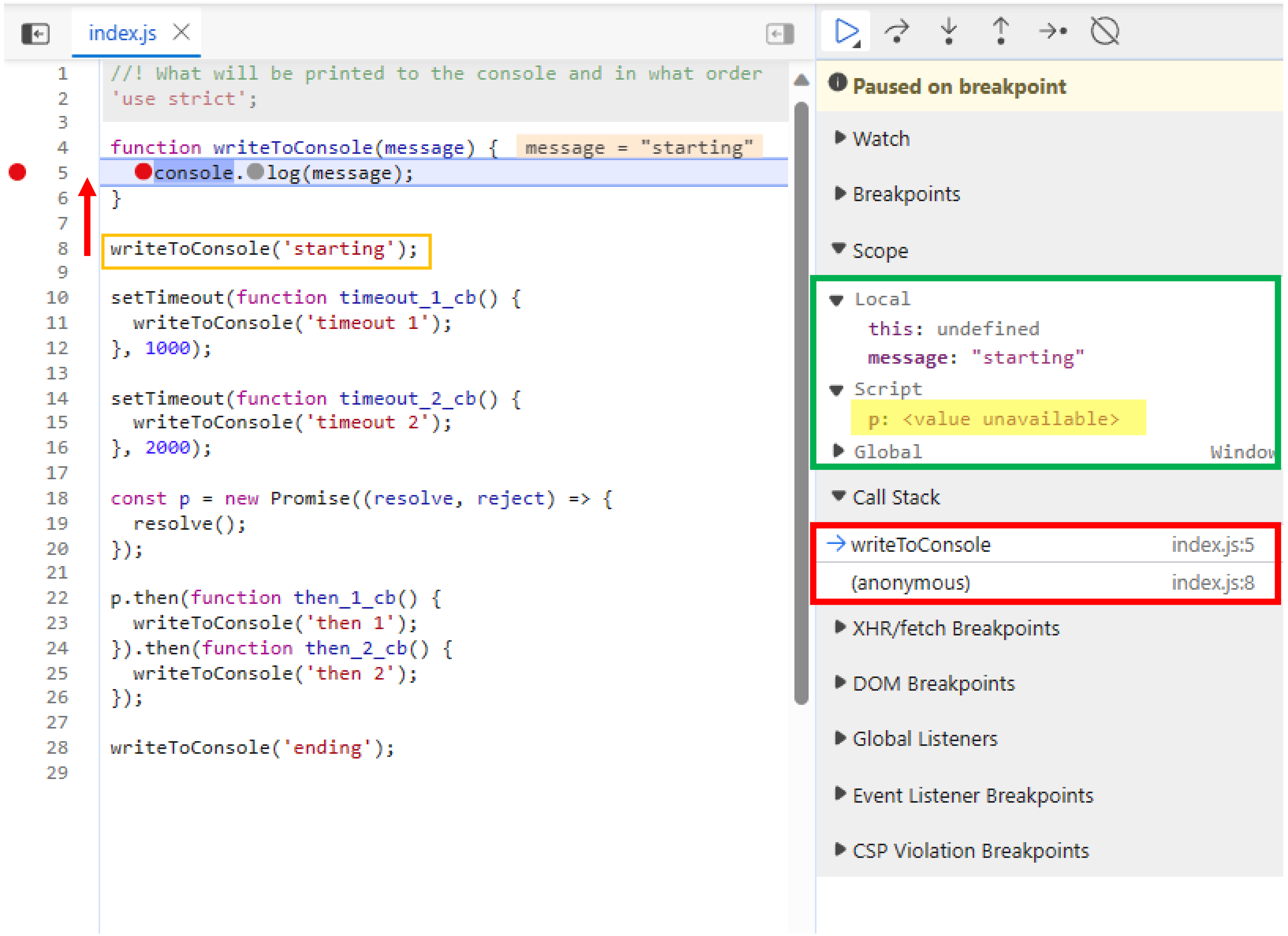
Event 1: Macrotask



index.js

```
 1  //! What will be printed to the console and in what order
 2  'use strict';
 3
 4  function writeToConsole(message) {   message = "starting"
 5    console.log(message);
 6  }
 7
 8  writeToConsole('starting');
 9
10  setTimeout(function timeout_1_cb() {
11    writeToConsole('timeout 1');
12  }, 1000);
13
14  setTimeout(function timeout_2_cb() {
15    writeToConsole('timeout 2');
16  }, 2000);
17
18  const p = new Promise((resolve, reject) => {
19    resolve();
20  });
21
22  p.then(function then_1_cb() {
23    writeToConsole('then 1');
24  }).then(function then_2_cb() {
25    writeToConsole('then 2');
26  });
27
28  writeToConsole('ending');
29
```

Paused on breakpoint

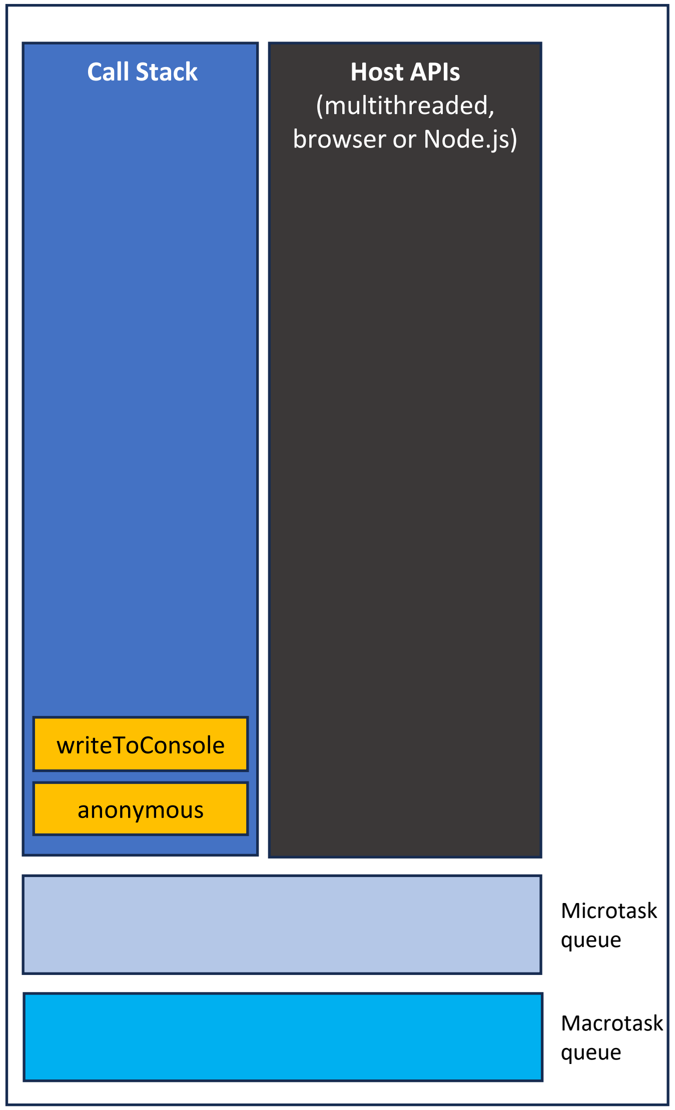
- Watch
- Breakpoints
- Scope

```
Local
  this: undefined
  message: "starting"
Script
  p: <value unavailable>
Global                                Windo
```

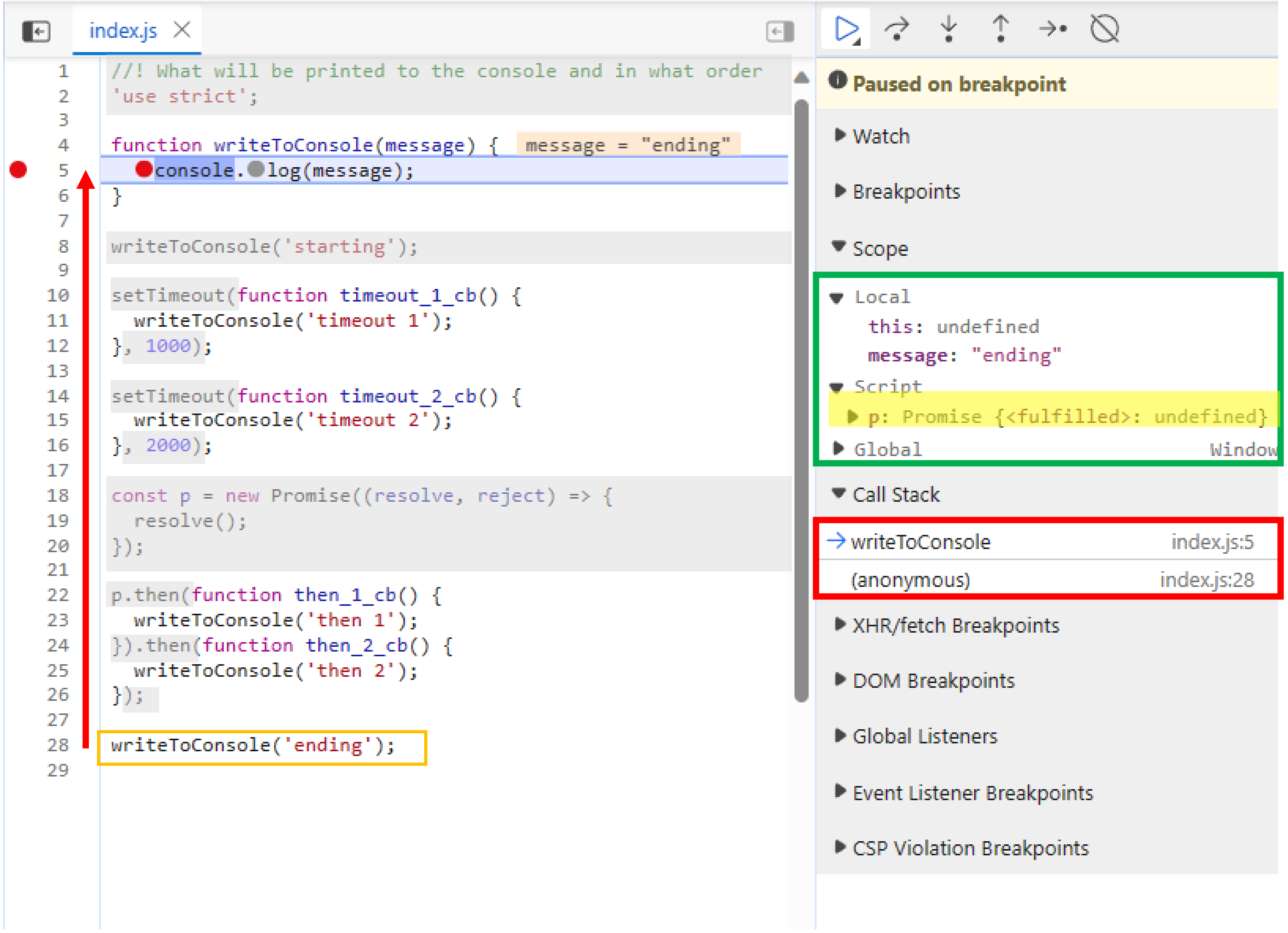
- Call Stack

writeToConsole	index.js:5
(anonymous)	index.js:8

- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints
- CSP Violation Breakpoints



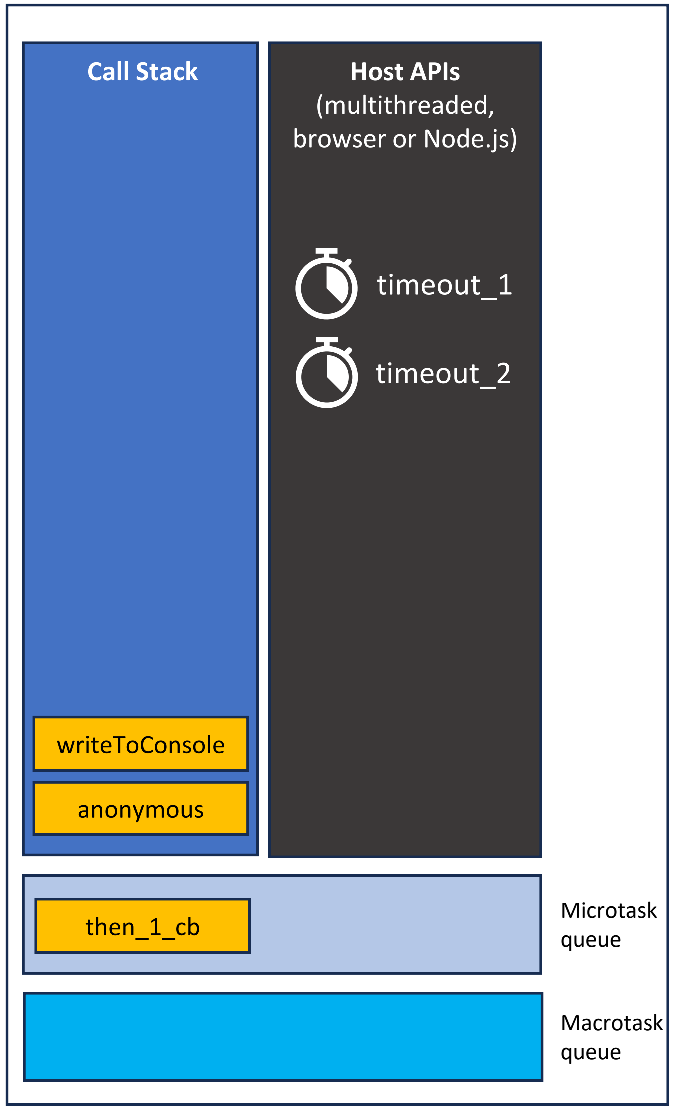
Event 1: Macrotask (cont'd)



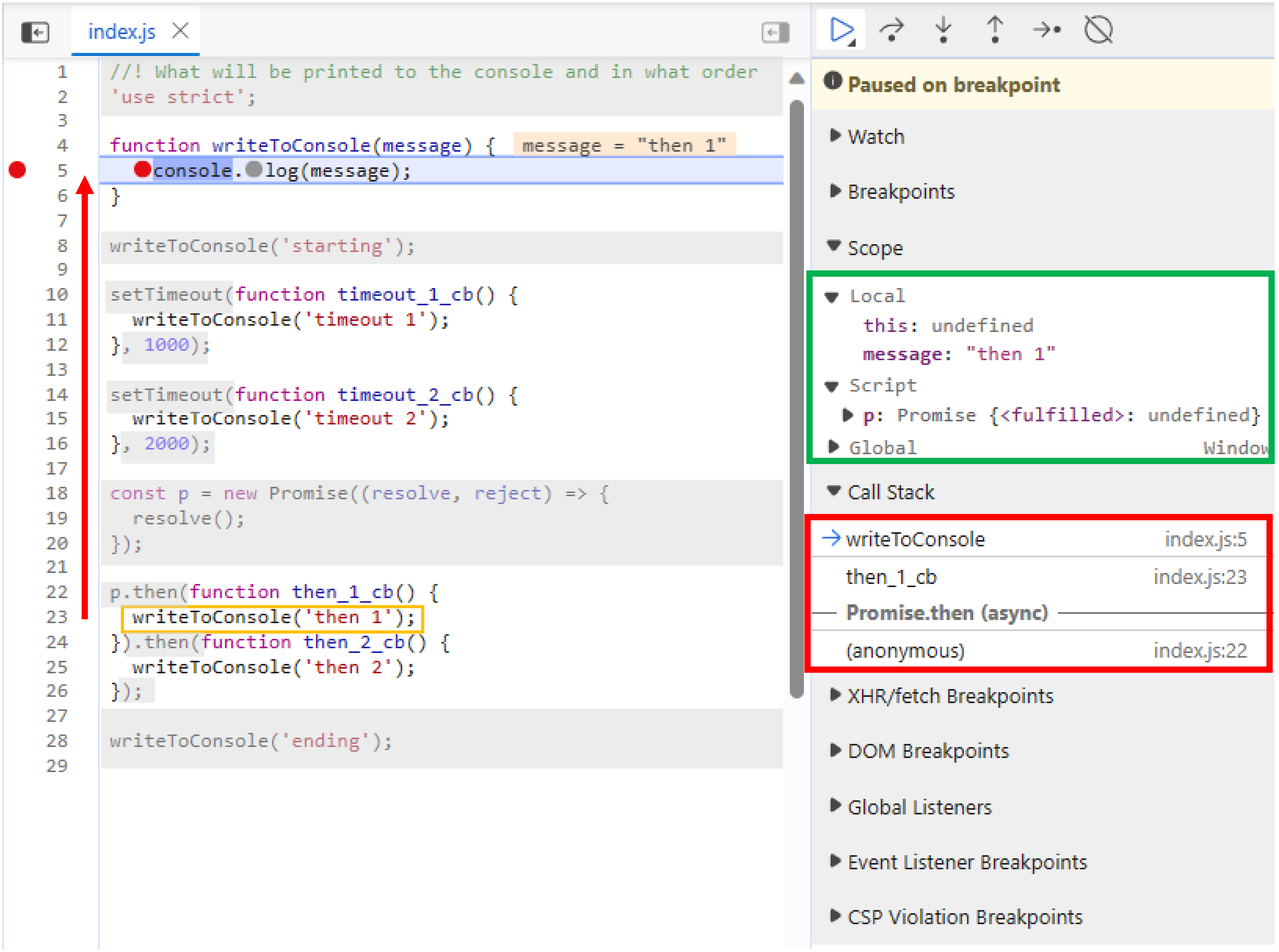
index.js

```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {  message = "ending"
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```

```
Paused on breakpoint
▶ Watch
▶ Breakpoints
▼ Scope
  ▼ Local
      this: undefined
      message: "ending"
  ▼ Script
    ▶ p: Promise {<fulfilled>: undefined}
  ▶ Global                                    Window
▼ Call Stack
  → writeToConsole                           index.js:5
    (anonymous)                              index.js:28
▶ XHR/fetch Breakpoints
▶ DOM Breakpoints
▶ Global Listeners
▶ Event Listener Breakpoints
▶ CSP Violation Breakpoints
```



Event 2: Microtask then_1_cb



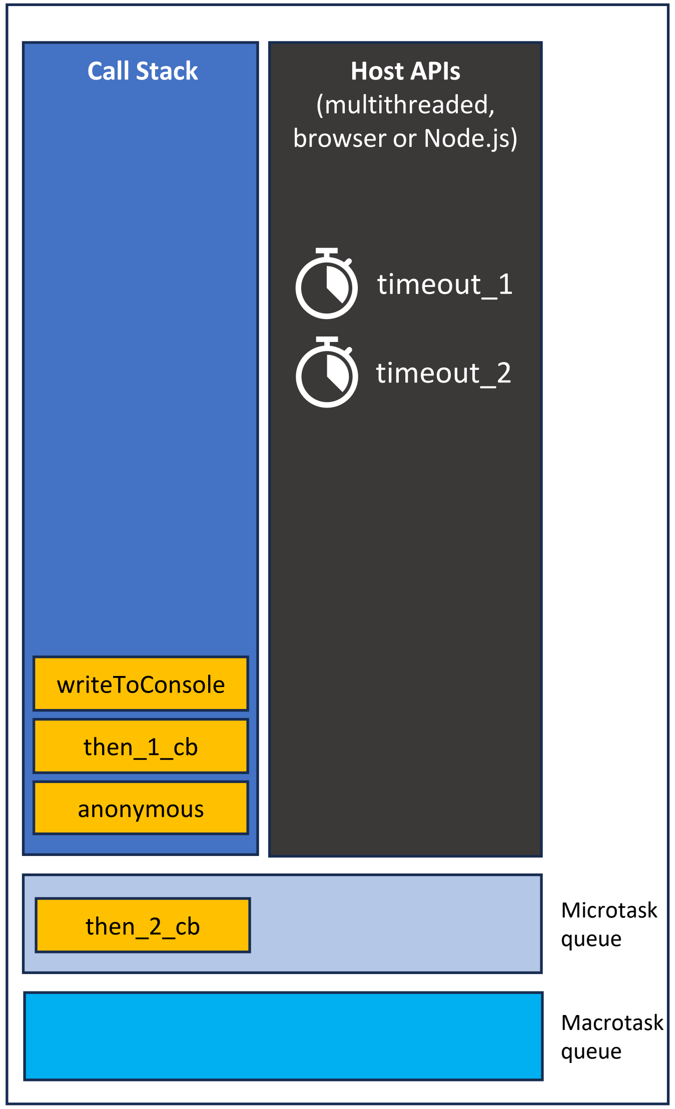
```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {  message = "then 1"
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```

Paused on breakpoint

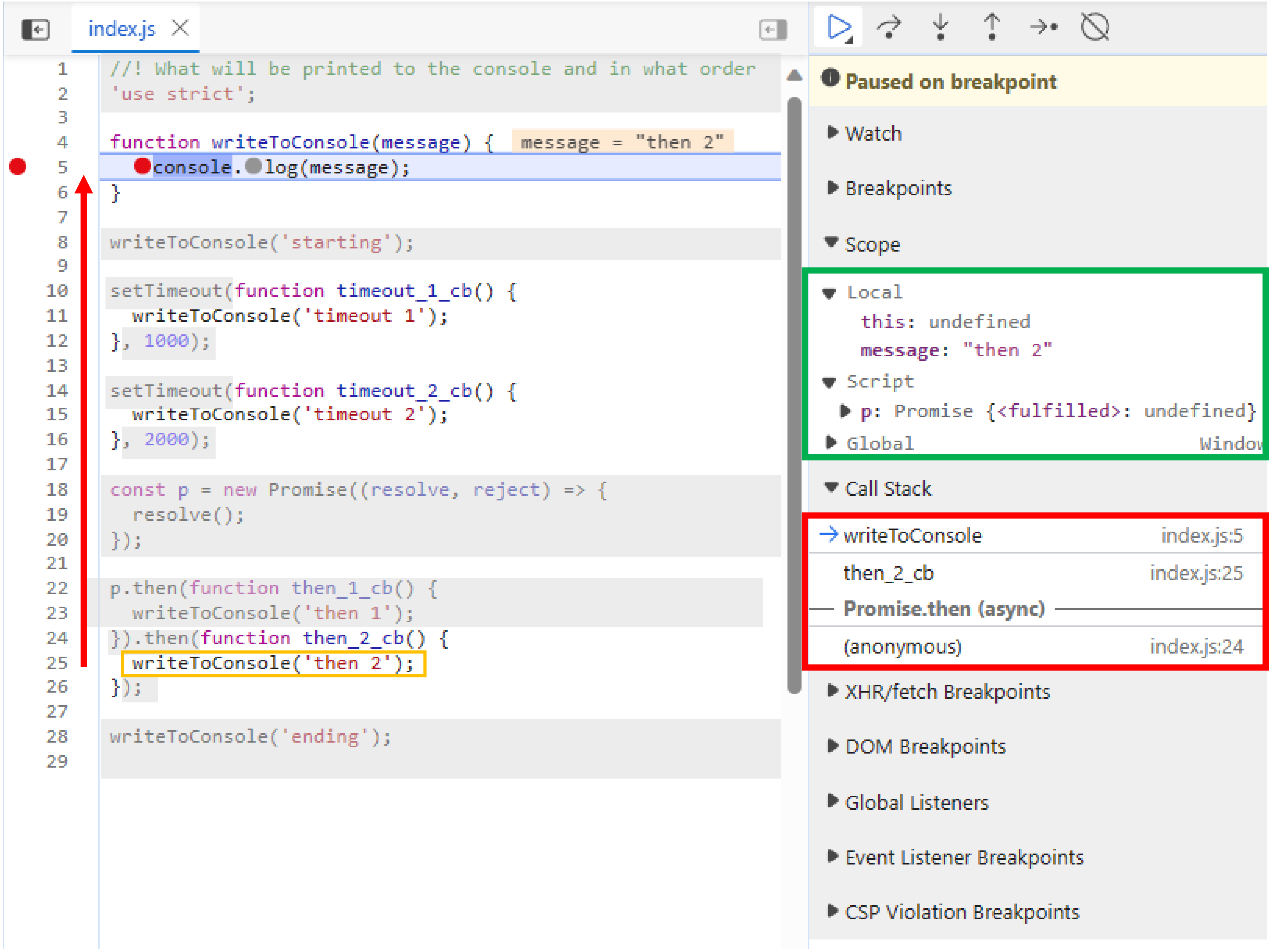
- Watch
- Breakpoints
- Scope
  - Local
    - this: undefined
    - message: "then 1"
  - Script
    - p: Promise {<fulfilled>: undefined}
  - Global Window
- Call Stack

```
→ writeToConsole      index.js:5
  then_1_cb           index.js:23
  Promise.then (async)
  (anonymous)         index.js:22
```

- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints
- CSP Violation Breakpoints



Event 3: Microtask then_2_cb



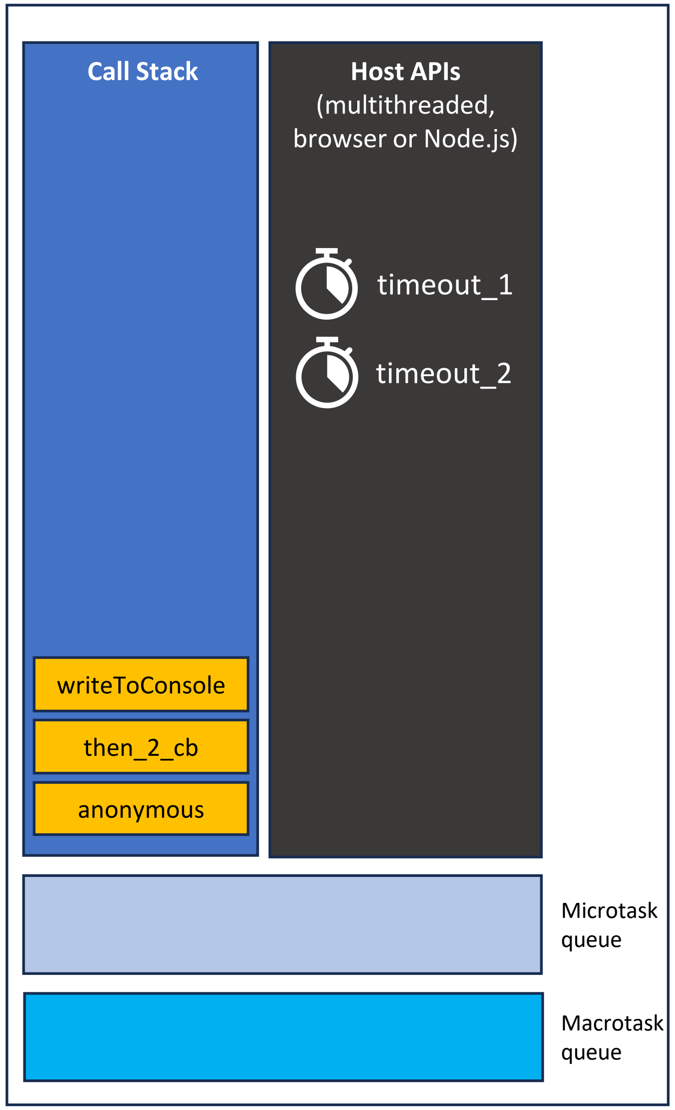
```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {   message = "then 2"
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```

Paused on breakpoint

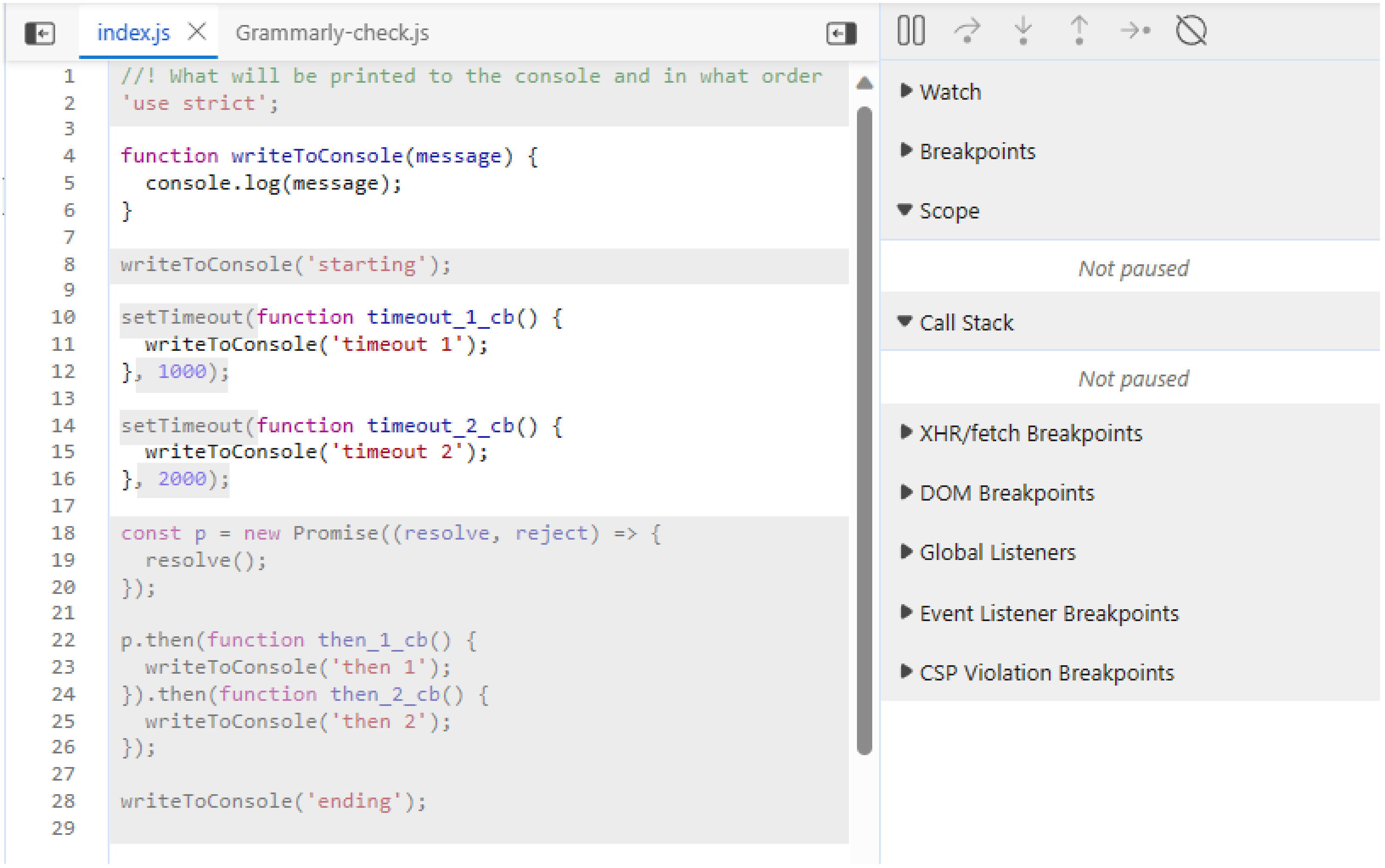
- Watch
- Breakpoints
- Scope
  - Local
    - this: undefined
    - message: "then 2"
  - Script
    - p: Promise {<fulfilled>: undefined}
  - Global Window
- Call Stack

writeToConsole	index.js:5
then_2_cb	index.js:25
Promise.then (async)	
(anonymous)	index.js:24

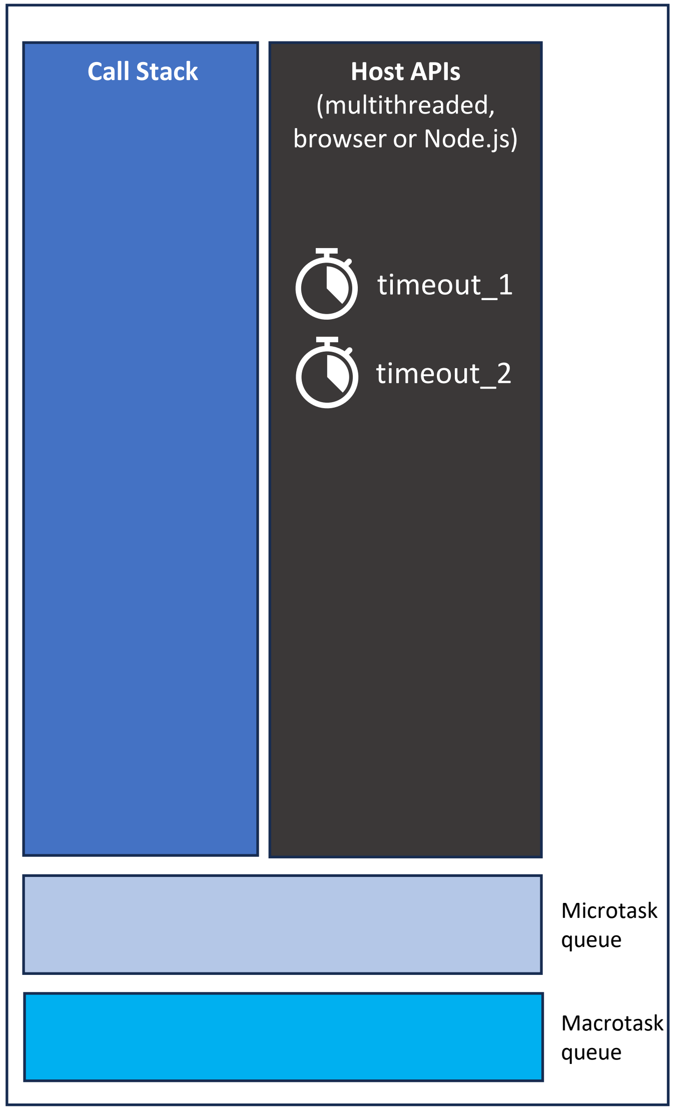
- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints
- CSP Violation Breakpoints



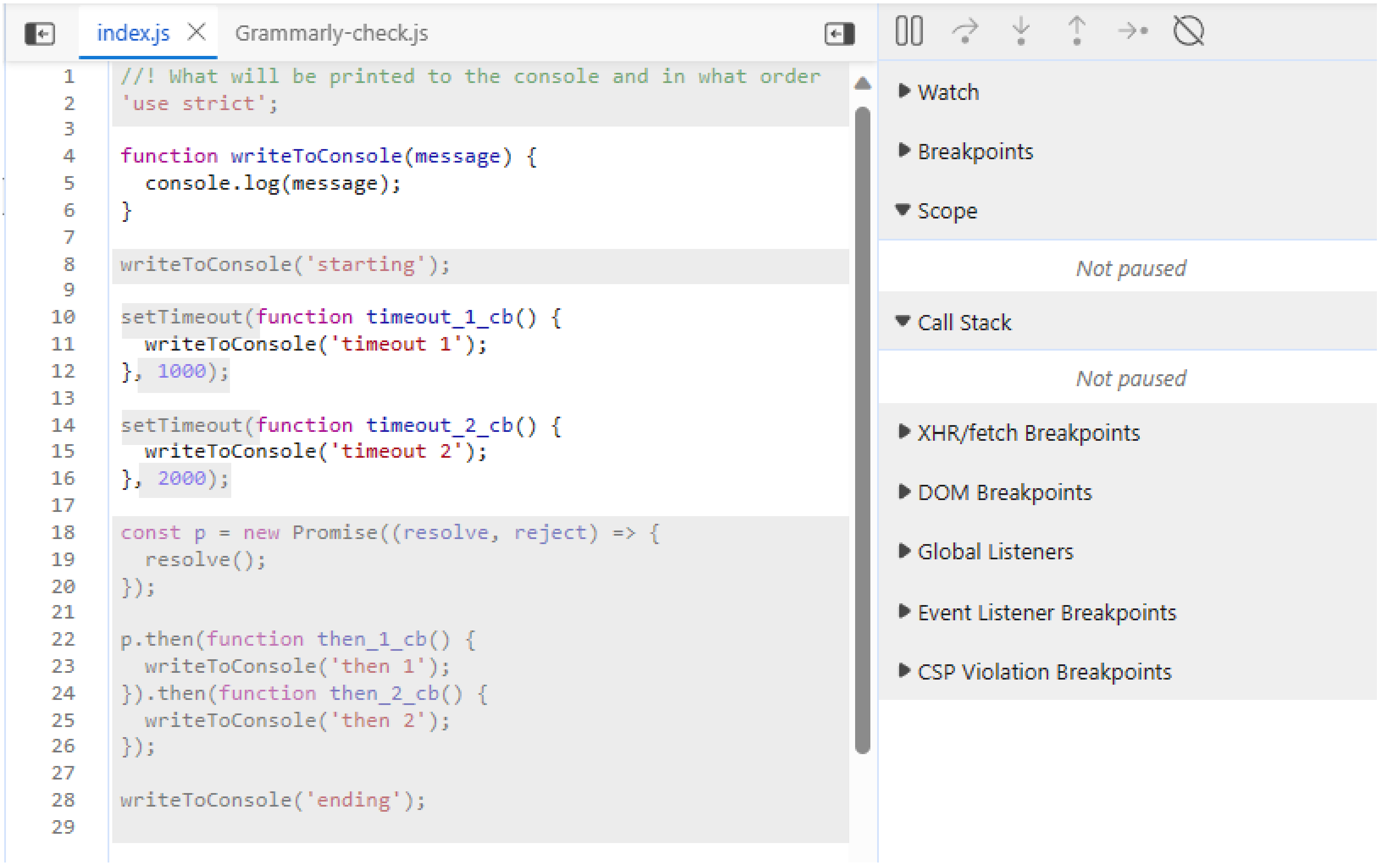
Idle, awaiting timeout_1



```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```



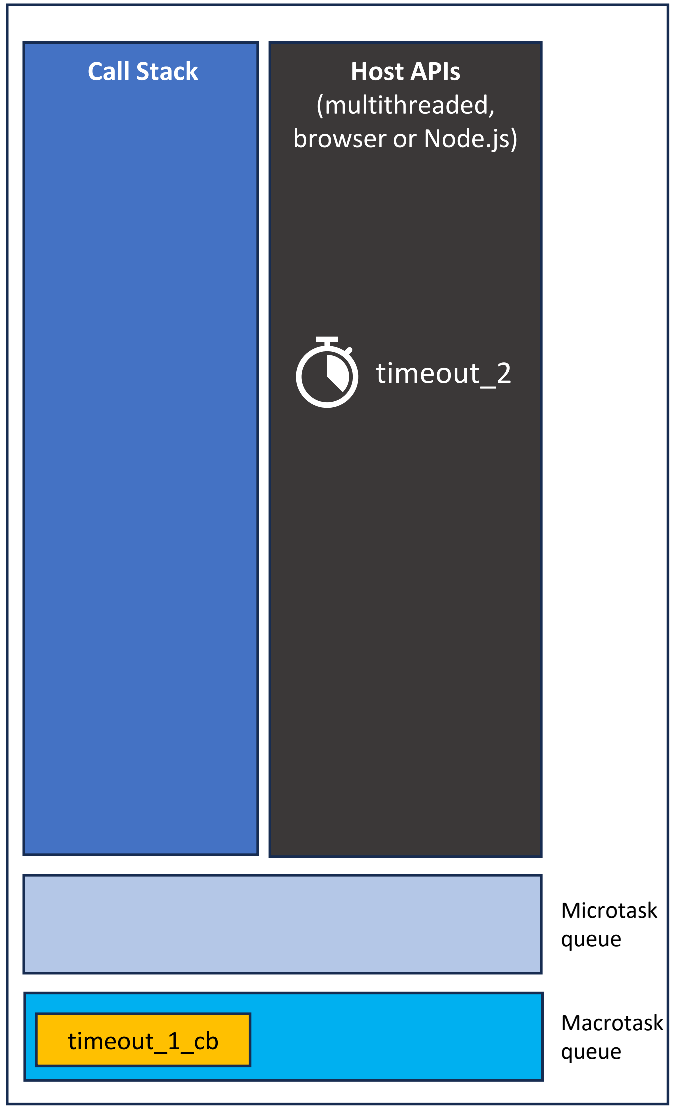
timeout_1 fired



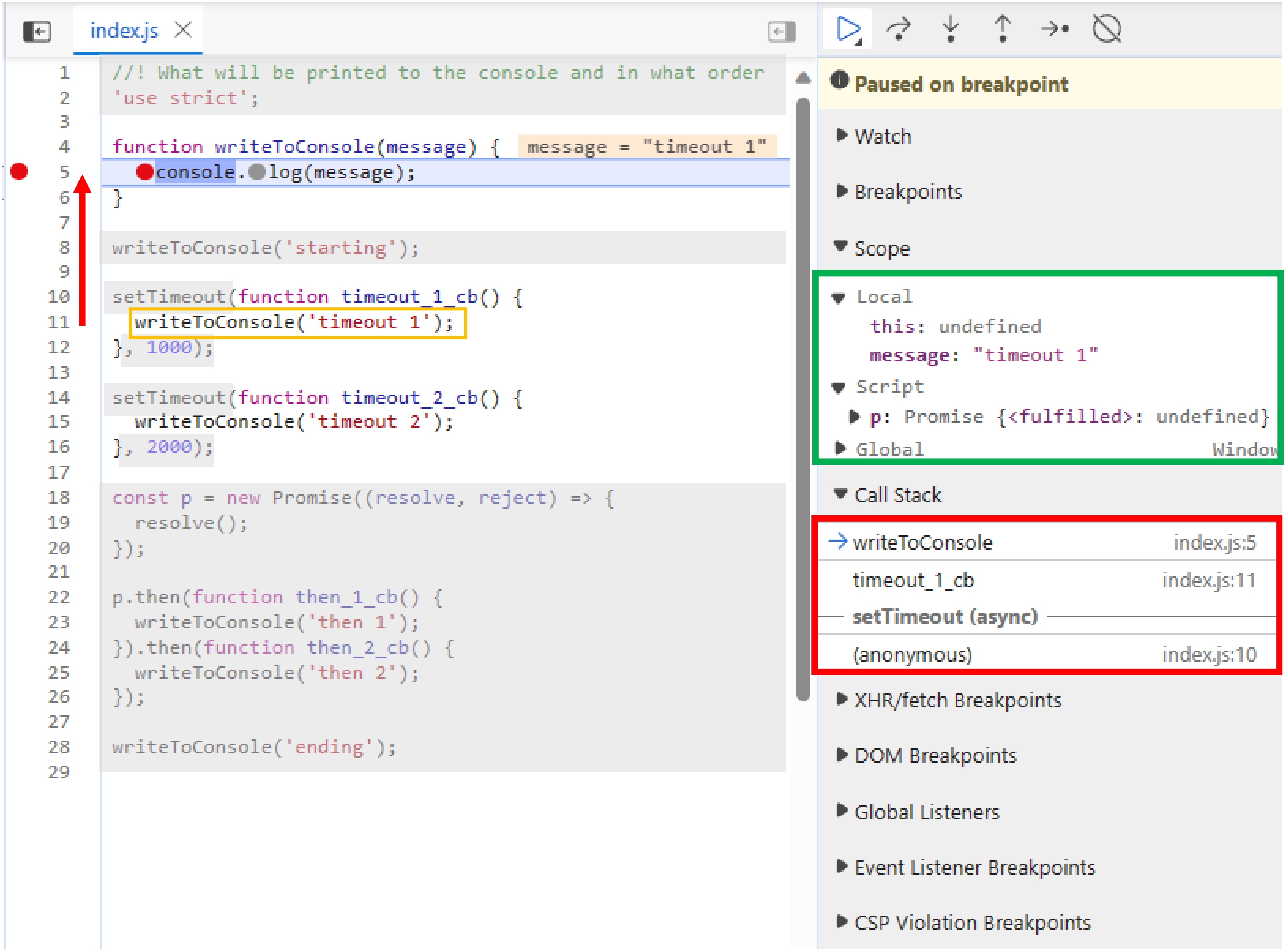
index.js × Grammarly-check.js

```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```

▶ Watch
▶ Breakpoints
▼ Scope
Not paused
▼ Call Stack
Not paused
▶ XHR/fetch Breakpoints
▶ DOM Breakpoints
▶ Global Listeners
▶ Event Listener Breakpoints
▶ CSP Violation Breakpoints



Event 4: Macrotask timeout_1_cb



index.js

```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {   message = "timeout 1"
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```

Paused on breakpoint

Watch

Breakpoints

Scope

```
Local
  this: undefined
  message: "timeout 1"
Script
  p: Promise {<fulfilled>: undefined}
Global                                    Window
```

Call Stack

writeToConsole	index.js:5
timeout_1_cb	index.js:11
setTimeout (async)	
(anonymous)	index.js:10

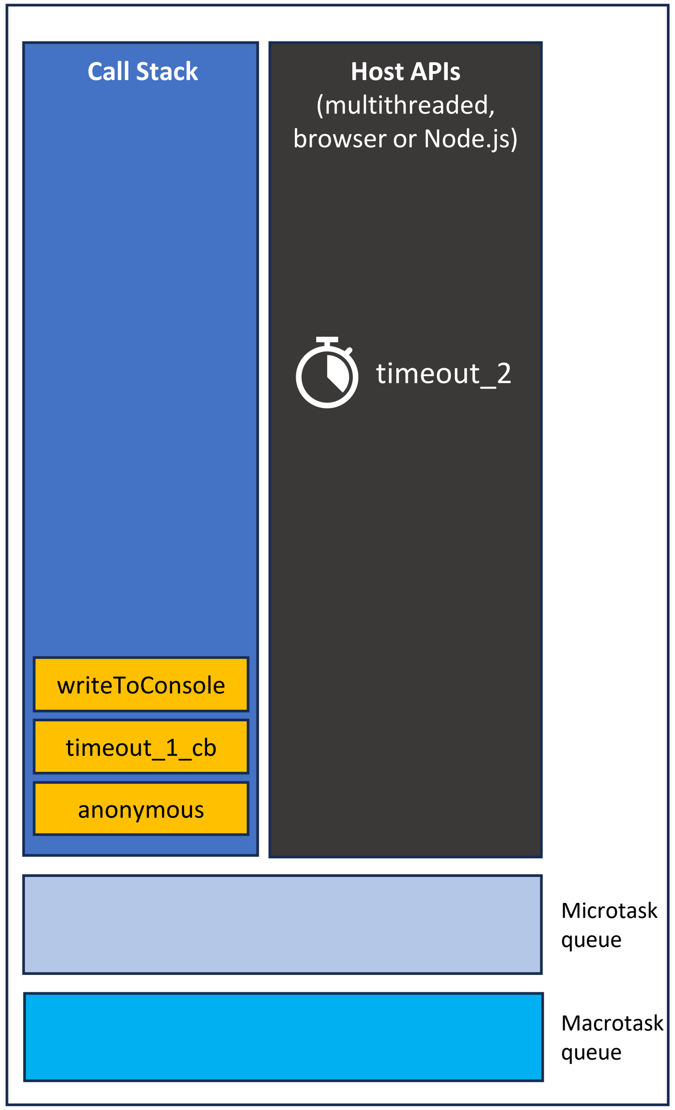
XHR/fetch Breakpoints

DOM Breakpoints

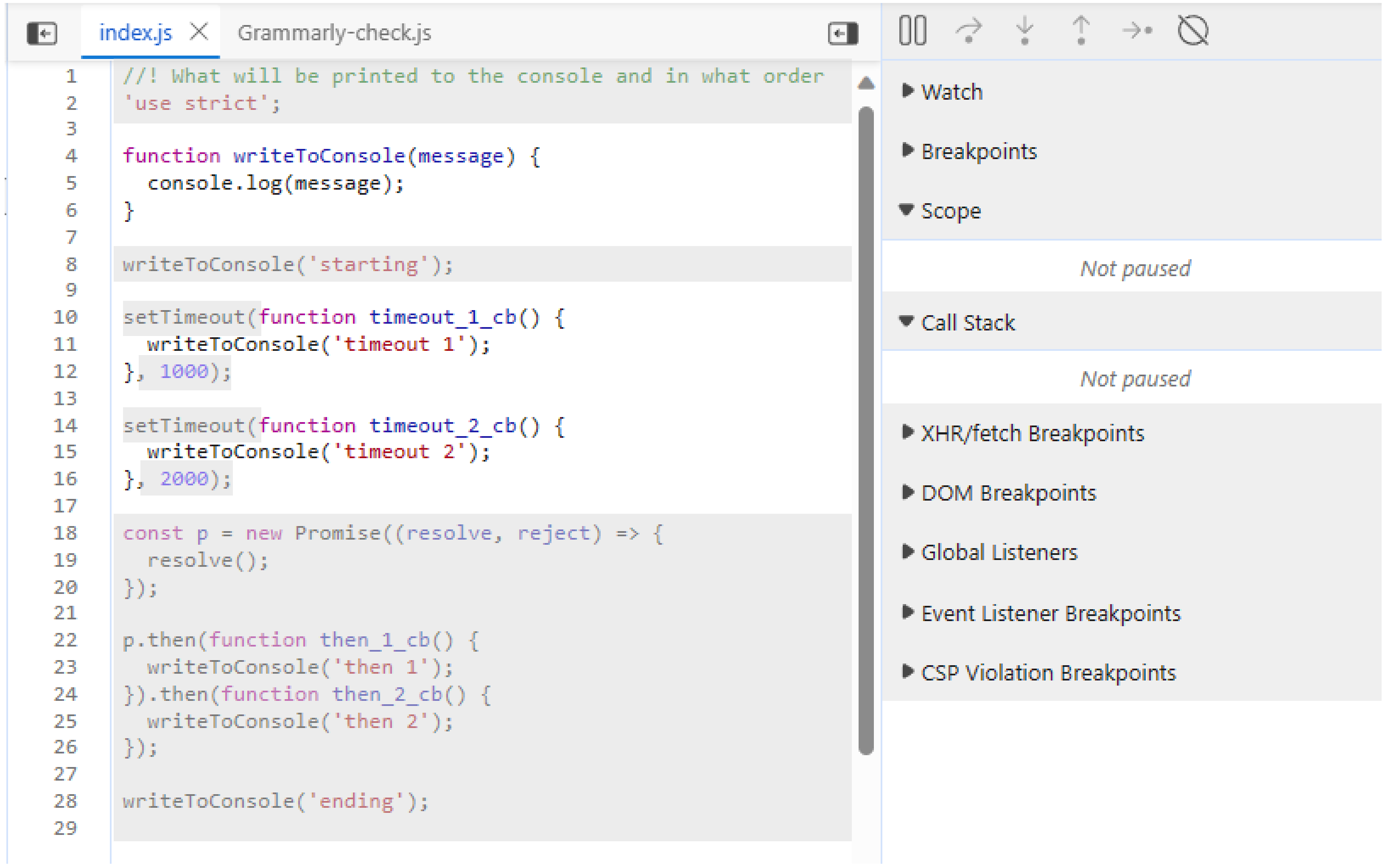
Global Listeners

Event Listener Breakpoints

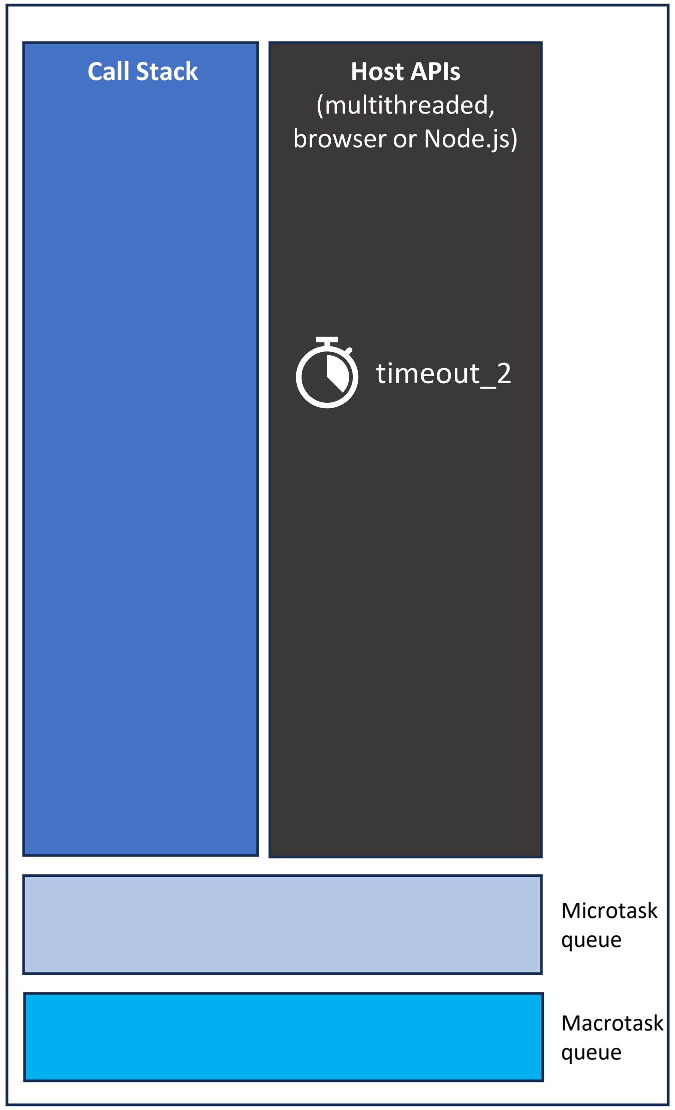
CSP Violation Breakpoints



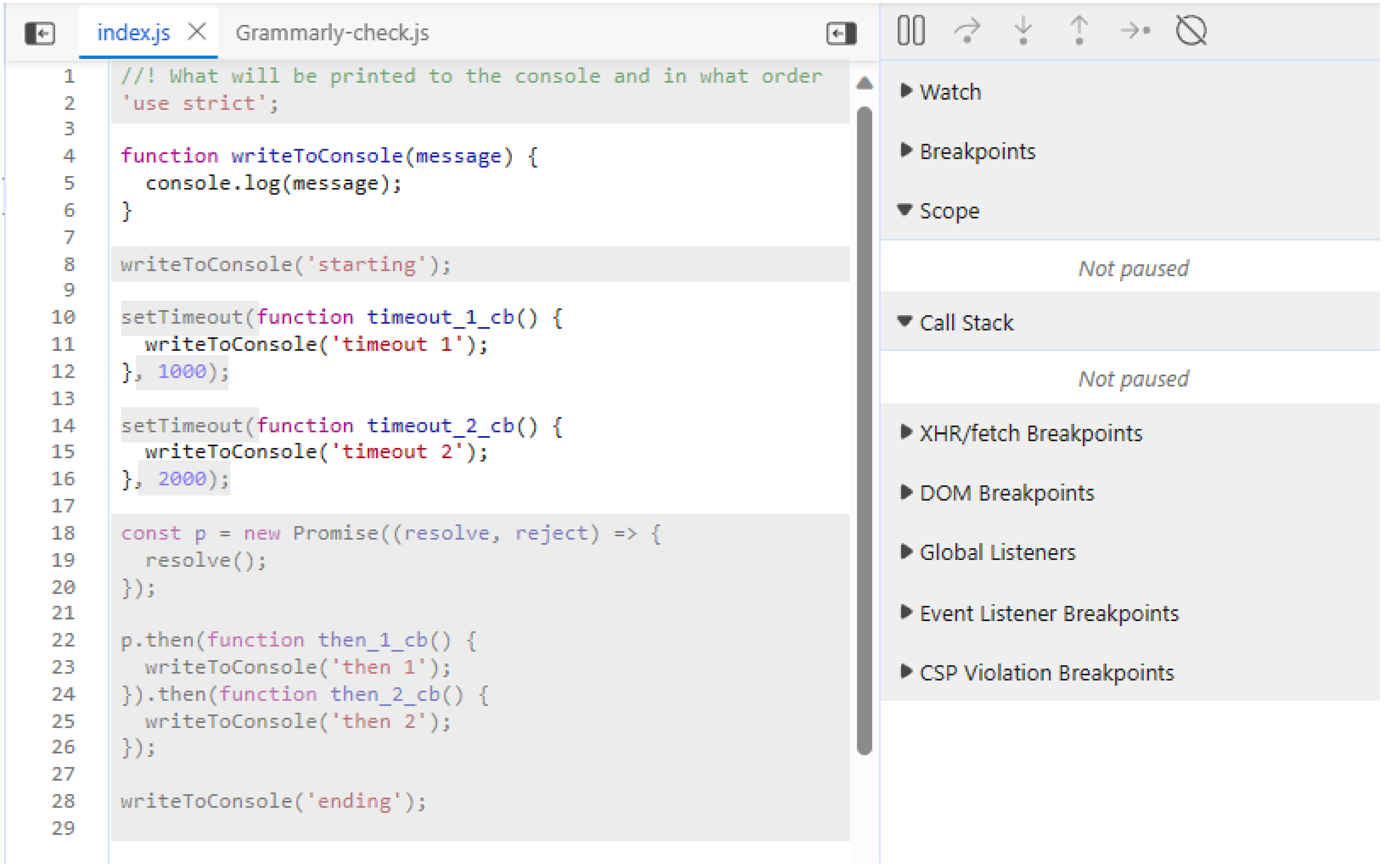
Idle 2, awaiting timeout_2



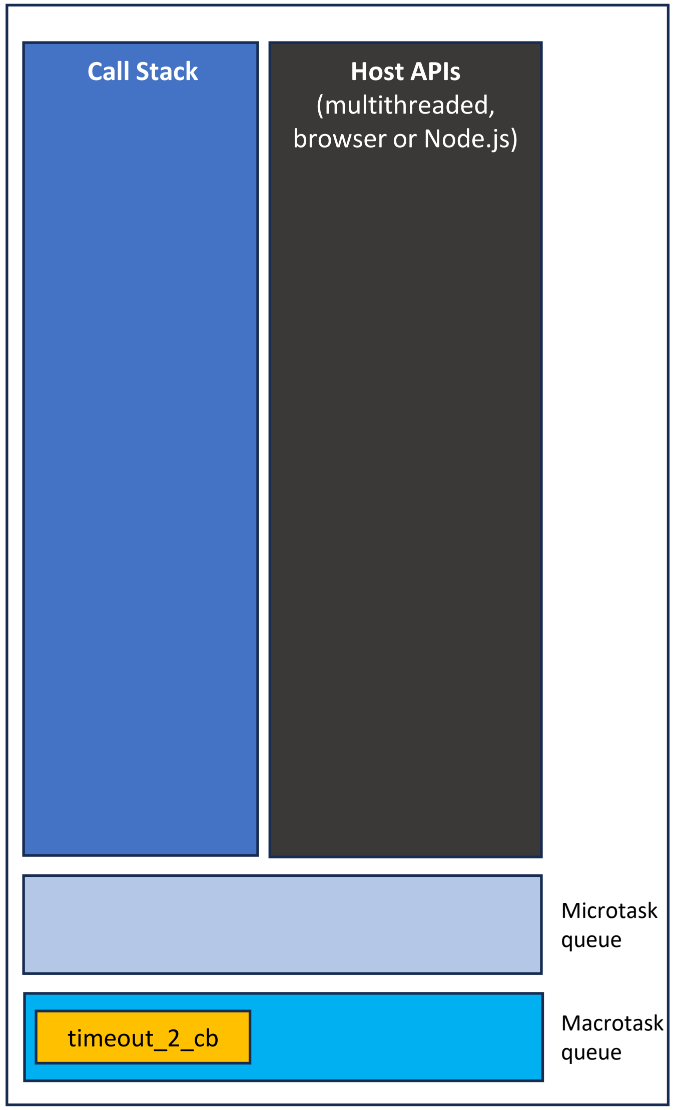
```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');

```



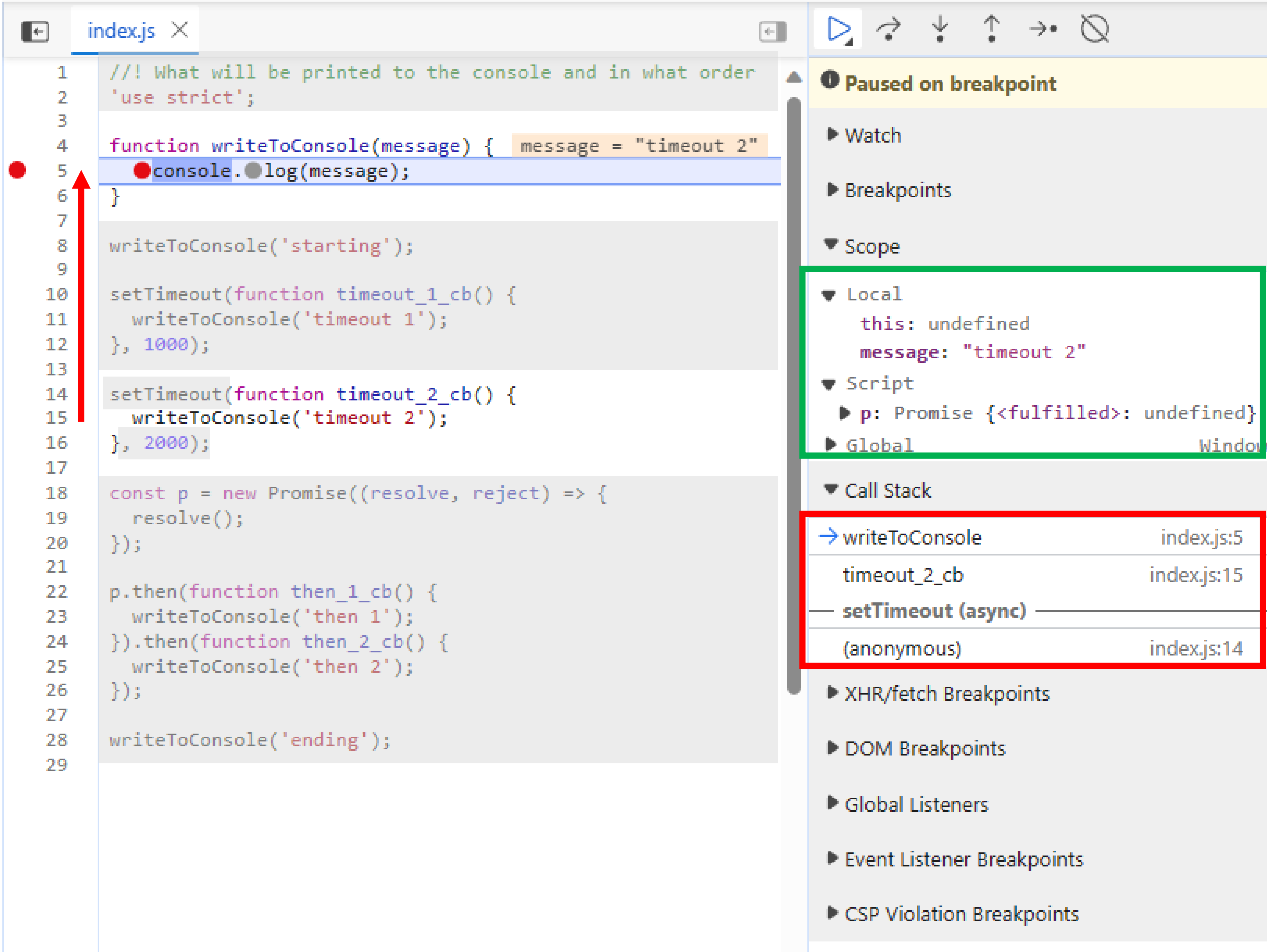
timeout_2 fired



```javascript
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');
```



Event 5: Macrotask timeout_2_callback



index.js

```
 1  //! What will be printed to the console and in what order
 2  'use strict';
 3
 4  function writeToConsole(message) {   message = "timeout 2"
 5    console.log(message);
 6  }
 7
 8  writeToConsole('starting');
 9
10  setTimeout(function timeout_1_cb() {
11    writeToConsole('timeout 1');
12  }, 1000);
13
14  setTimeout(function timeout_2_cb() {
15    writeToConsole('timeout 2');
16  }, 2000);
17
18  const p = new Promise((resolve, reject) => {
19    resolve();
20  });
21
22  p.then(function then_1_cb() {
23    writeToConsole('then 1');
24  }).then(function then_2_cb() {
25    writeToConsole('then 2');
26  });
27
28  writeToConsole('ending');
29
```

Paused on breakpoint

▶ Watch

▶ Breakpoints

▼ Scope

```
▼ Local
    this: undefined
    message: "timeout 2"
▼ Script
  ▶ p: Promise {<fulfilled>: undefined}
▶ Global                                  Windo
```

▼ Call Stack

→ writeToConsole	index.js:5
timeout_2_cb	index.js:15
setTimeout (async)	
(anonymous)	index.js:14

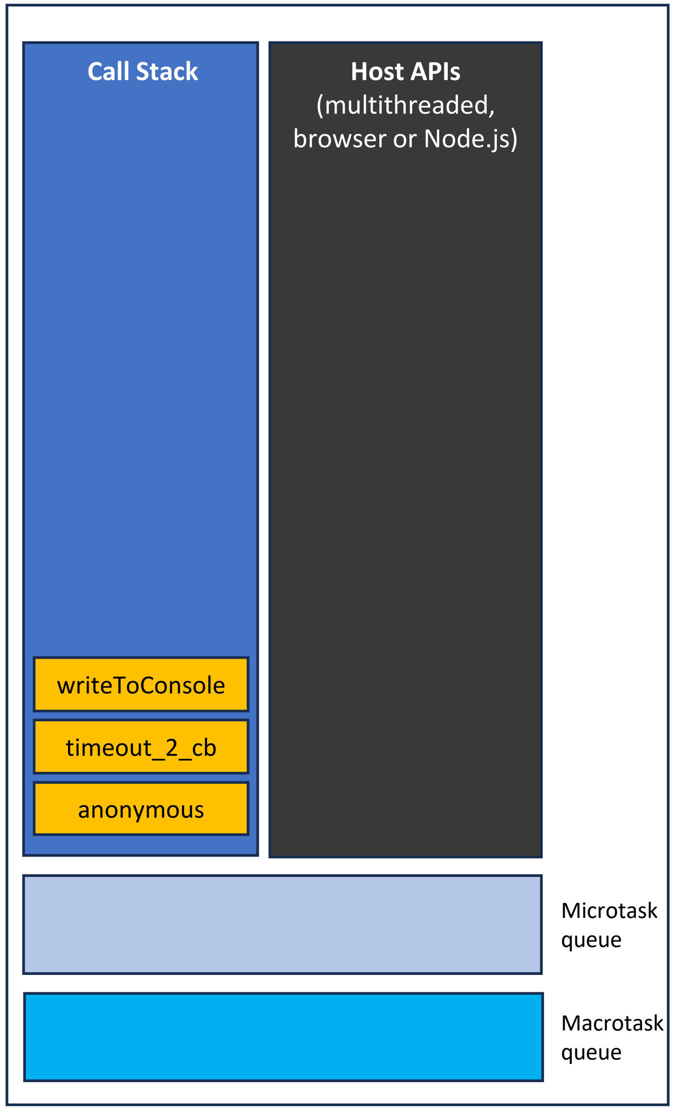
▶ XHR/fetch Breakpoints

▶ DOM Breakpoints

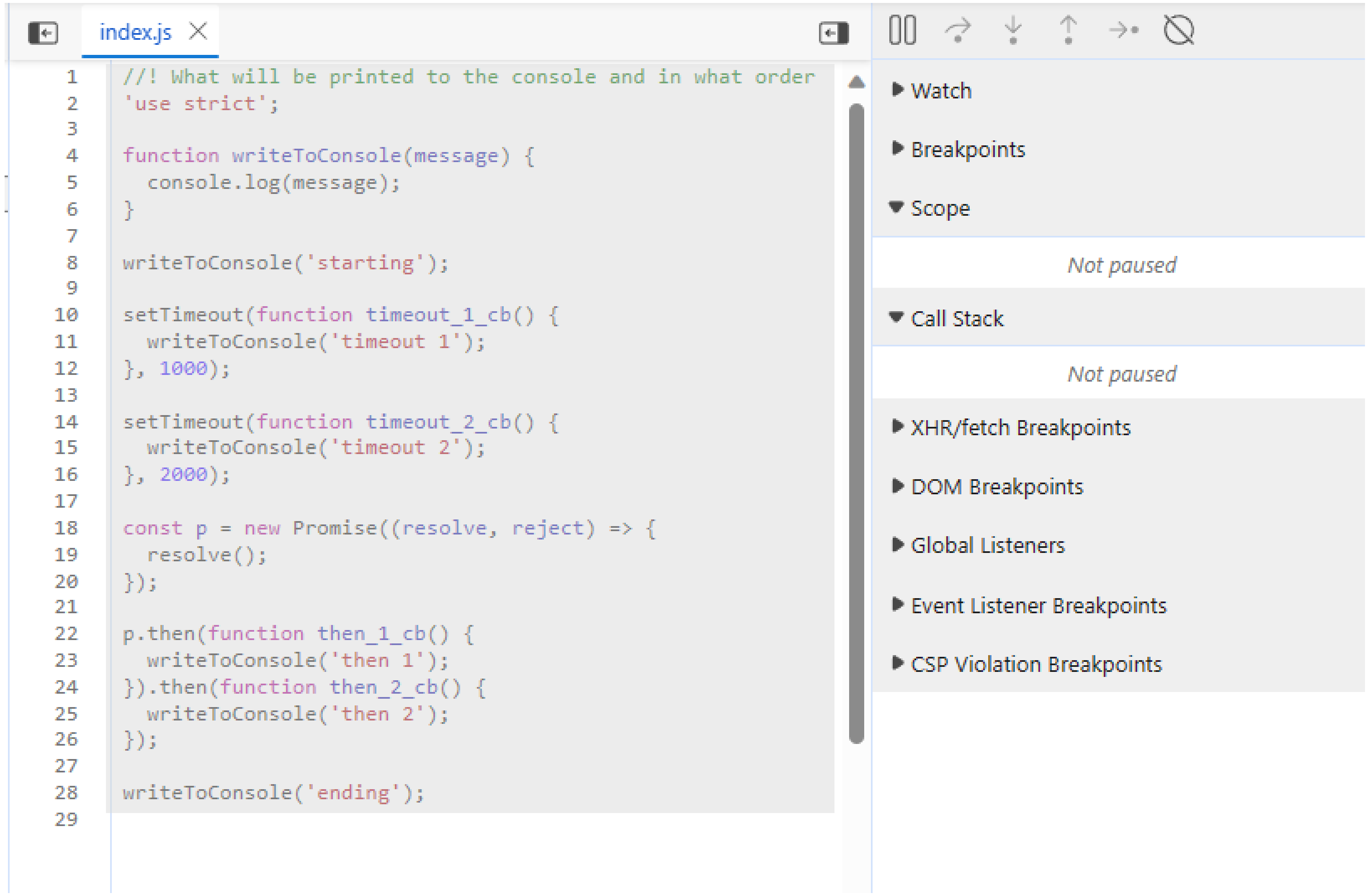
▶ Global Listeners

▶ Event Listener Breakpoints

▶ CSP Violation Breakpoints

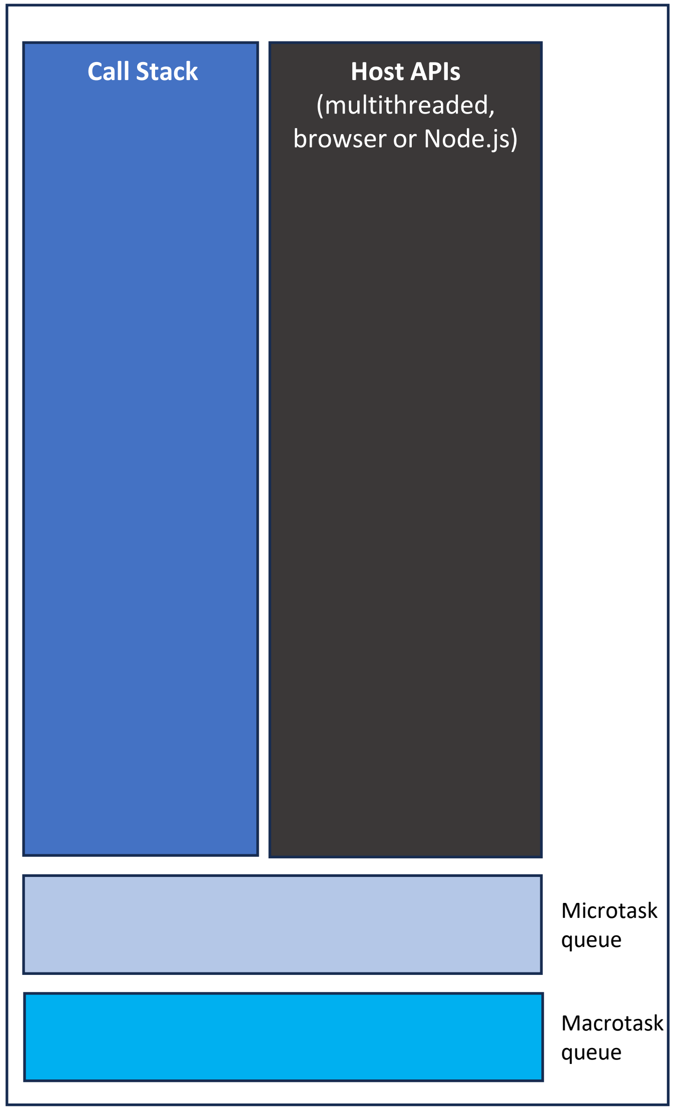


Program finished


index.js

```
//! What will be printed to the console and in what order
'use strict';

function writeToConsole(message) {
  console.log(message);
}

writeToConsole('starting');

setTimeout(function timeout_1_cb() {
  writeToConsole('timeout 1');
}, 1000);

setTimeout(function timeout_2_cb() {
  writeToConsole('timeout 2');
}, 2000);

const p = new Promise((resolve, reject) => {
  resolve();
});

p.then(function then_1_cb() {
  writeToConsole('then 1');
}).then(function then_2_cb() {
  writeToConsole('then 2');
});

writeToConsole('ending');

```

Watch
Breakpoints
Scope
Not paused
Call Stack
Not paused
XHR/fetch Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints
CSP Violation Breakpoints


Call Stack
Host APIs (multithreaded, browser or Node.js)
Microtask queue
Macrotask queue