

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

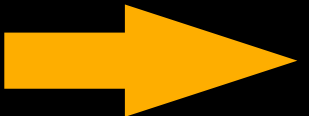
call stack

Web APIs

event loop



callback queue



```
console.log('A');  
  
setTimeout(function() {  
  console.log('B');  
}, 1000);  
  
setTimeout(function() {  
  console.log('C');  
}, 500);  
  
console.log('D');
```

call stack

console.log('A')

Web APIs

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A



call stack

Web APIs

event loop



callback queue

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

➔ 

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
setTimeout(function() {  
  console.log('C');  
}, 500);
```

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

A

call stack

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

Web APIs

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A

call stack

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

**1000 밀리세컨 기다림**

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

➡ setTimeout(function() {
  console.log('C');
}, 500);

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

A

call stack

```
setTimeout(function() {
  console.log('C');
}, 500);
```

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A

call stack

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

```
setTimeout(function() {
  console.log('C');
}, 500);
```

500 밀리세컨 기다림

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);
```

→ console.log('D');

A

call stack

console.log('D')

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

```
setTimeout(function() {
  console.log('C');
}, 500);
```

event loop



callback queue



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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D ←

call stack

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

```
setTimeout(function() {
  console.log('C');
}, 500);
```

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D

call stack

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

event loop



callback queue

```
setTimeout(function() {
  console.log('C');
}, 500);
```

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D

call stack

```
setTimeout(function() {
  console.log('C');
}, 500);
```

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D  
C ←

call stack

Web APIs

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D  
C

call stack

Web APIs

event loop



callback queue

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D  
C

call stack

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

Web APIs

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D  
C  
B ←

call stack

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

Web APIs

event loop



callback queue

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

setTimeout(function() {
  console.log('B');
}, 1000);

setTimeout(function() {
  console.log('C');
}, 500);

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

A  
D  
C  
B

call stack

Web APIs

event loop



callback queue





```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

call stack

Web APIs

callback queue



```

→ console.log("A");

setTimeout(function() {
  console.log("B");
}, 0);

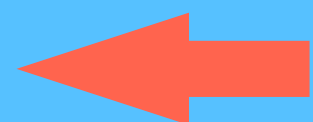
console.log("C");

```



```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

A



call stack

Web APIs

event loop



callback queue

```

console.log("A");
➡ setTimeout(function() {
  console.log("B");
}, 0);
console.log("C");
  
```

A



```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

A

call stack

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 0);
```

event loop



callback queue

```
console.log("A");
```

```
setTimeout(function() {  
  console.log("B");  
}, 0);
```

➡ 

```
console.log("C");
```

A

call stack

```
console.log('C')
```

Web APIs

event loop



callback queue

```
setTimeout(function() {  
  console.log('B');  
}, 0);
```

```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

A  
C ←

call stack

Web APIs



callback queue

```
setTimeout(function() {  
  console.log('B');  
}, 0);
```



```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

A  
C

call stack

```
setTimeout(function() {  
  console.log('B');  
}, 0);
```

Web APIs

event loop



callback queue

```
console.log("A");  
  
setTimeout(function() {  
  console.log("B");  
}, 0);  
  
console.log("C");
```

A  
C  
B ←

call stack

Web APIs

callback queue





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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

call stack

Web APIs

event loop



callback queue

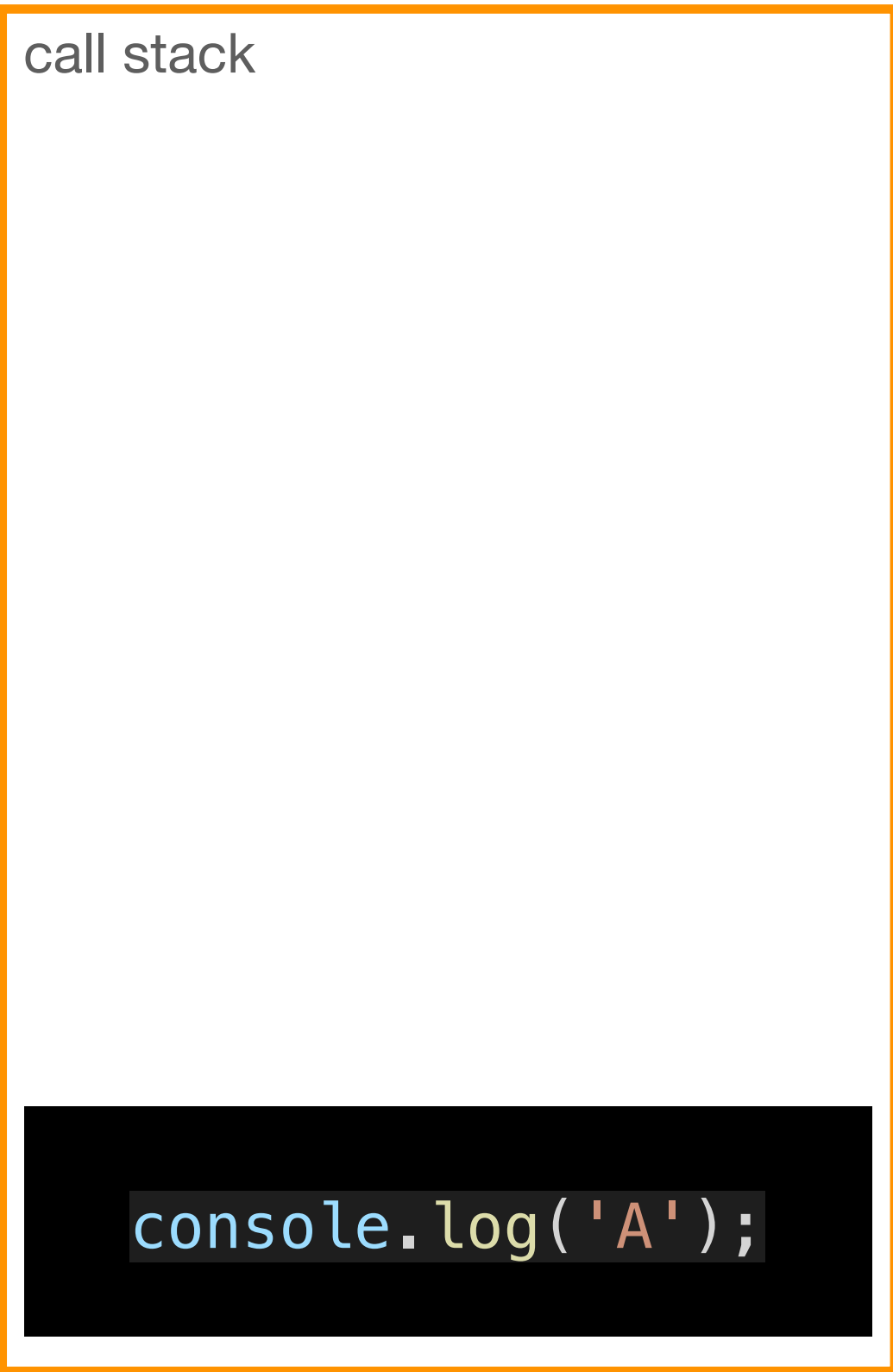


```
console.log('A');
```

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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



9번



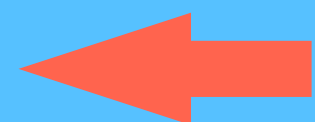
```
console.log('A');
```

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A



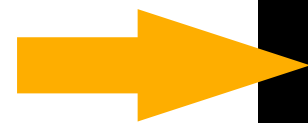
call stack

Web APIs

event loop



callback queue



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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A

call stack

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

Web APIs

event loop



callback queue

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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A

call stack

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

event loop



callback queue



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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

→ superLongComputation(); // 5초가 걸리는 함수

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

A

call stack

```
superLongComputation();
```

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

callback queue

event loop



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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

