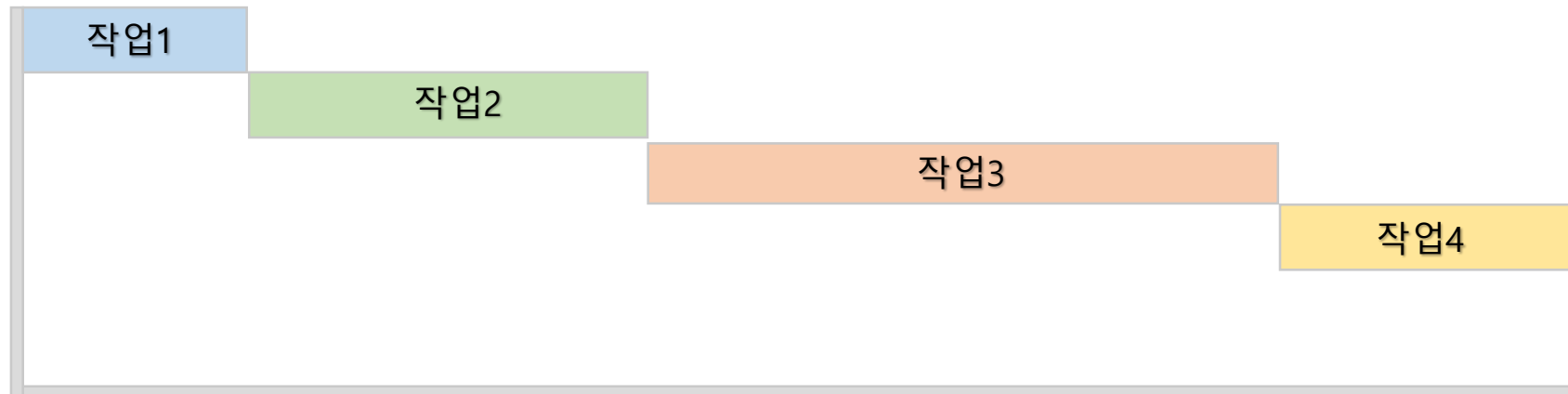


비동기

비동기

- 동기적(Synchronous)

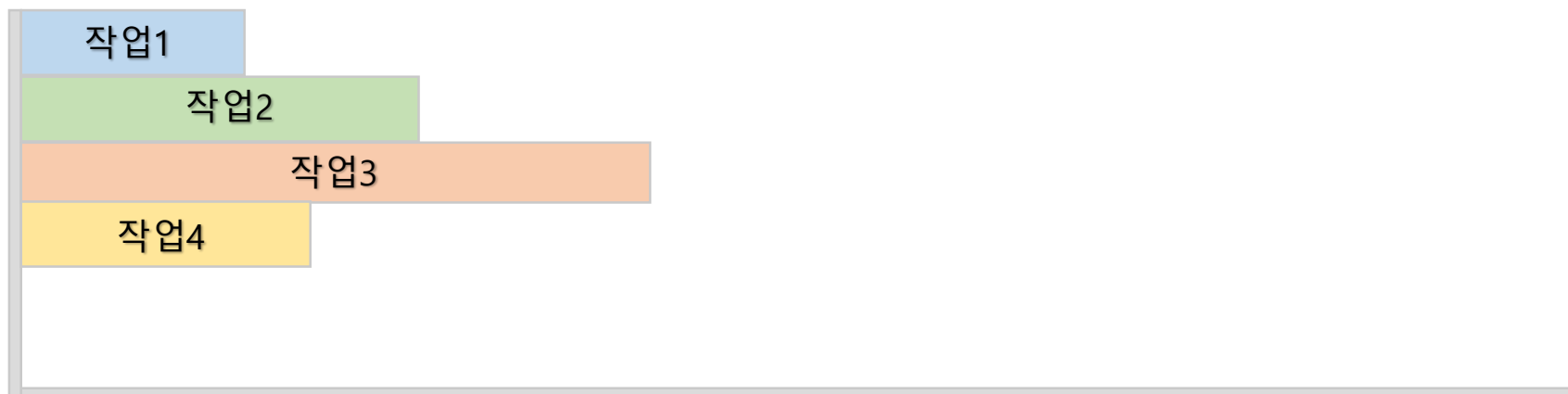
동기적은 하나의 작업이 모두 수행될 때 까지 다른 작업을 수행 하지 못함



비동기

- 비동기적(Asynchronous)

비동기는 작업의 결과를 기다리는 동안 다른 작업을 미리 수행하는 것
오래 걸리는 작업들에 대해 백그라운드에서 처리되며 그동안 다른 작업을 계속
수행할 수 있도록 함



비동기

- 비동기 작업의 대표적인 예시와 함수/문법

1. Ajax, setTimeout
2. Callback 함수, Promise, async/await

- Background vs Foreground

Background : 사용자가 보이지 않는 뒷 단에서 실행되는 작업

→ ex) 특정 앱 닫기 버튼을 눌렀을 때 사용자는 닫힌 걸로 보이지만 뒤에서 계속 실행되고 있는 상태

Foreground : 사용자가 보이는 앞 단에서 실행되는 작업

→ ex) 프로그램을 다운 받으려 할 때 설치창이 나오고 사용자가 Next를 눌러야 다음 화면으로 넘어가는 등, 사용자가 보고 명령을 내릴 수 있는 상태

비동기

• 비동기 확인하기

예상했던 코드의 동작과 콘솔 로그에 찍히는 로그의 차이점 확인하기
→ 비동기적

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

setTimeout

흐름 파악하기 - 1

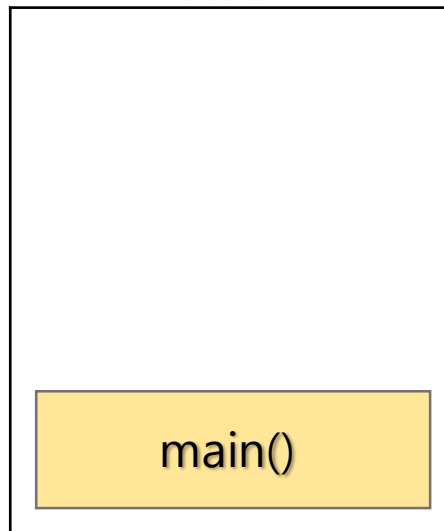
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

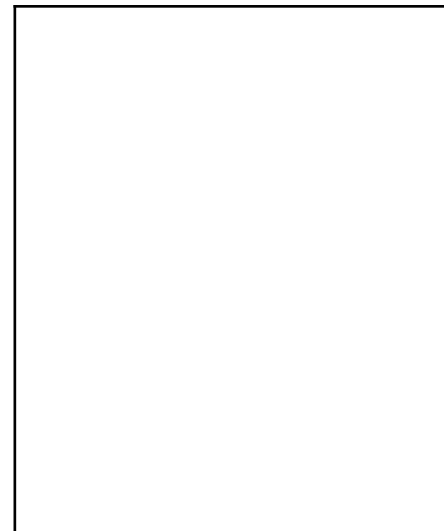
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



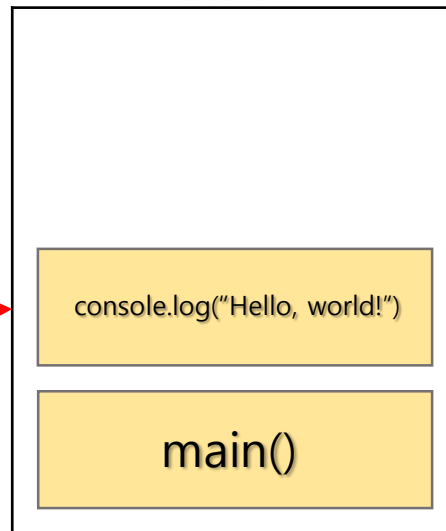
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

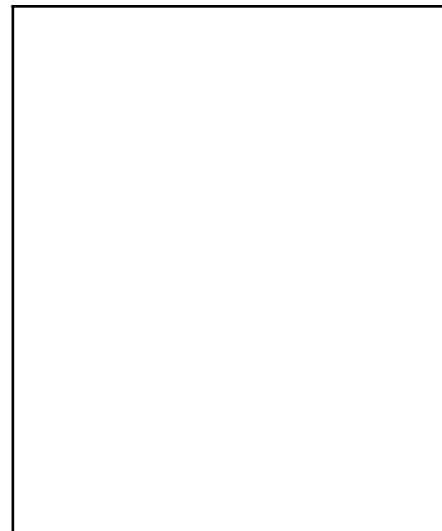
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



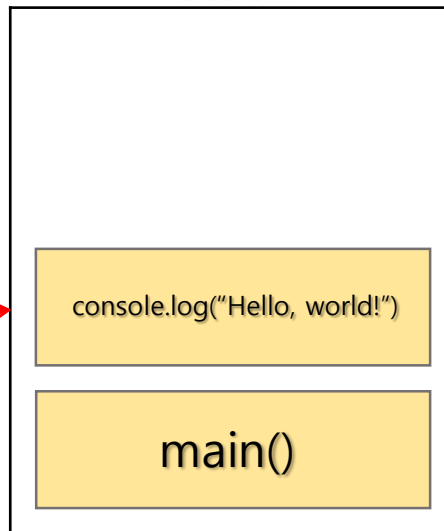
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

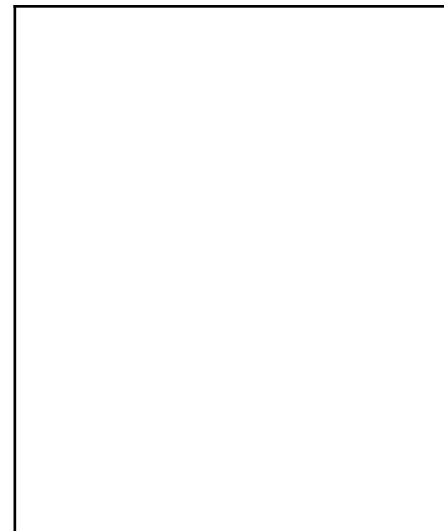
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack

main()

Web APIs

console

Hello, world!

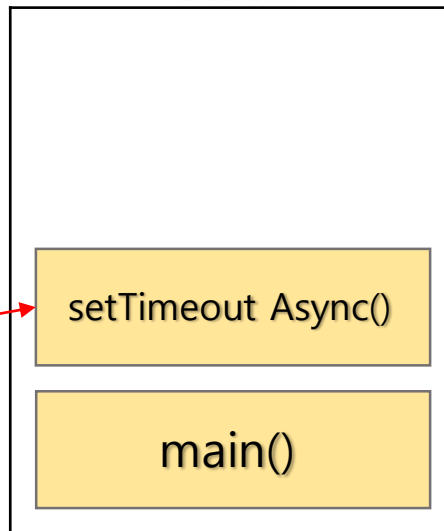
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



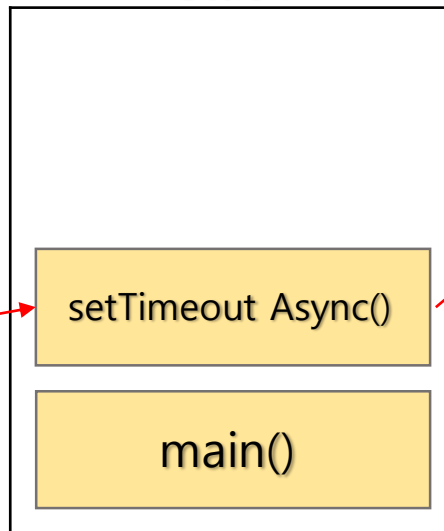
console



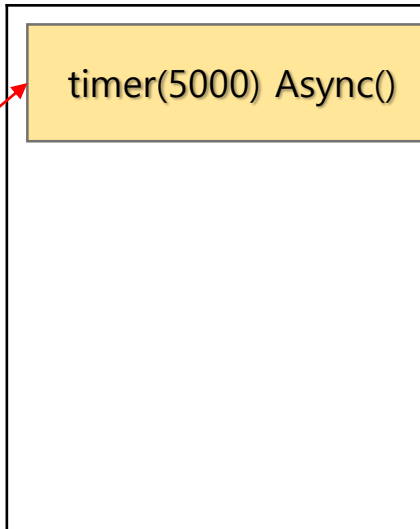
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);
    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



5초
카운트다운
시작

console

Hello, world!

setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

Stack

setTimeout 호출이 완료되었으므로
스택에서 제거

main()

Web APIs

timer(5000) Async()

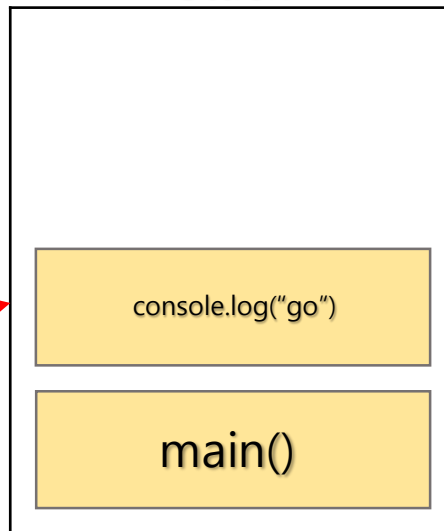
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

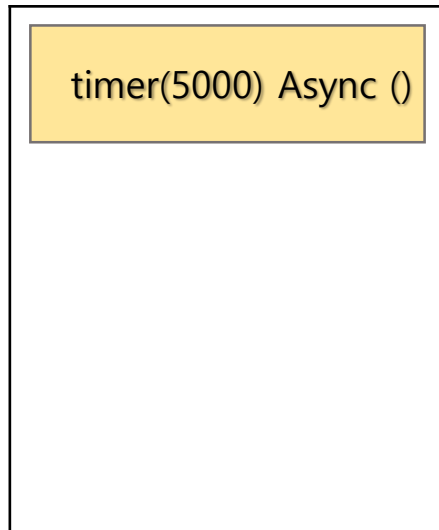
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



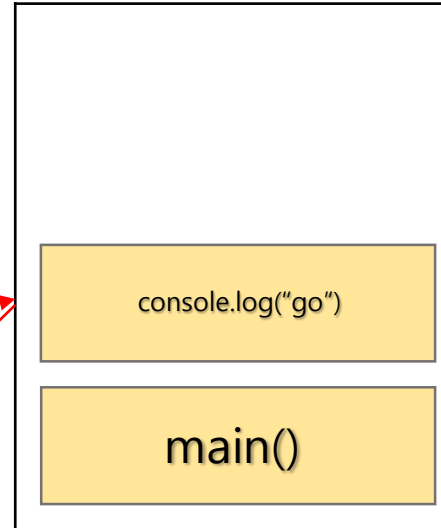
setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

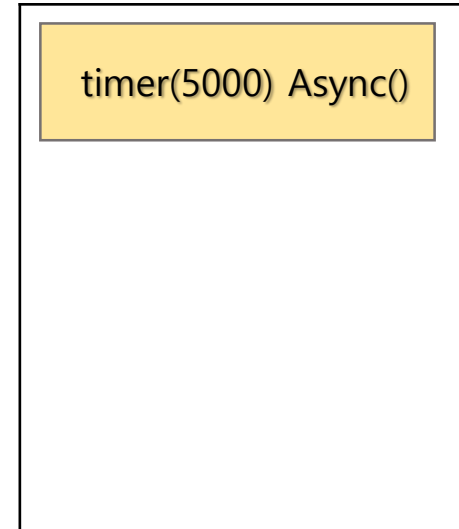
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack

main()

Web APIs

timer(5000) Async()

console

Hello, world!

go

setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

Web APIs

timer(5000) Async()

Task Queue

Async()

타이머가 종료되어 (5초가 지남)
Task Queue로 이동

setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

Web APIs

Task Queue

Async()

setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

Web APIs

Event Loop



* Stack과 Task Queue를 주시함.
Stack이 비어지고, 대기중인 Task Queue가 있을 경우

Task Queue

Async()

setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

Stack

Async()

Web APIs

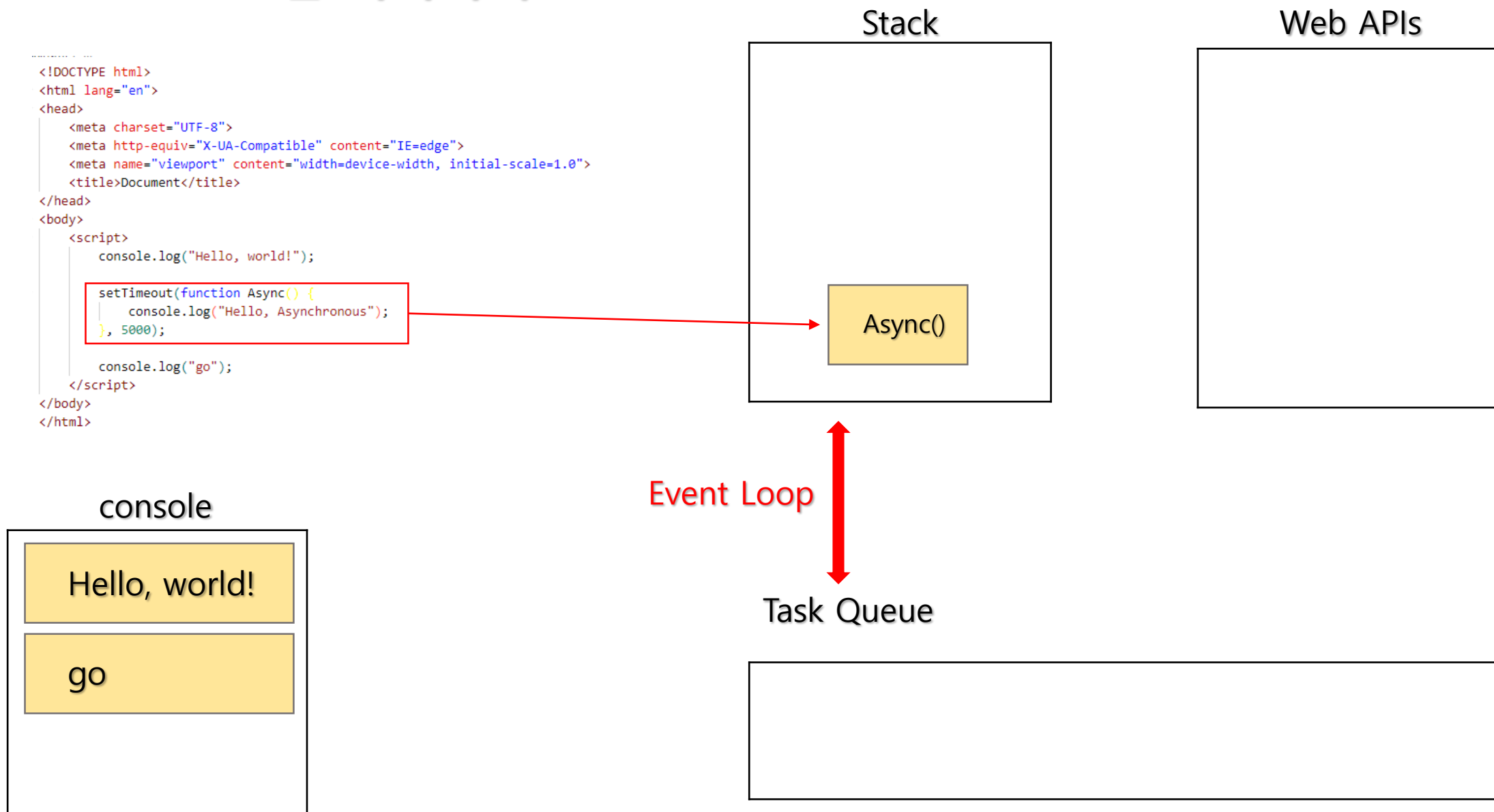
console

Hello, world!

go

Event Loop

Task Queue

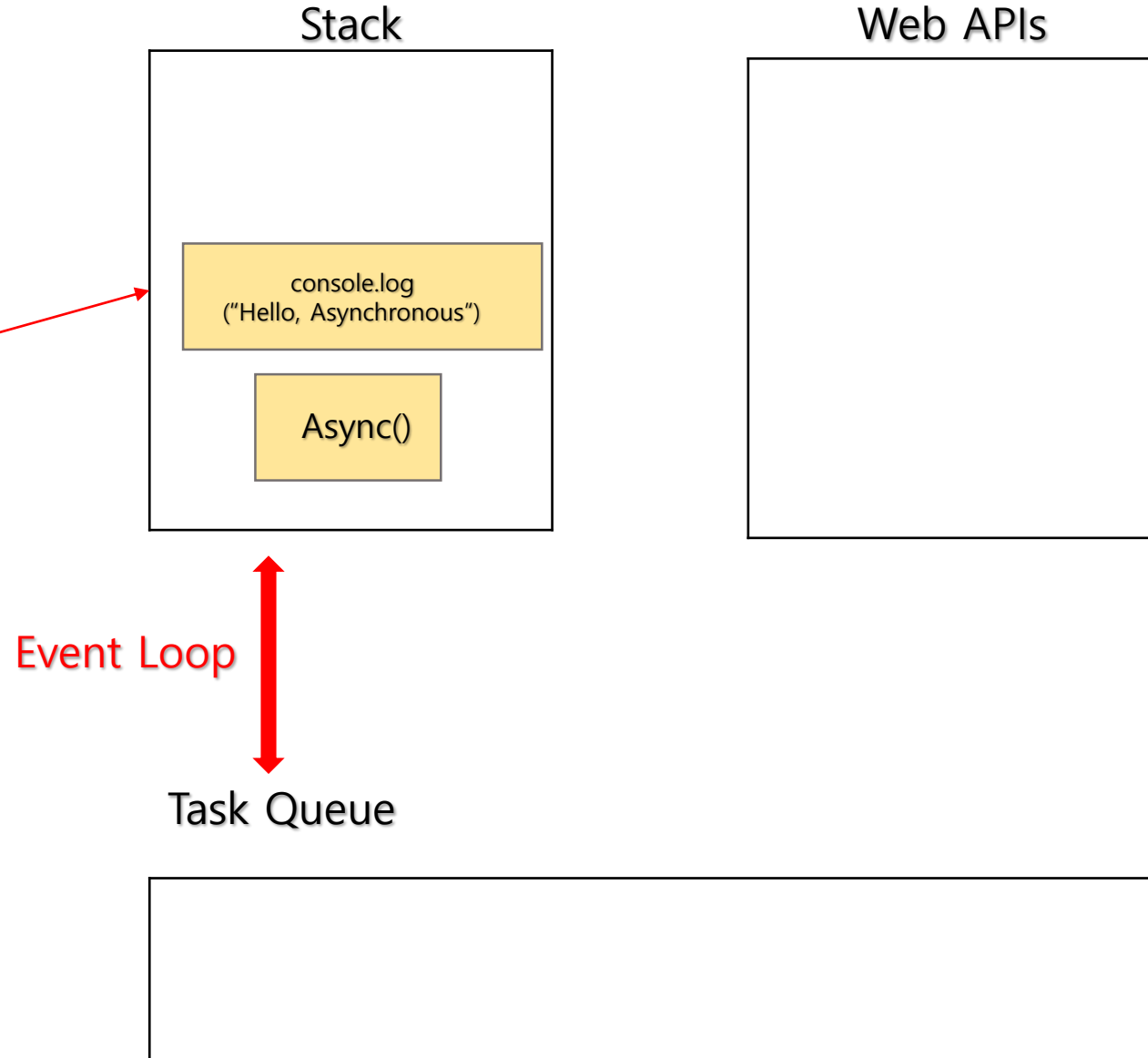
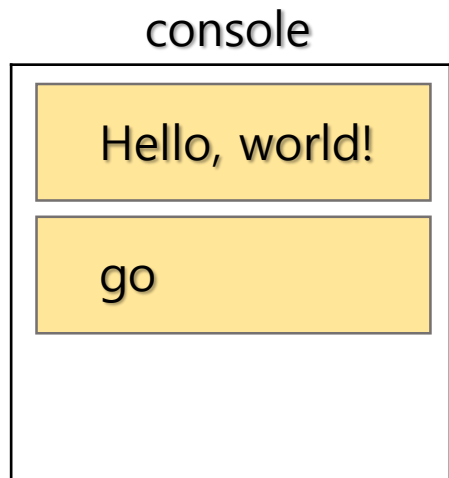


setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

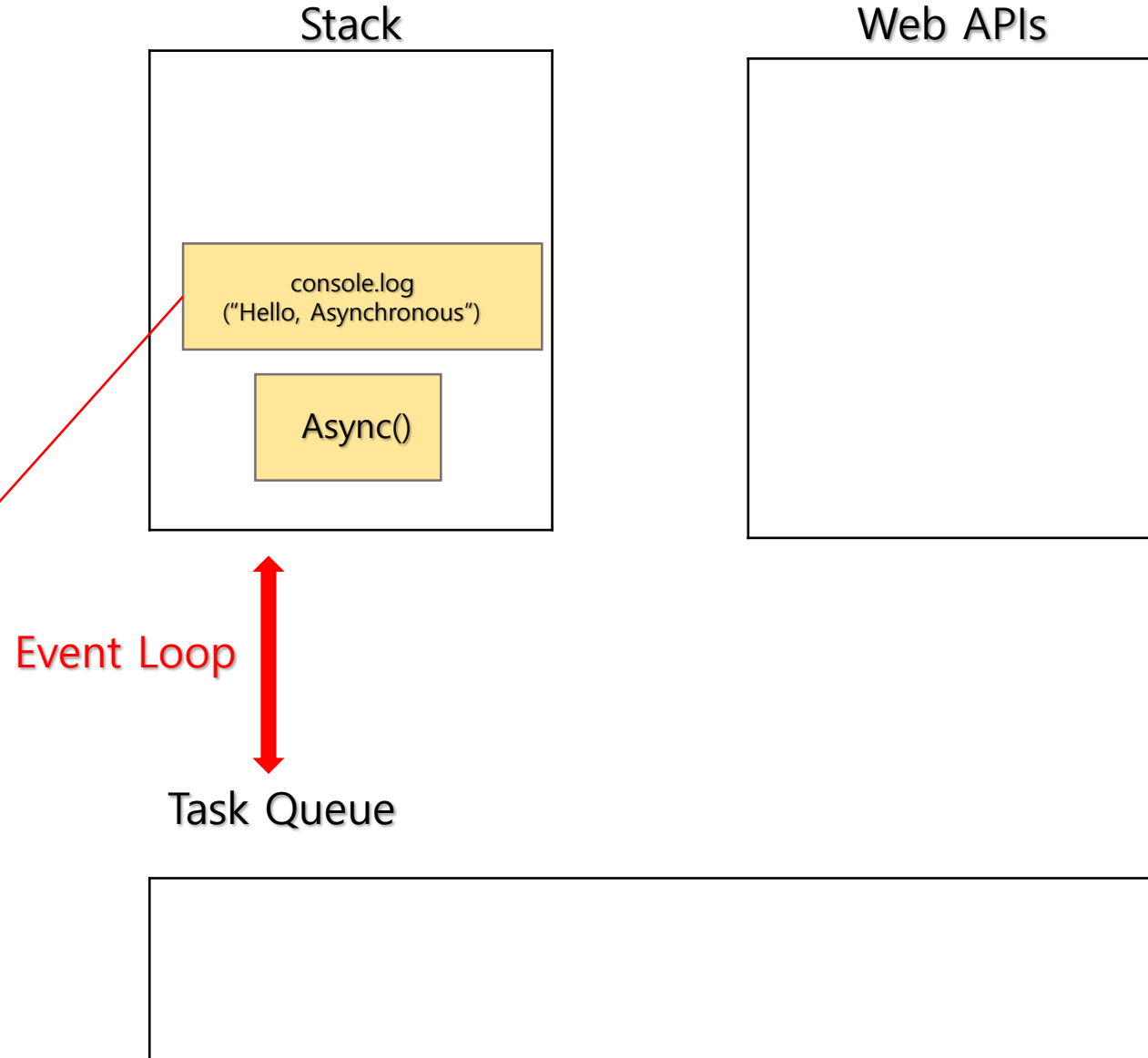
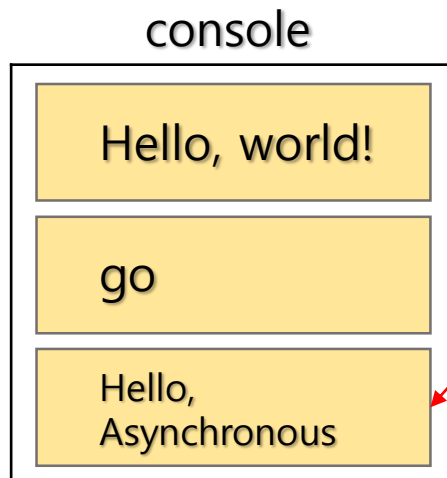


setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```



setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Hello,
Asynchronous

Stack

Async()

Web APIs

Event Loop

Task Queue



setTimeout 흐름 파악하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 5000);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Hello,
Asynchronous

Stack

Web APIs

Event Loop

Task Queue



setTimeout

흐름 파악하기 - 2

setTimeout 흐름 파악하기 - 2

- 상황

setTimeout이 0초로 설정되어 있다면?

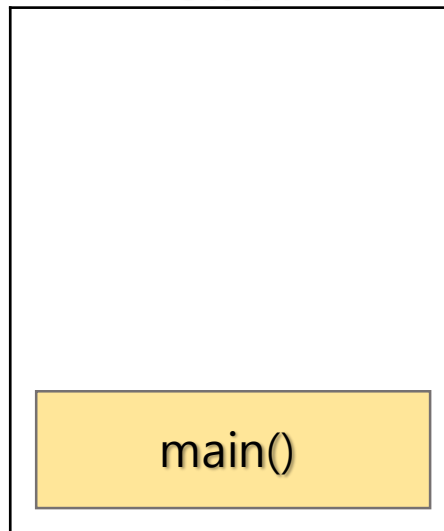
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

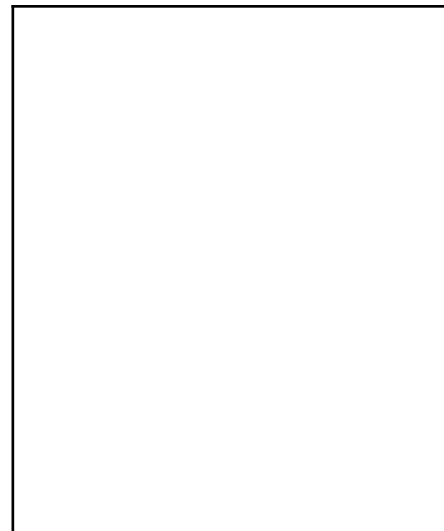
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



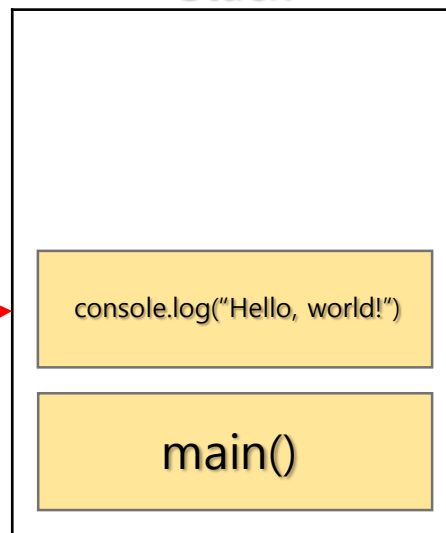
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

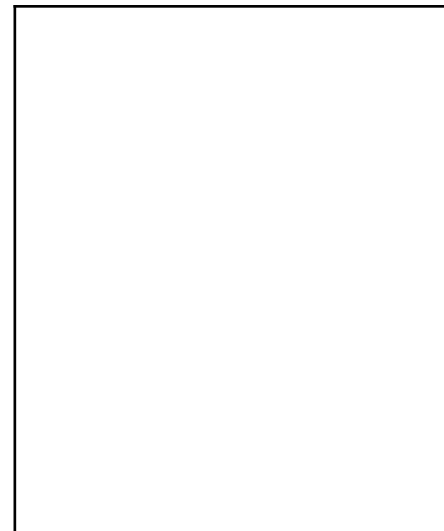
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



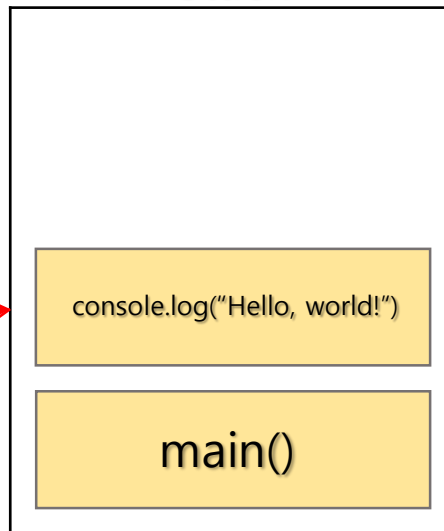
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

Stack

main()

Web APIs

setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);
    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

Stack

setTimeout Async()

main()

Web APIs

timer(0) Async()

Task Queue

Async()

카운트다운 없이
바로 TaskQueue로 이동

setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

Stack

main()

Web APIs

Task Queue

Async()

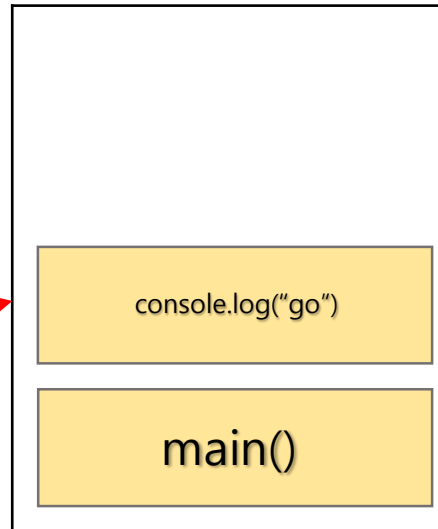
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

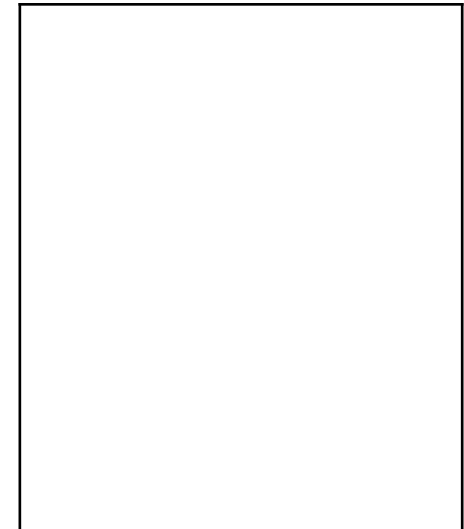
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



Task Queue



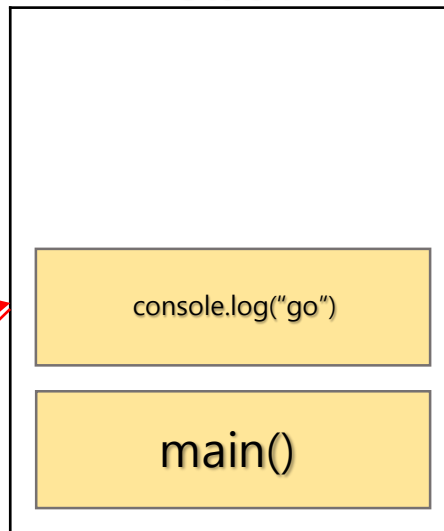
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

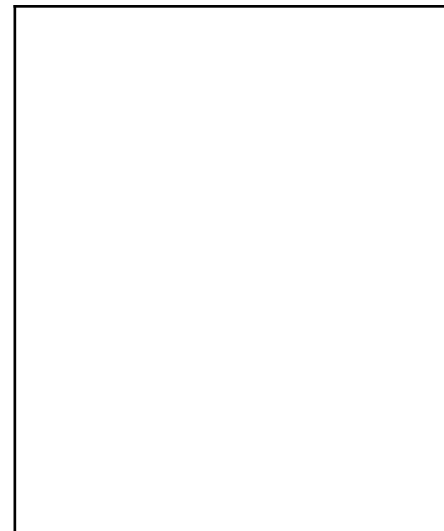
    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



console



Task Queue



setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

main()

Web APIs

Task Queue

Async()

setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

Web APIs

Task Queue

Async()

setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Stack

Web APIs

Event Loop



* Stack이 비어지고, 대기중인 Task Queue가 있기 때문에 이동

Task Queue

Async()

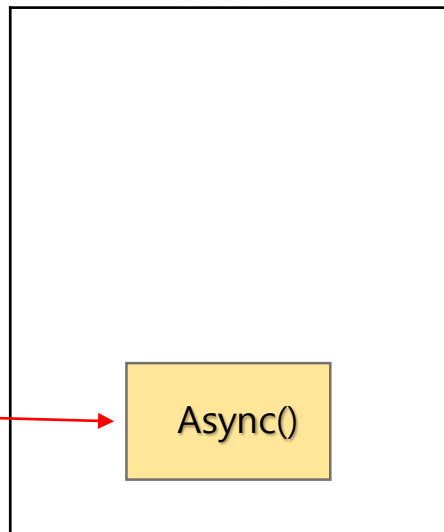
setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

Stack



Web APIs



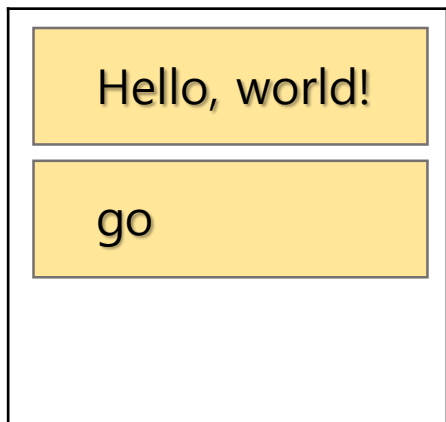
Event Loop



Task Queue



console

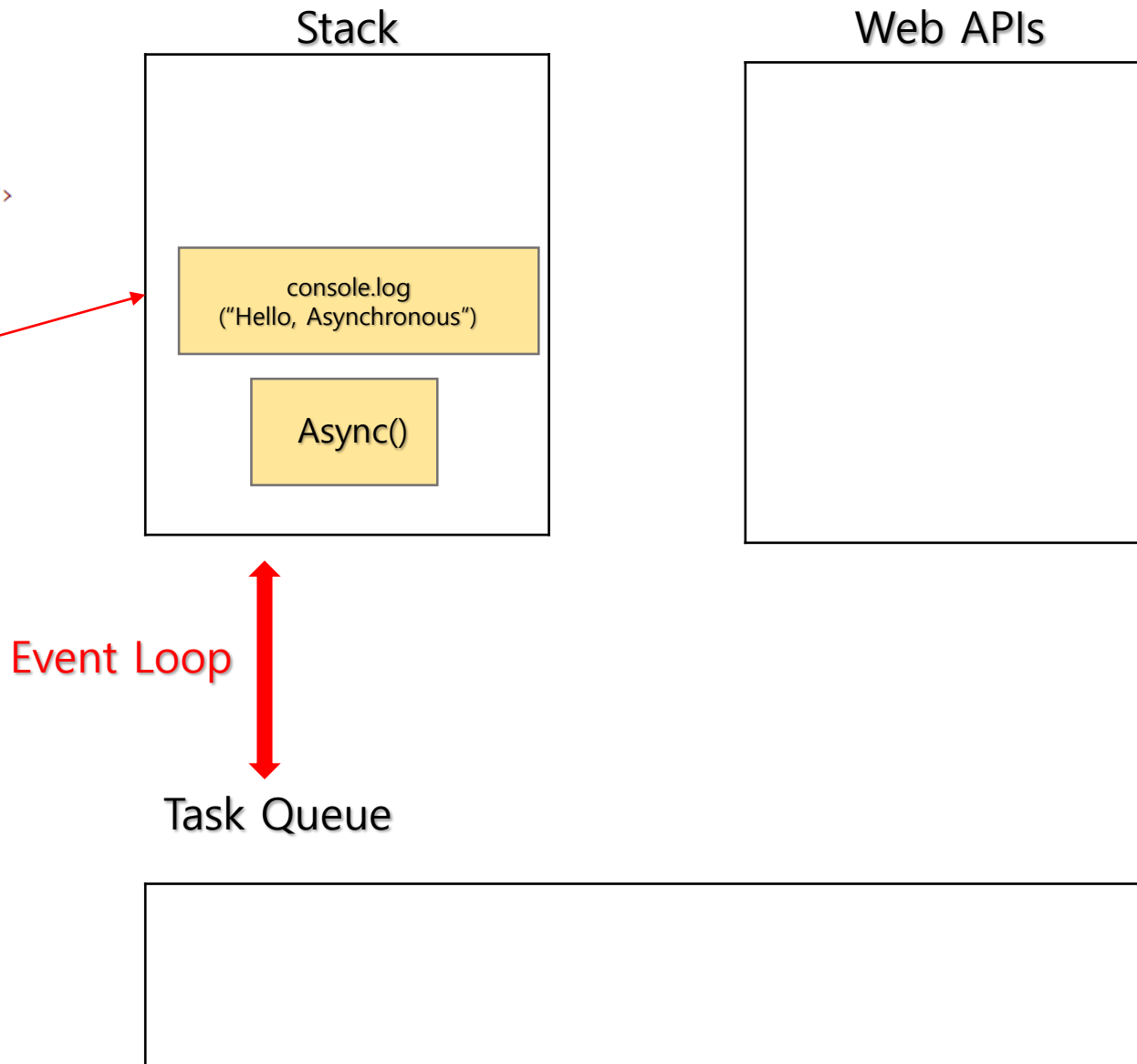
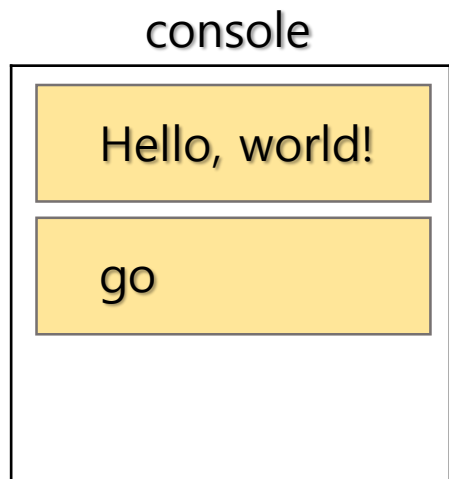


setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

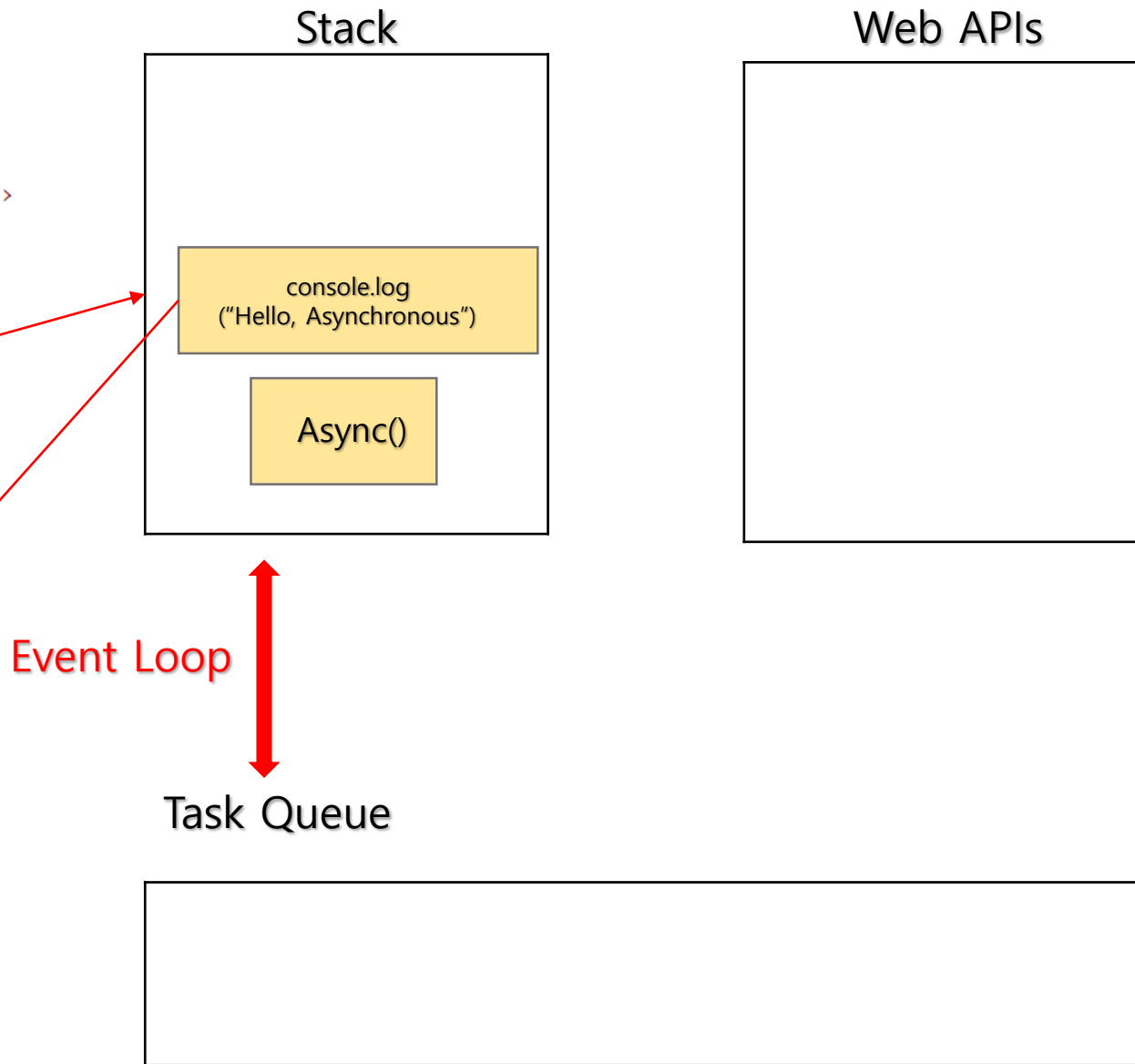
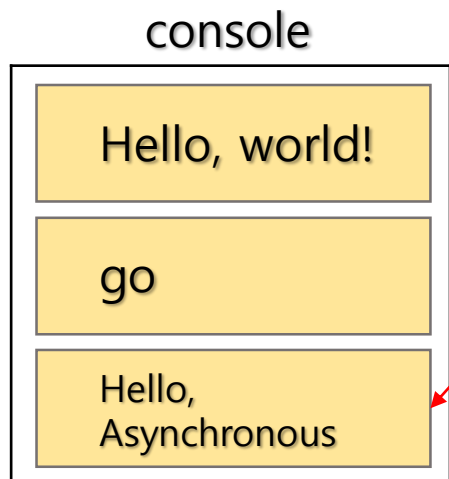


setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```



setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Hello,
Asynchronous

Stack

Async()

Web APIs

Event Loop

Task Queue



setTimeout 흐름 파악하기 - 2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    console.log("Hello, world!");

    setTimeout(function Async() {
      console.log("Hello, Asynchronous");
    }, 0);

    console.log("go");
  </script>
</body>
</html>
```

console

Hello, world!

go

Hello,
Asynchronous

Stack

Web APIs

Event Loop

Task Queue



setTimeout 흐름 파악하기 - 2

- 결론

0초로 설정되어 있더라도 앞에서 실행되는 코드를 기다려야 하므로
코드가 실제로 실행 되는 건 0초가 아니고 $+\alpha$ 된 시간이며, 아래와 같은 코드들이 있더라도 모두 1초에 실행 되지 않음

```
setTimeout(function Async() {  
  console.log("Hello, Asynchronous");  
}, 1000);
```

```
setTimeout(function Async() {  
  console.log("Hello, Asynchronous");  
}, 1000);
```

```
setTimeout(function Async() {  
  console.log("Hello, Asynchronous");  
}, 1000);
```

```
setTimeout(function Async() {  
  console.log("Hello, Asynchronous");  
}, 1000);
```