1:30472/0feca61e-195e-455a-958b-8aa9117ccfff
0 0
Installing packages...
Ln 6, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript