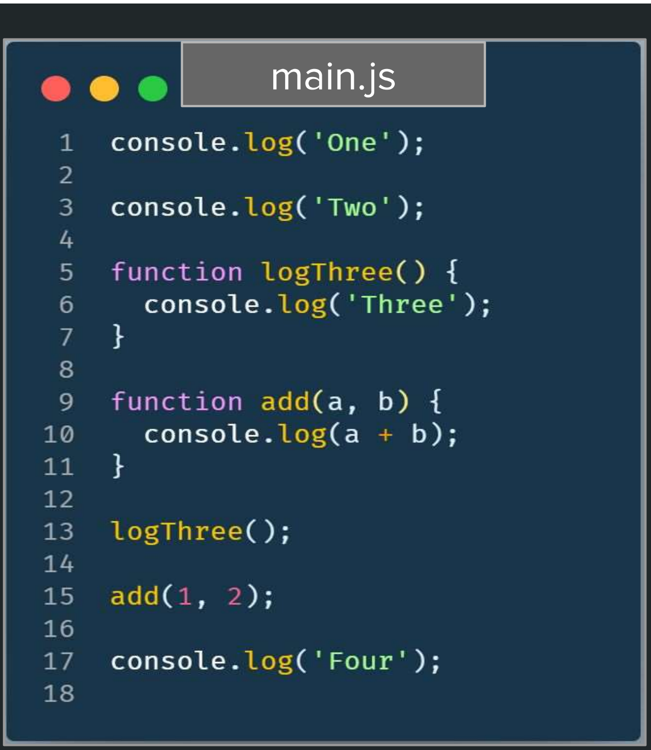
Concurrency In Javascript

Presented by Mubashir Ali

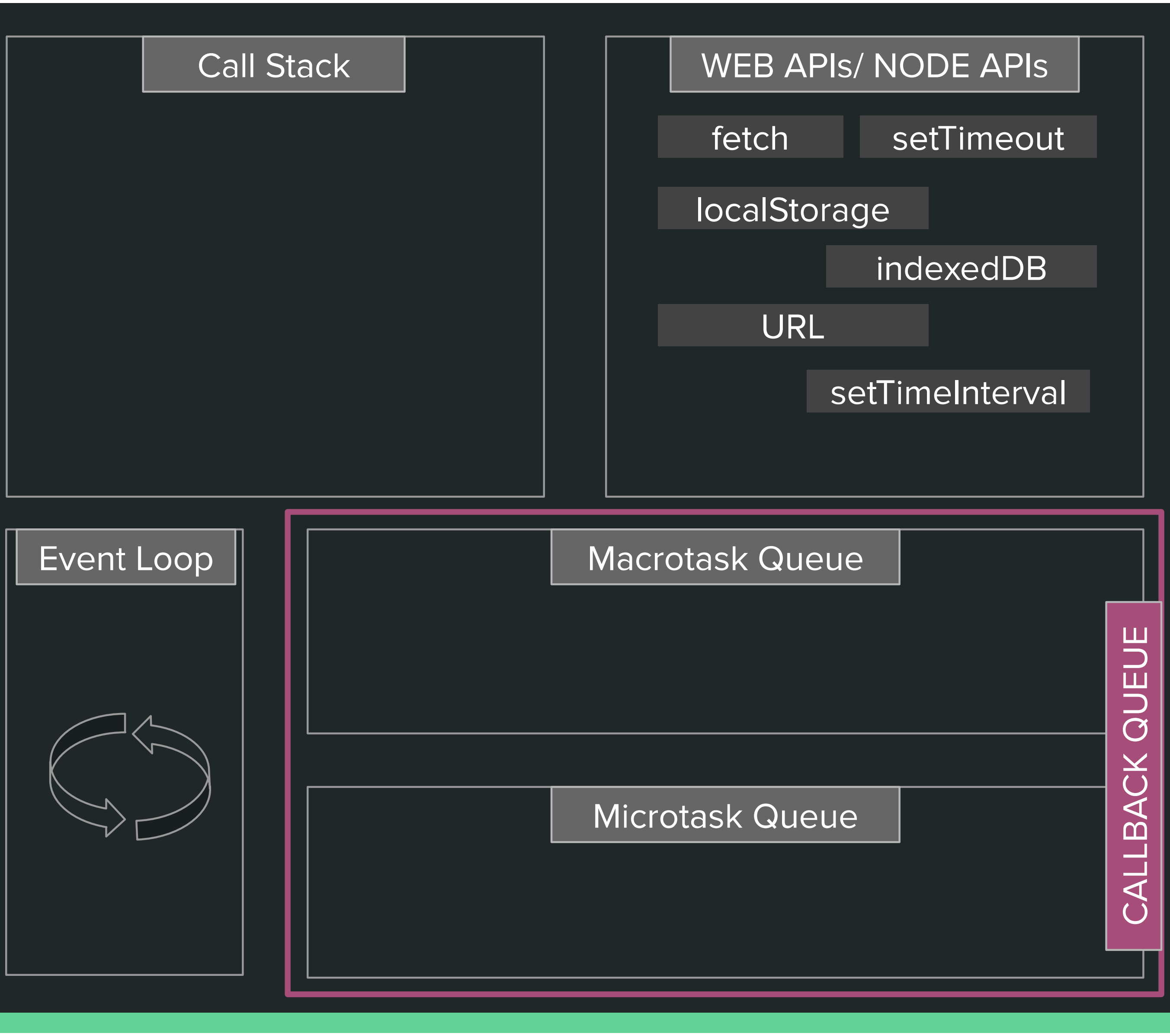
Synchronous Javascript

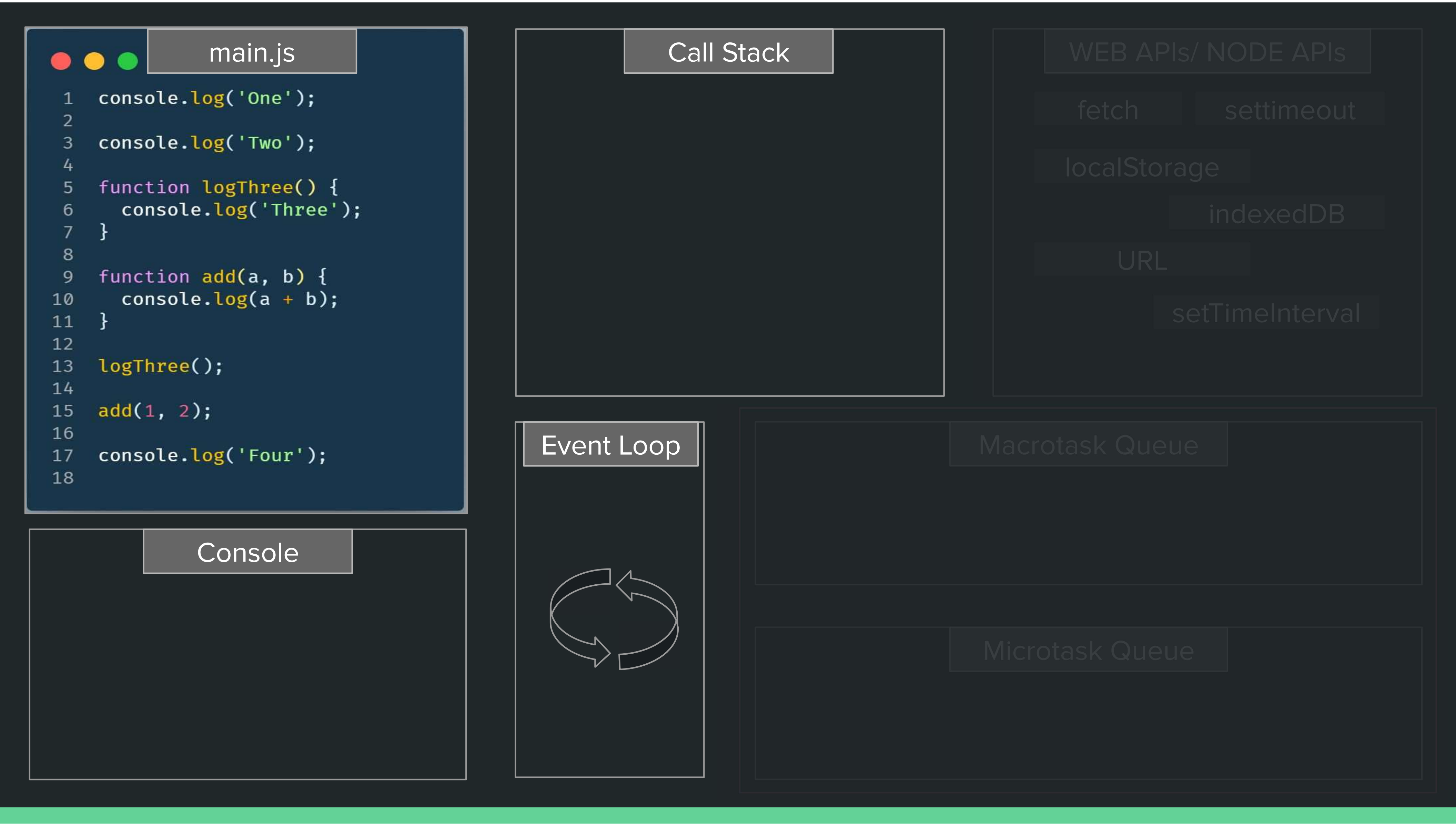


main.js

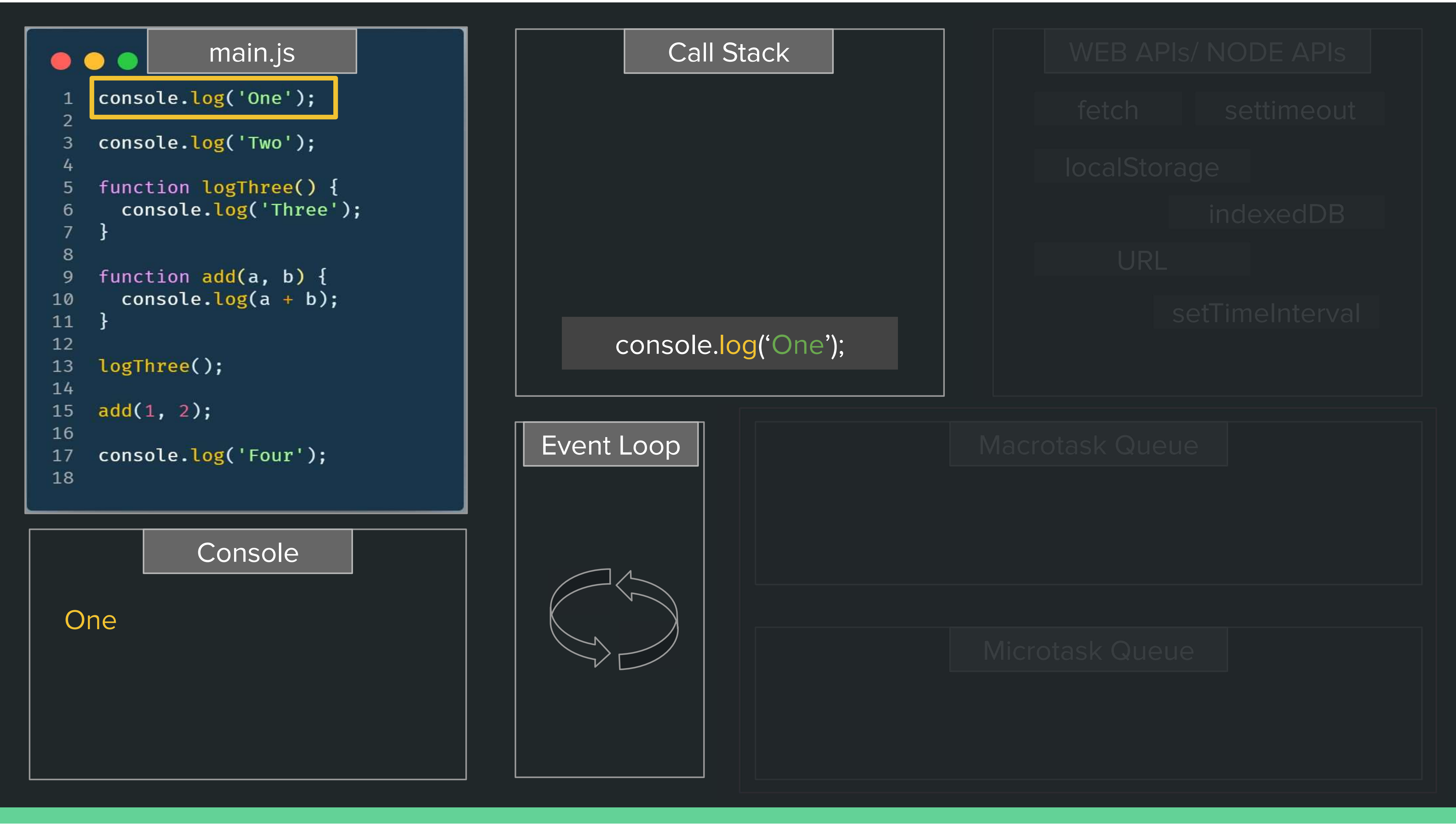
```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');

```

Console





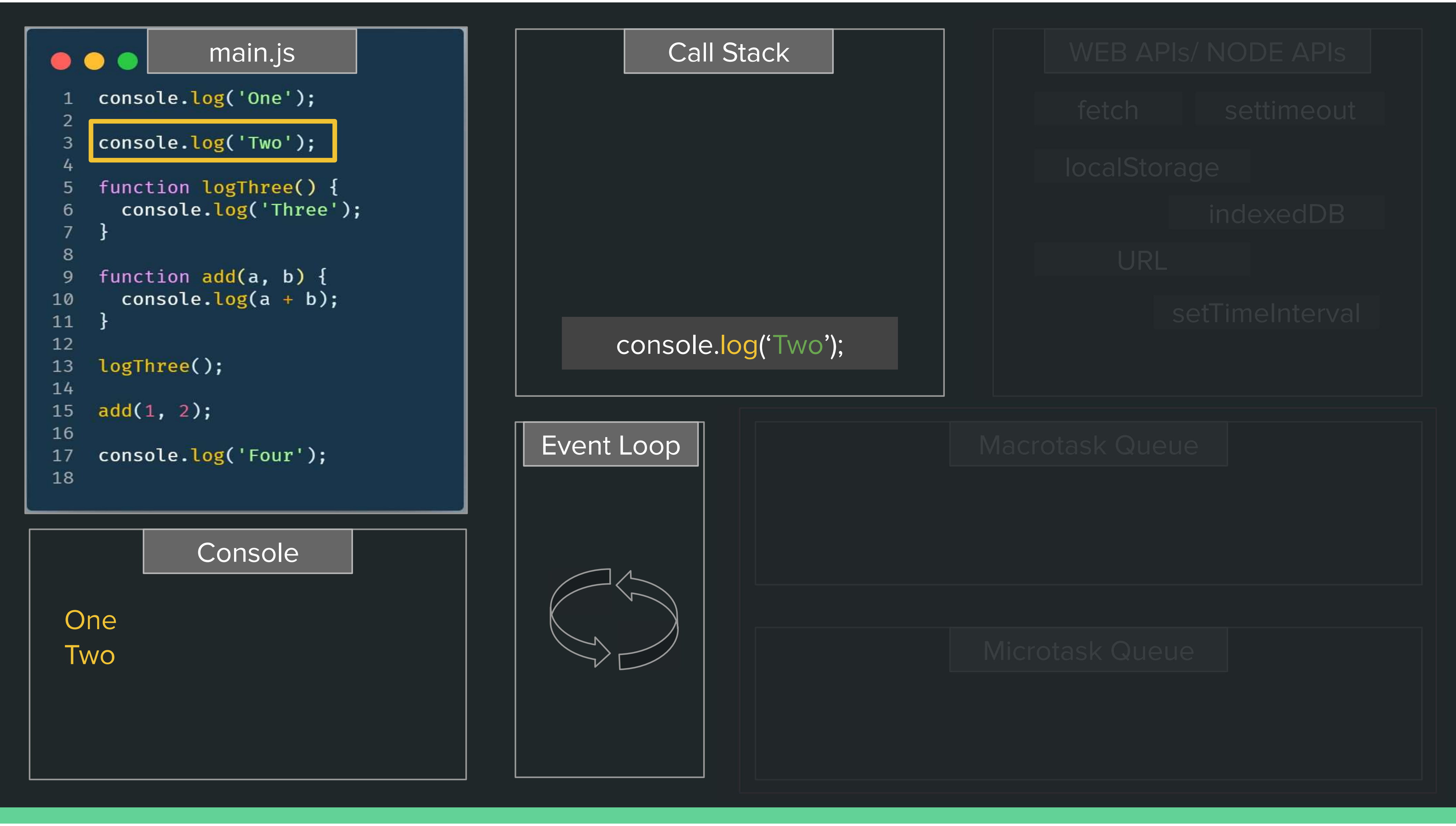
```javascript
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');

```



```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');
```

```
console.log('One');
```

```
One
```



main.js

```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');
```

Call Stack

```
console.log('Two');
```

WEB APIs/ NODE APIs

fetch

settimeout

localStorage

indexedDB

URL

setTimeInterval

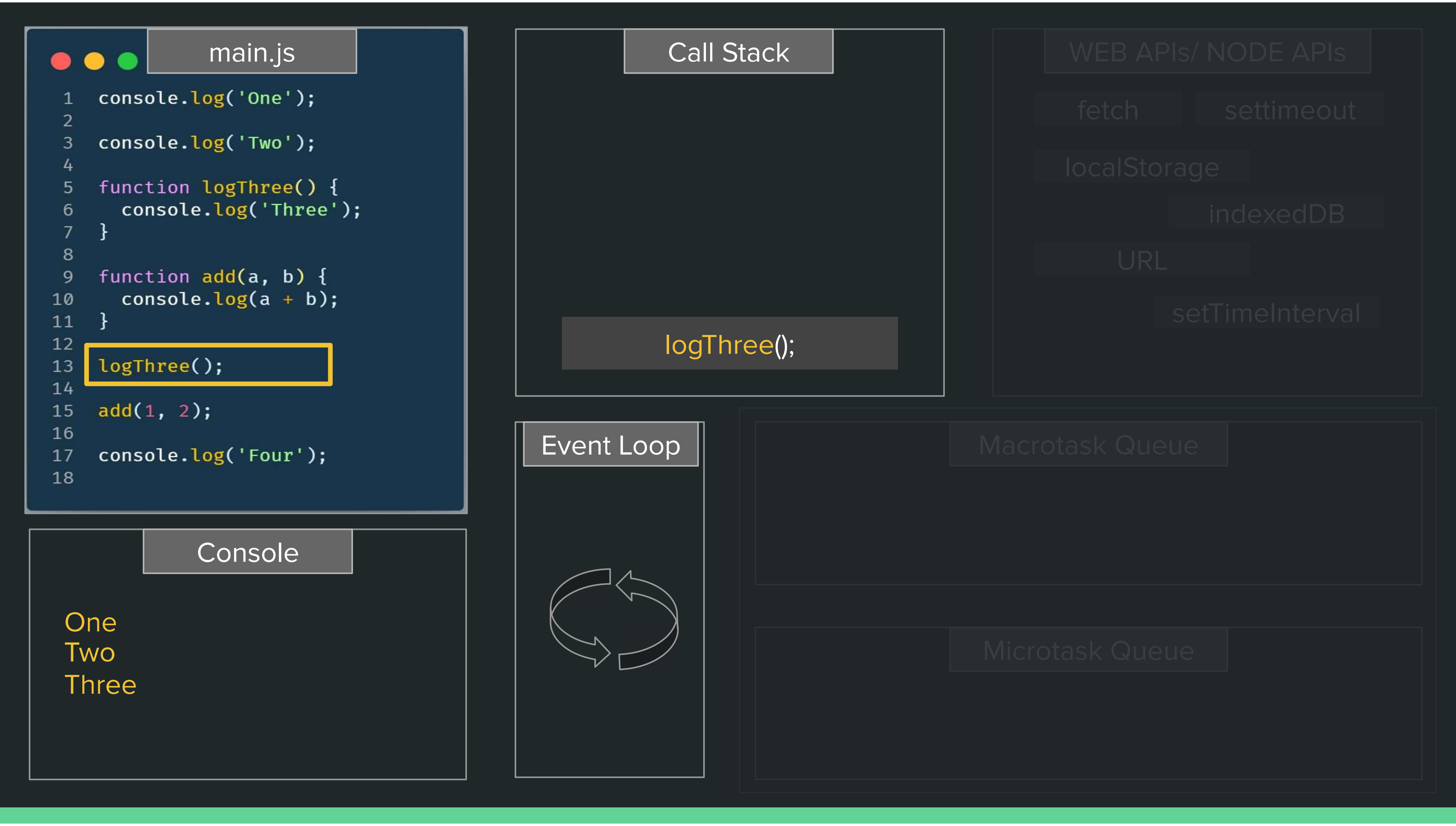
Console

```
One
Two
```

Event Loop

Macrotask Queue

Microtask Queue



main.js

```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');

```

Call Stack

logThree();

Console

```
One
Two
Three
```

Event Loop

WEB APIs/ NODE APIs

fetch

settimeout

localStorage

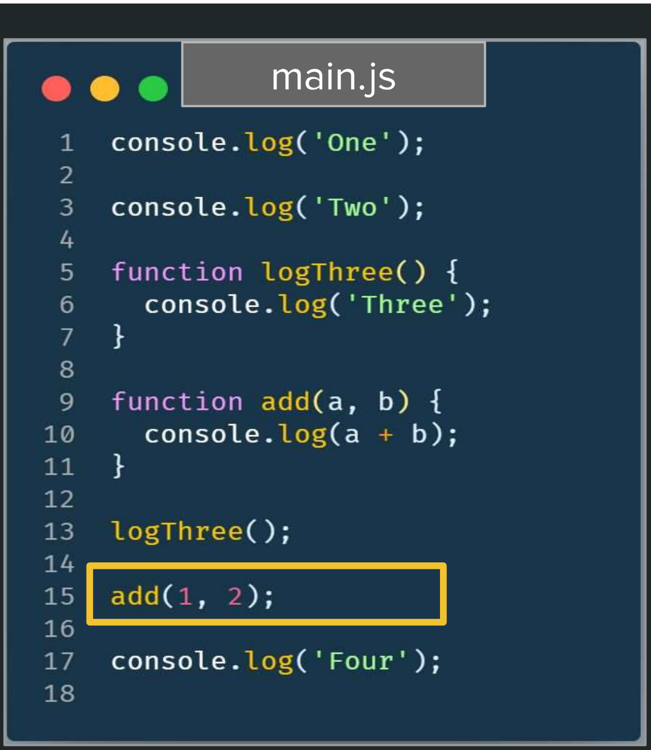
indexedDB

URL

setTimeInterval

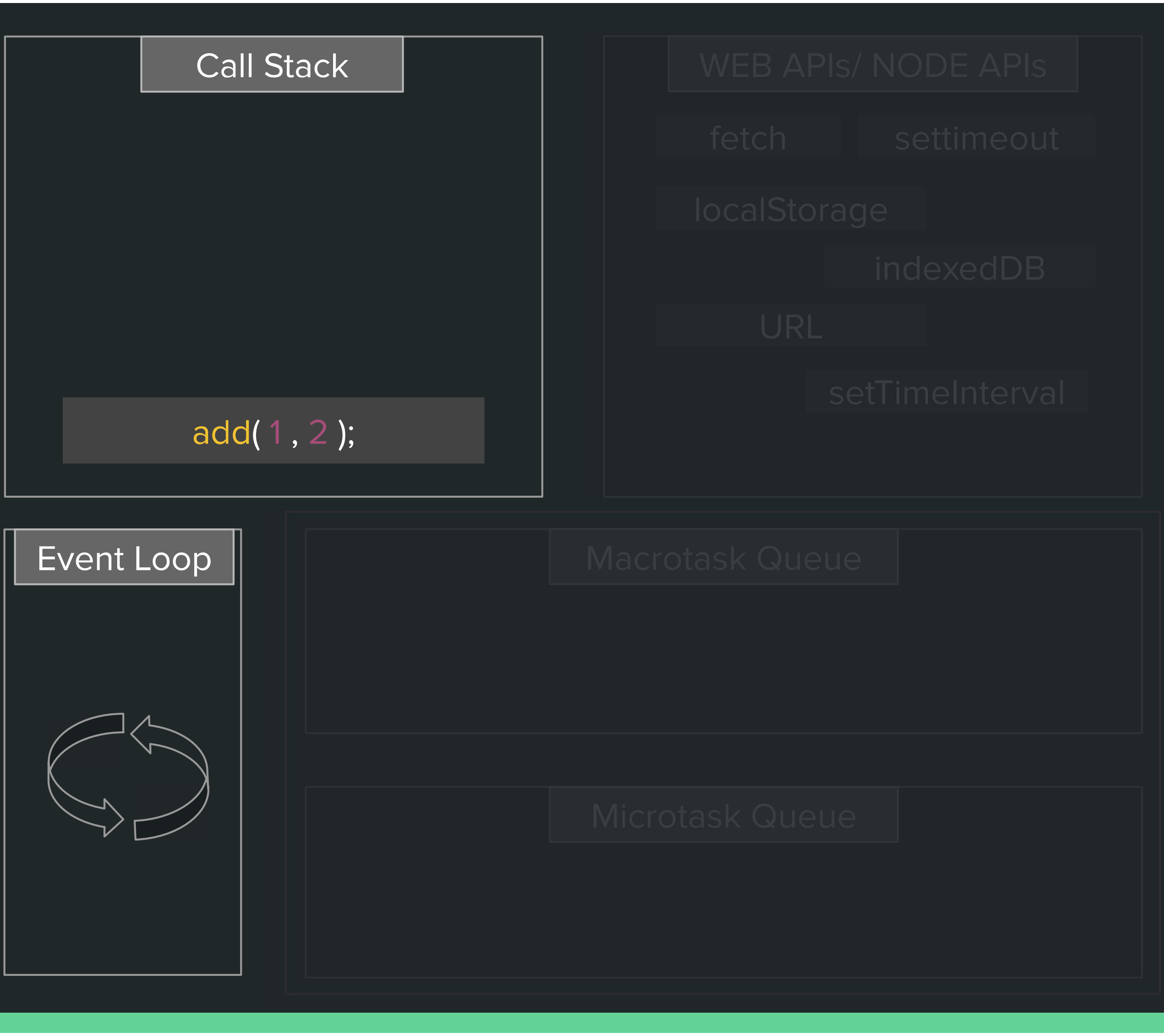
Macrotask Queue

Microtask Queue



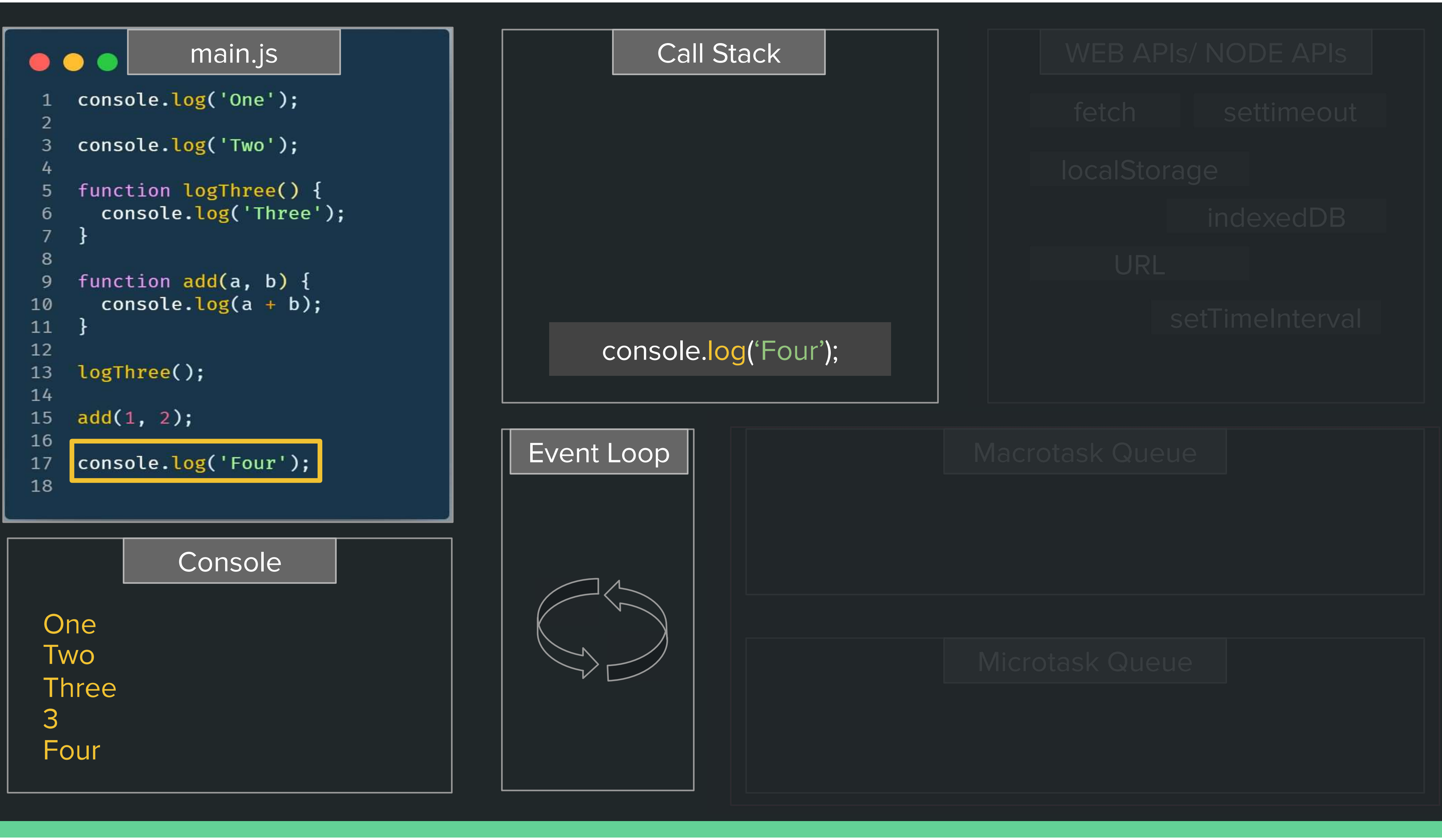
main.js

```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');

```



Console

```
One
Two
Three
3
```



main.js

```
console.log('One');

console.log('Two');

function logThree() {
  console.log('Three');
}

function add(a, b) {
  console.log(a + b);
}

logThree();

add(1, 2);

console.log('Four');

```

Call Stack

console.log('Four');

WEB APIs/ NODE APIs

fetch

settimeout

localStorage

indexedDB

URL

setTimeInterval

Event Loop

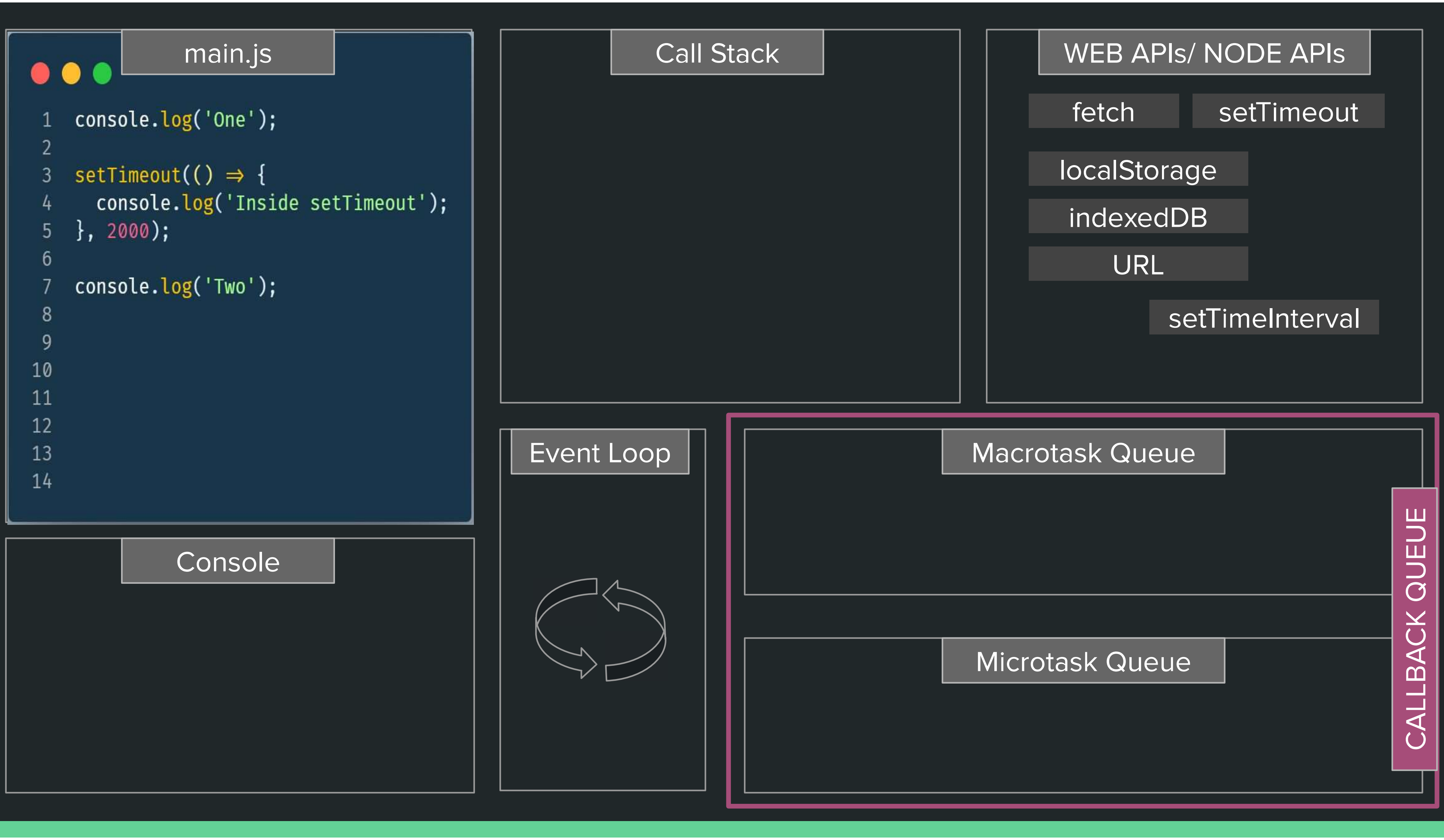
Macrotask Queue

Microtask Queue

Console

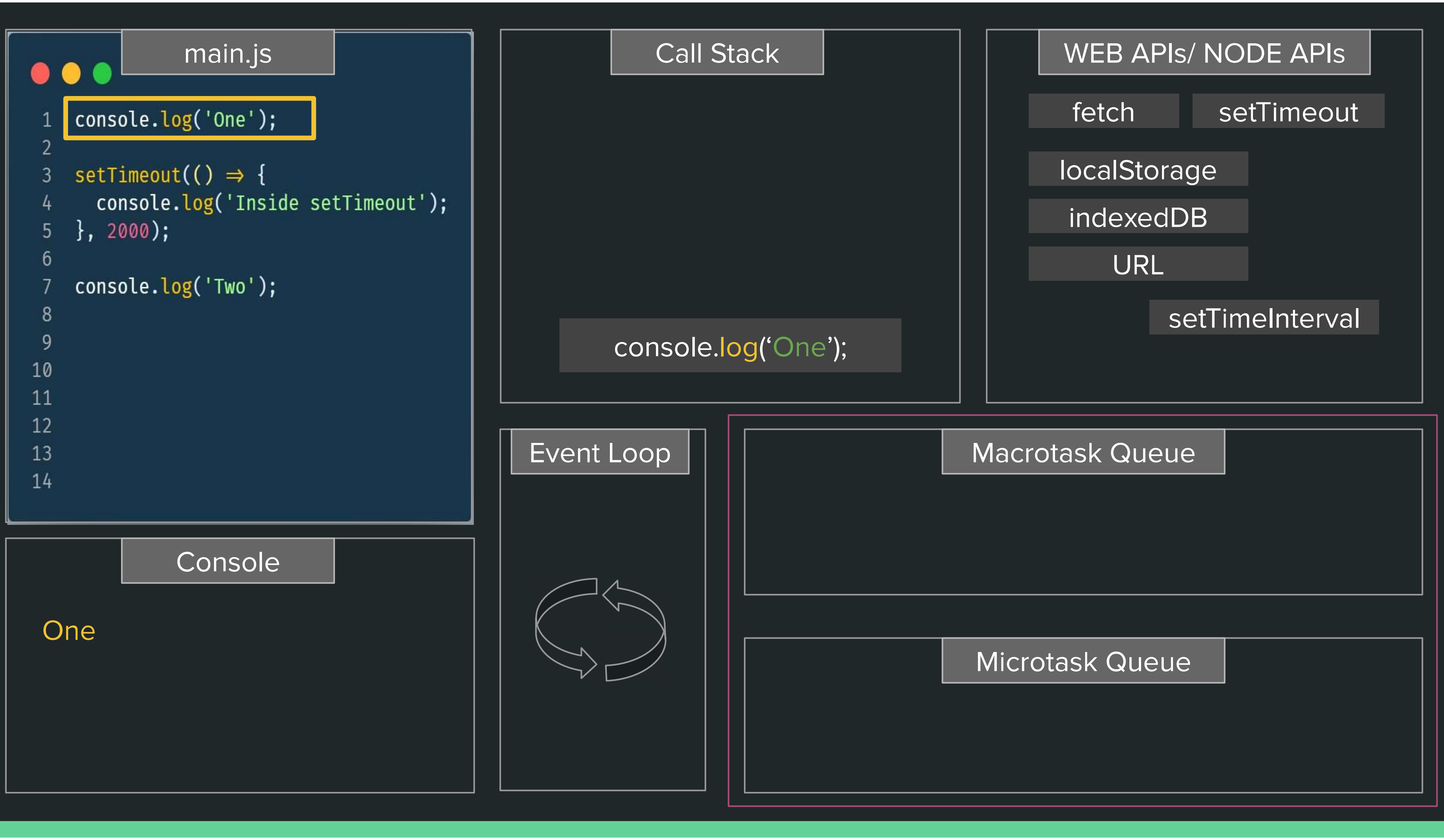
```
One
Two
Three
3
Four
```

Asynchronous Javascript



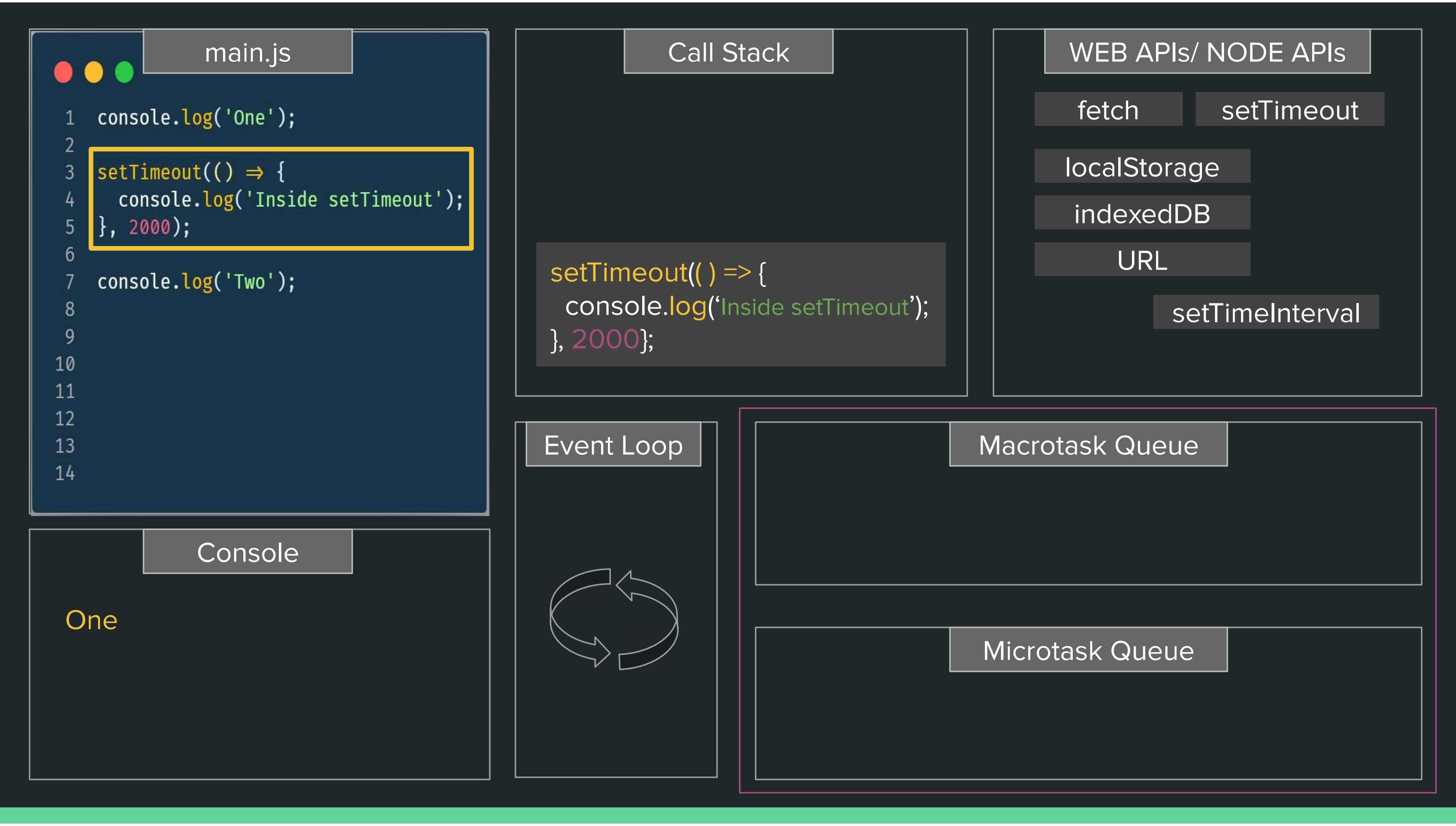
main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```



```js
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

```
One
```



main.js

```
console.log('One');

setTimeout(() ⇒ {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

```
setTimeout(( ) => {
  console.log('Inside setTimeout');
}, 2000};
```

WEB APIs/ NODE APIs

fetch

setTimeout

localStorage

indexedDB

URL

setTimeInterval

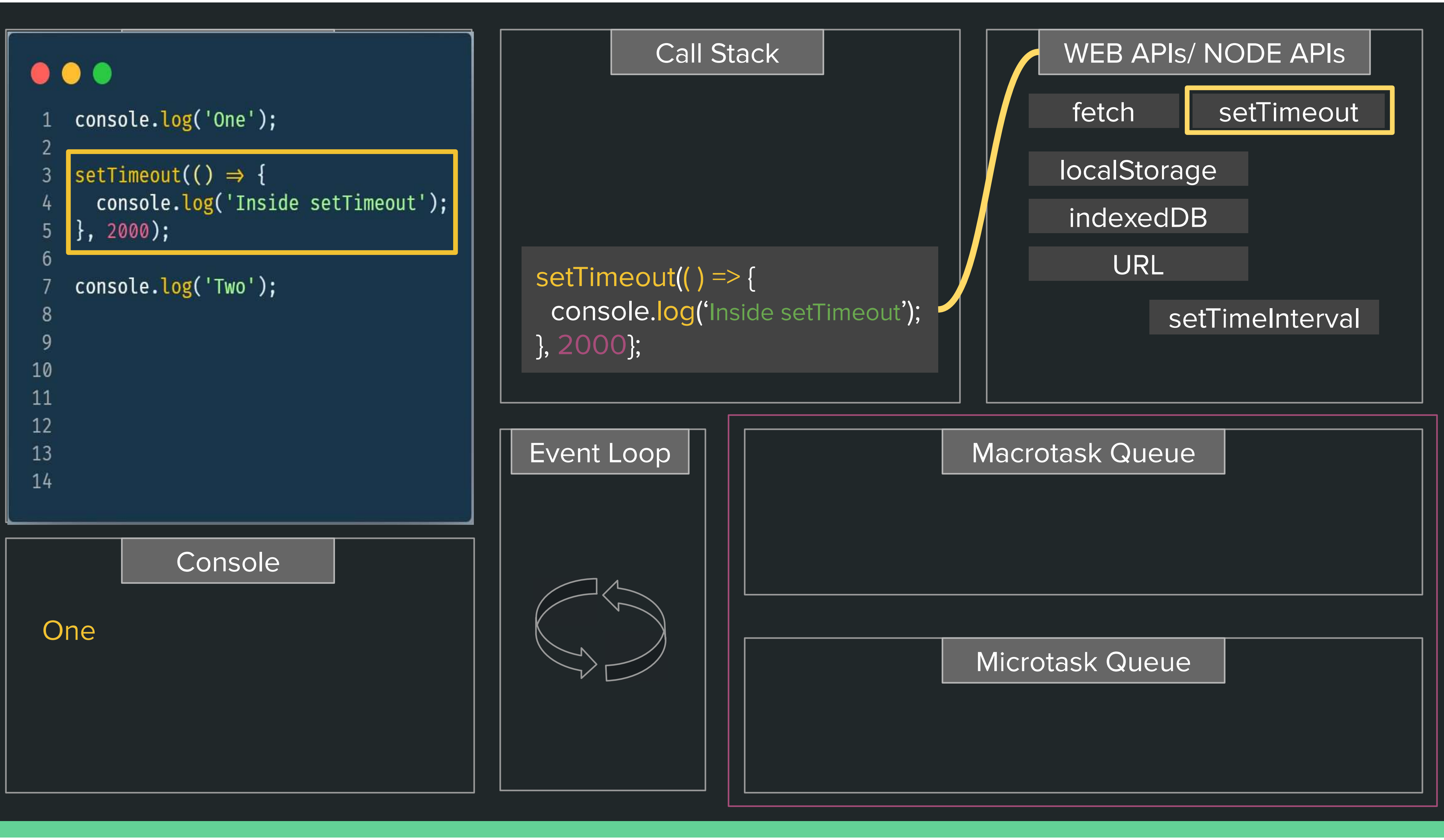
Console

```
One
```

Event Loop

Macrotask Queue

Microtask Queue



```
console.log('One');

setTimeout(() ⇒ {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

```
setTimeout(( ) => {
  console.log('Inside setTimeout');
}, 2000};
```

WEB APIs/ NODE APIs

- fetch
- setTimeout
- localStorage
- indexedDB
- URL
- setTimeInterval

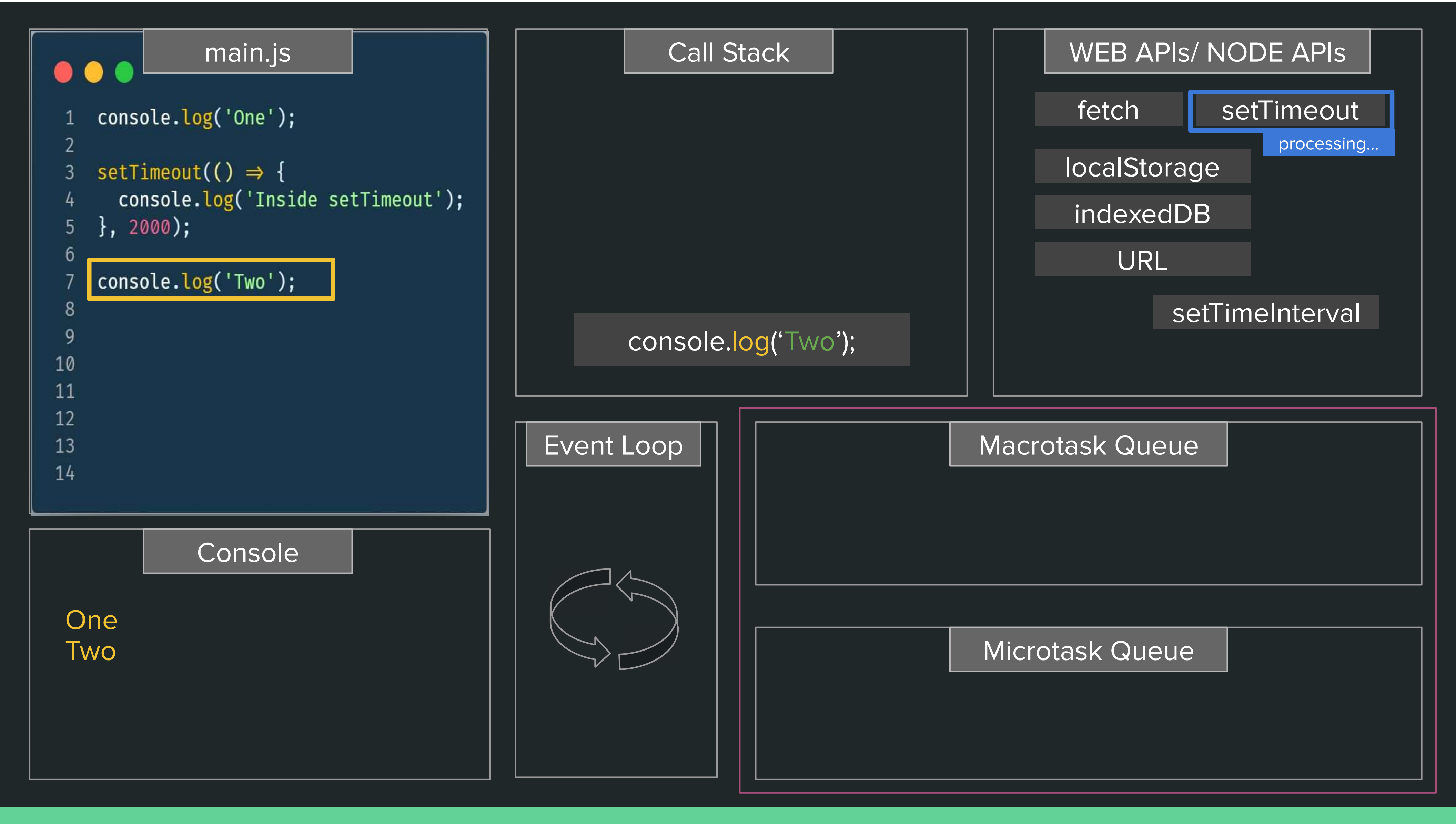
Event Loop

Macrotask Queue

Microtask Queue

Console

```
One
```



main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

console.log(‘Two’);

WEB APIs/ NODE APIs

fetch

setTimeout

processing...

localStorage

indexedDB

URL

setTimeInterval

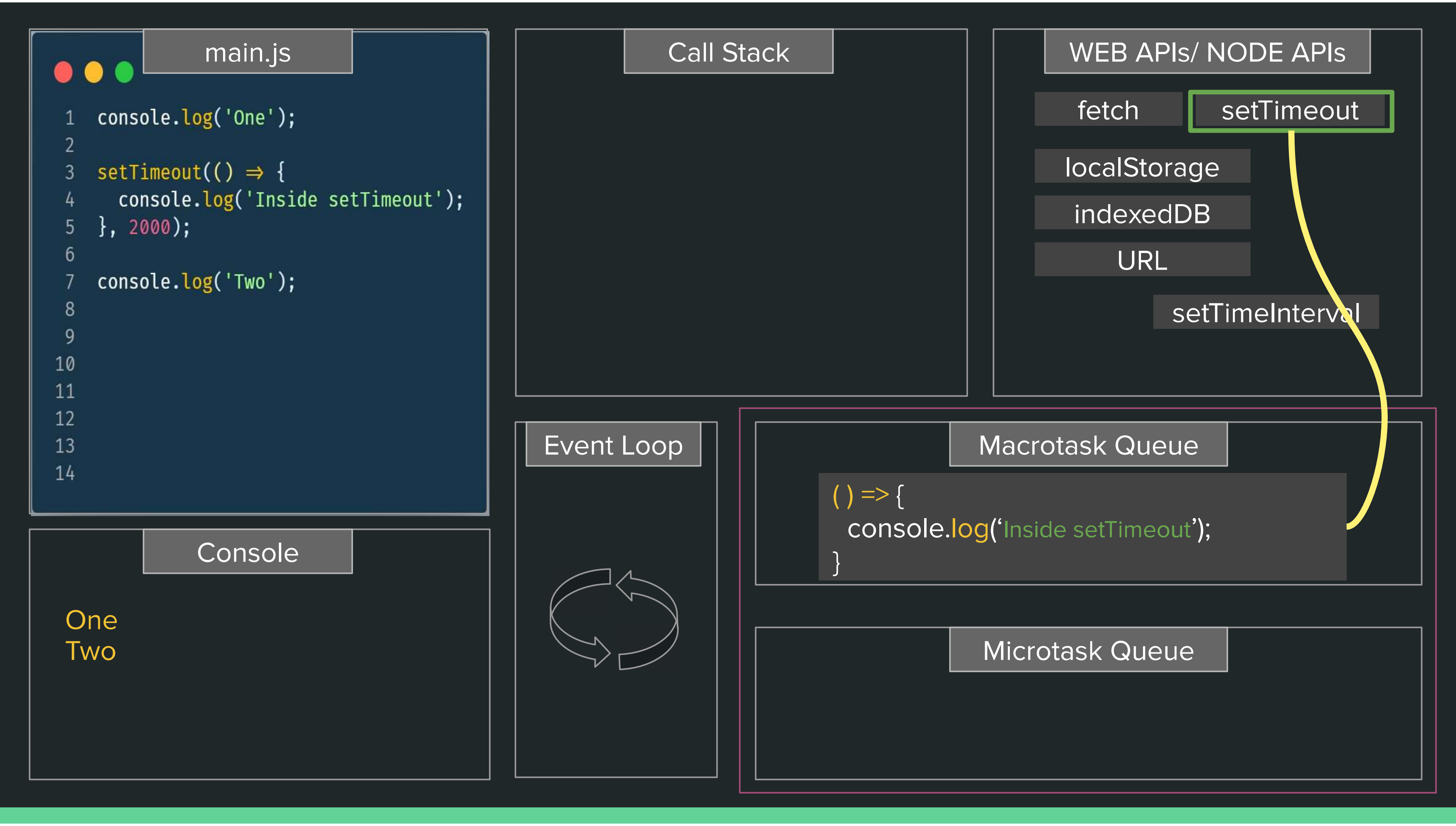
Event Loop

Macrotask Queue

Microtask Queue

Console

One
Two



main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

WEB APIs/ NODE APIs

fetch

setTimeout

localStorage

indexedDB

URL

setTimeInterval

Console

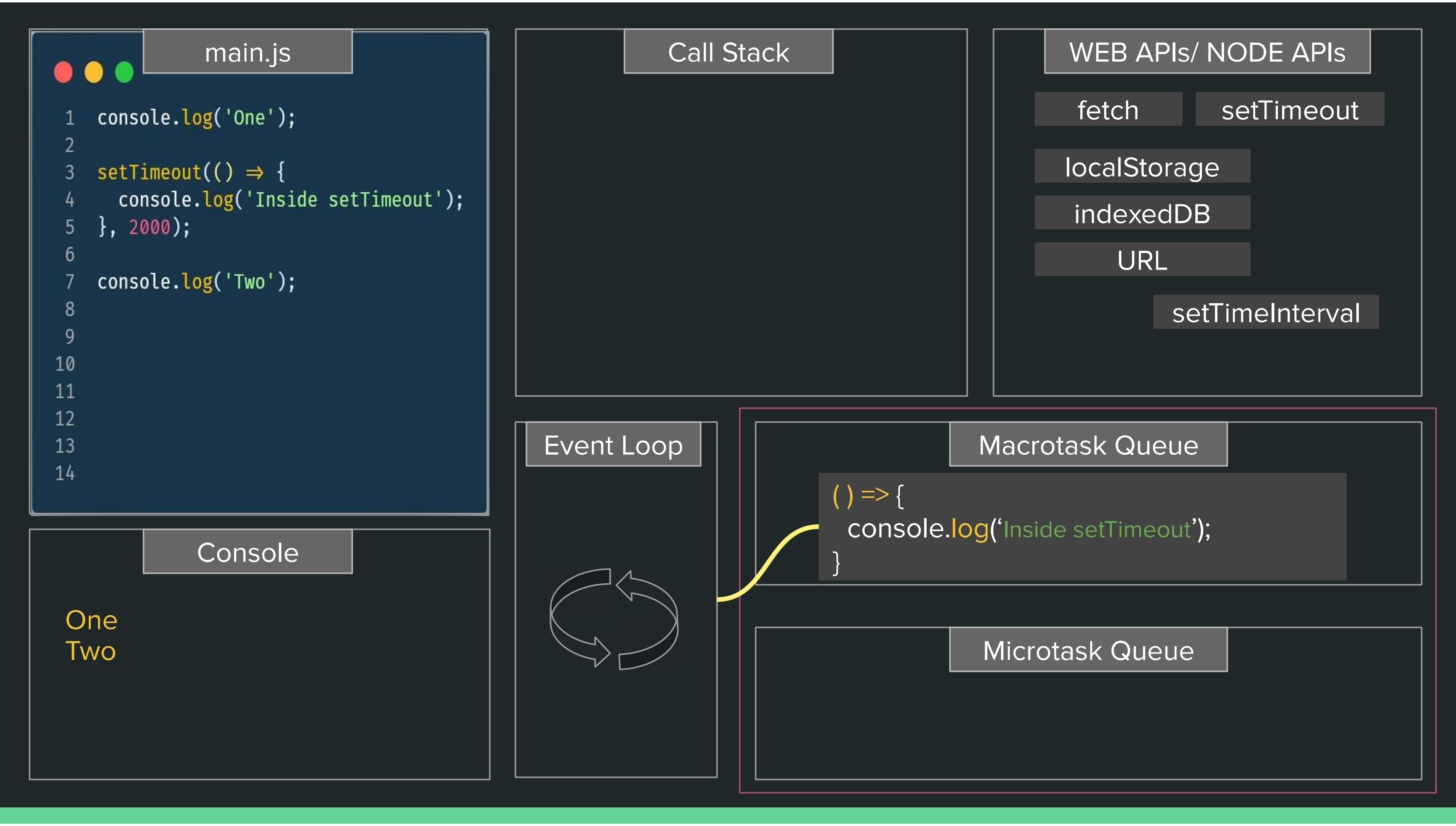
```
One
Two
```

Event Loop

Macrotask Queue

```
( ) => {
  console.log(‘Inside setTimeout’);
}
```

Microtask Queue



main.js

```
console.log('One');

setTimeout(() ⇒ {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

WEB APIs/ NODE APIs

Console

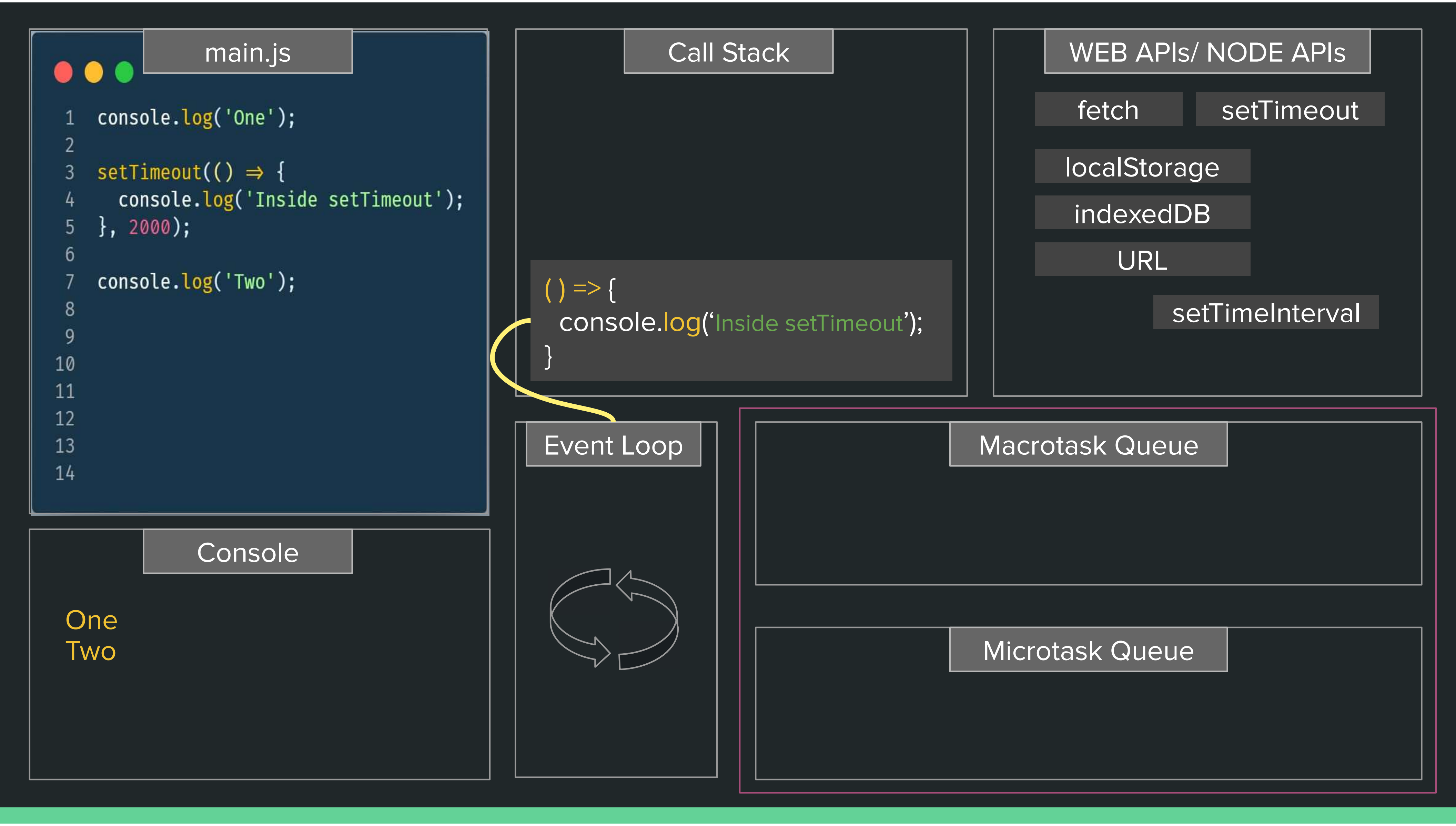
```
One
Two
```

Event Loop

Macrotask Queue

```
( ) => {
  console.log('Inside setTimeout');
}
```

Microtask Queue



main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

```
( ) => {
  console.log(‘Inside setTimeout’);
}
```

WEB APIs/ NODE APIs

fetch

setTimeout

localStorage

indexedDB

URL

setTimeInterval

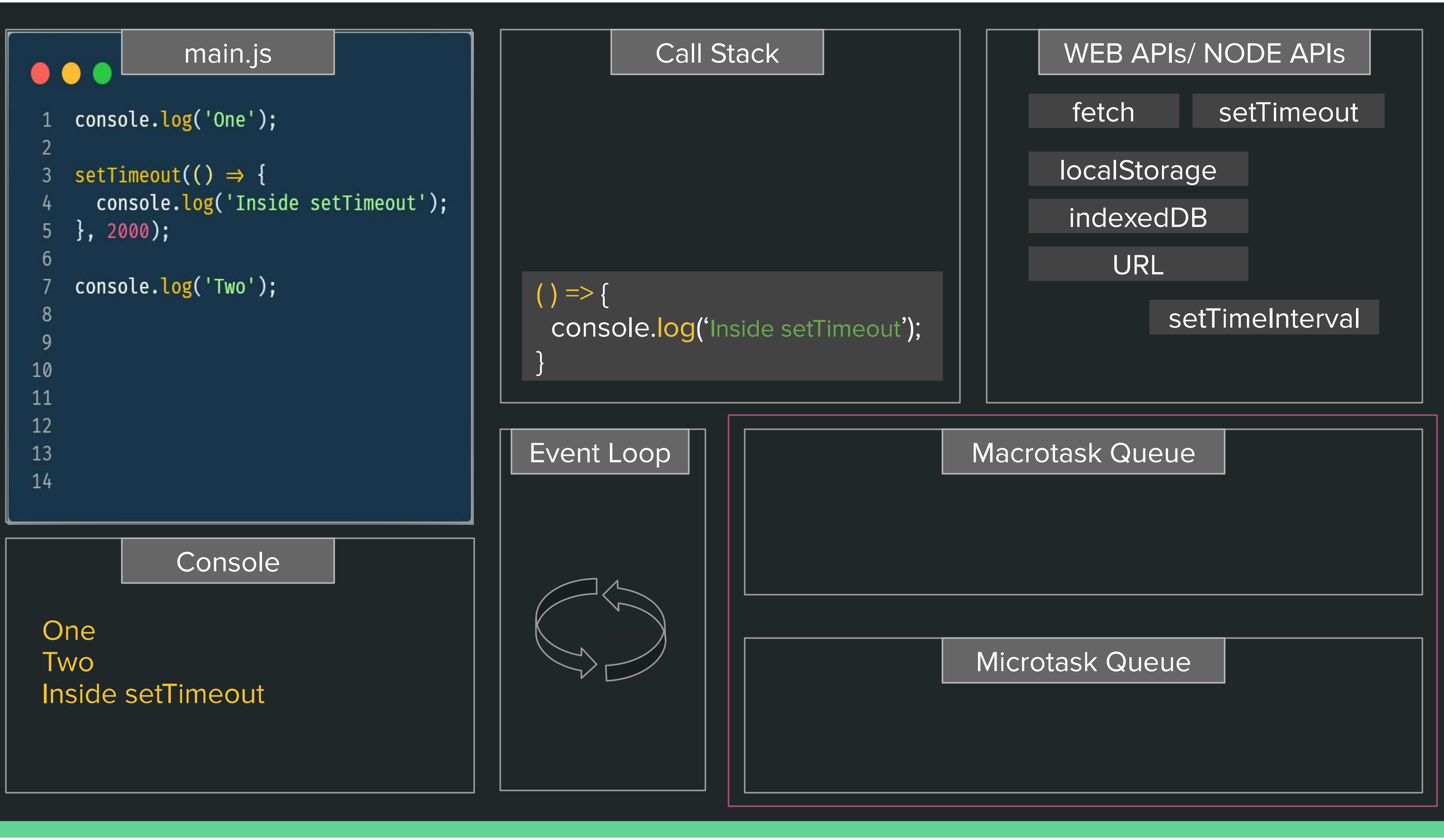
Event Loop

Macrotask Queue

Microtask Queue

Console

```
One
Two
```



main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Call Stack

```
( ) => {
  console.log('Inside setTimeout');
}
```

WEB APIs/ NODE APIs

fetch

setTimeout

localStorage

indexedDB

URL

setTimeInterval

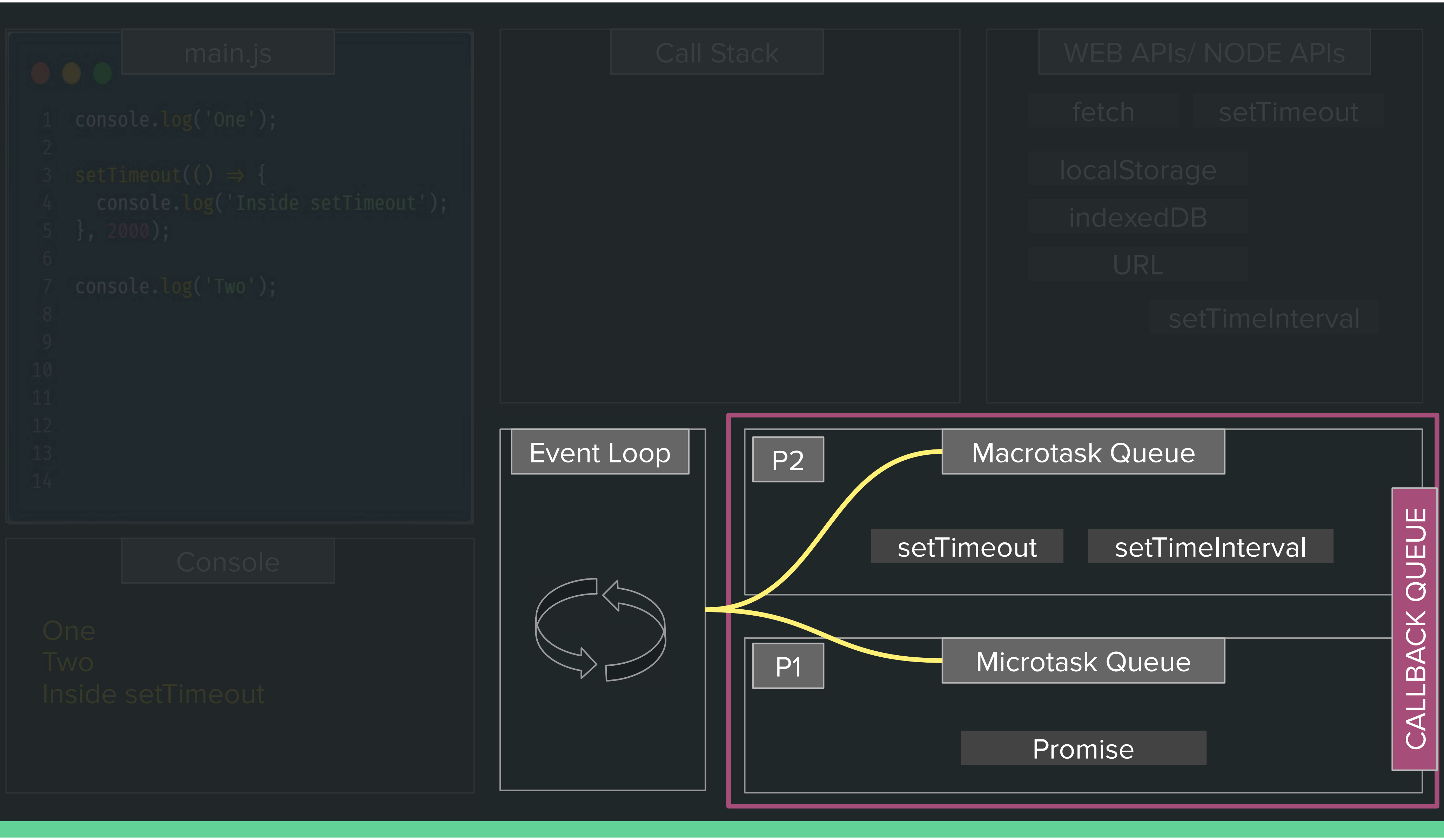
Console

```
One
Two
Inside setTimeout
```

Event Loop

Macrotask Queue

Microtask Queue



main.js

```
console.log('One');

setTimeout(() => {
  console.log('Inside setTimeout');
}, 2000);

console.log('Two');
```

Console

```
One
Two
Inside setTimeout
```

Thank You!