→ superLongComputation(); // 5초가 걸리는 함수

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

A

call stack

```
superLongComputation();
```

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

동기적으로 실행되기 때문에 5초 기다림

event loop



callback queue

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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

→ superLongComputation(); // 5초가 걸리는 함수

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

A

call stack

```
superLongComputation();
```

Web APIs

event loop



callback queue

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

애는 1초만 기다리는 애니까 1초 땡하면 콜백 큐로

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

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A

call stack

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

Web APIs

event loop



callback queue

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

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

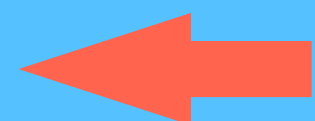
```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A

C



call stack

Web APIs

event loop



callback queue

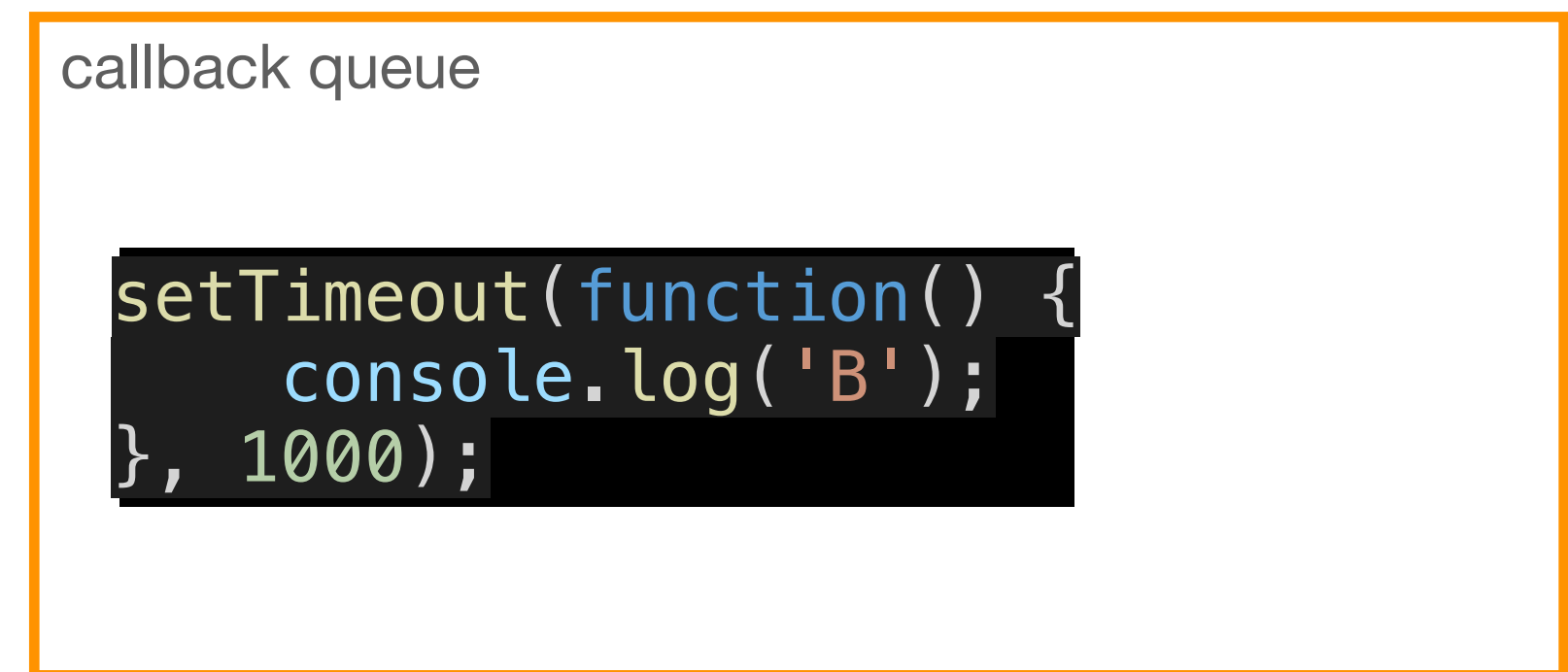
```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

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

setTimeout(function() {
  console.log('B');
}, 1000);

superLongComputation(); // 5초가 걸리는 함수

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



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

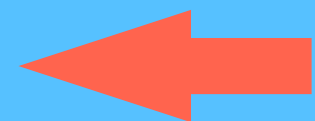
```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A

C



call stack

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

Web APIs

event loop



callback queue

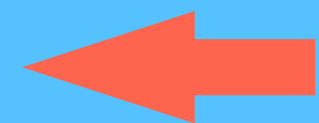
```
console.log('A');
```

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

```
superLongComputation(); // 5초가 걸리는 함수
```

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

A  
C  
B



call stack

Web APIs

event loop



callback queue





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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

call stack

Web APIs

callback queue

event loop



# 10번



```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
  
console.log('D');
```

call stack

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

Web APIs

event loop



callback queue

```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
  
console.log('D');
```

A



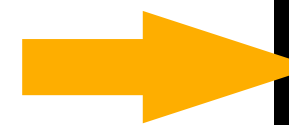
call stack

Web APIs

event loop



callback queue



```
console.log('A');  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
  
console.log('D');
```

A

call stack

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

Web APIs

event loop



callback queue

```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
  
console.log('D');
```

A

call stack

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

event loop



callback queue

```
console.log('A');  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
→ superLongComputation(); // 5초가 걸리는 함수  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
console.log('D');
```

A

call stack

superLongComputation();

Web APIs

```
setTimeout(function() {  
  console.log('B');  
}, 1000);
```

5초를 기다리는 동안 setTimeout의 1초가 지나버림

event loop



callback queue

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A

call stack

superLongComputation();

Web APIs

5초를 기다리는 동안 setTimeout의 1초가 지나버림

event loop



callback queue

```
setTimeout(function() {
  console.log('B');
}, 1000);
```



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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

→ setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A

call stack

```
setTimeout(function() {
  console.log('C');
}, 500);
```

Web APIs

event loop



callback queue

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

**call stack 이 빌 때까지 기다려야 함**

```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
, 500) // 0.5초 대기  
  
console.log('D');
```

A

call stack

Web APIs

```
setTimeout(function() {  
  console.log('C');  
, 500);
```

event loop



callback queue

```
setTimeout(function() {  
  console.log('B');  
, 1000);
```

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A

call stack

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

Web APIs

```
setTimeout(function() {
  console.log('C');
}, 500);
```

event loop



callback queue

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A  
D ←

call stack

콜 스택이 비었다!

Web APIs

```
setTimeout(function() {
  console.log('C');
}, 500);
```

event loop



callback queue

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
, 500) // 0.5초 대기  
  
console.log('D');
```

A  
D

call stack

```
setTimeout(function() {  
  console.log('B');  
, 1000);
```

Web APIs

```
setTimeout(function() {  
  console.log('C');  
, 500);
```

event loop



callback queue

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A  
D

call stack

```
setTimeout(function() {
  console.log('B');
}, 1000);
```

Web APIs

event loop



callback queue

```
setTimeout(function() {
  console.log('C');
}, 500);
```

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A  
D  
B ←

call stack

Web APIs

event loop



callback queue

```
setTimeout(function() {
  console.log('C');
}, 500);
```

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

setTimeout(function () {
  console.log('B');
}, 1000); // 1초 대기

superLongComputation(); // 5초가 걸리는 함수

setTimeout(function() {
  console.log('C');
}, 500) // 0.5초 대기

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

A  
D  
B

call stack

```
setTimeout(function() {
  console.log('C');
}, 500);
```

Web APIs

event loop



callback queue



```
console.log('A');  
  
setTimeout(function () {  
  console.log('B');  
}, 1000); // 1초 대기  
  
superLongComputation(); // 5초가 걸리는 함수  
  
setTimeout(function() {  
  console.log('C');  
}, 500) // 0.5초 대기  
  
console.log('D');
```

A  
D  
B  
C ←

call stack

Web APIs

event loop



callback queue