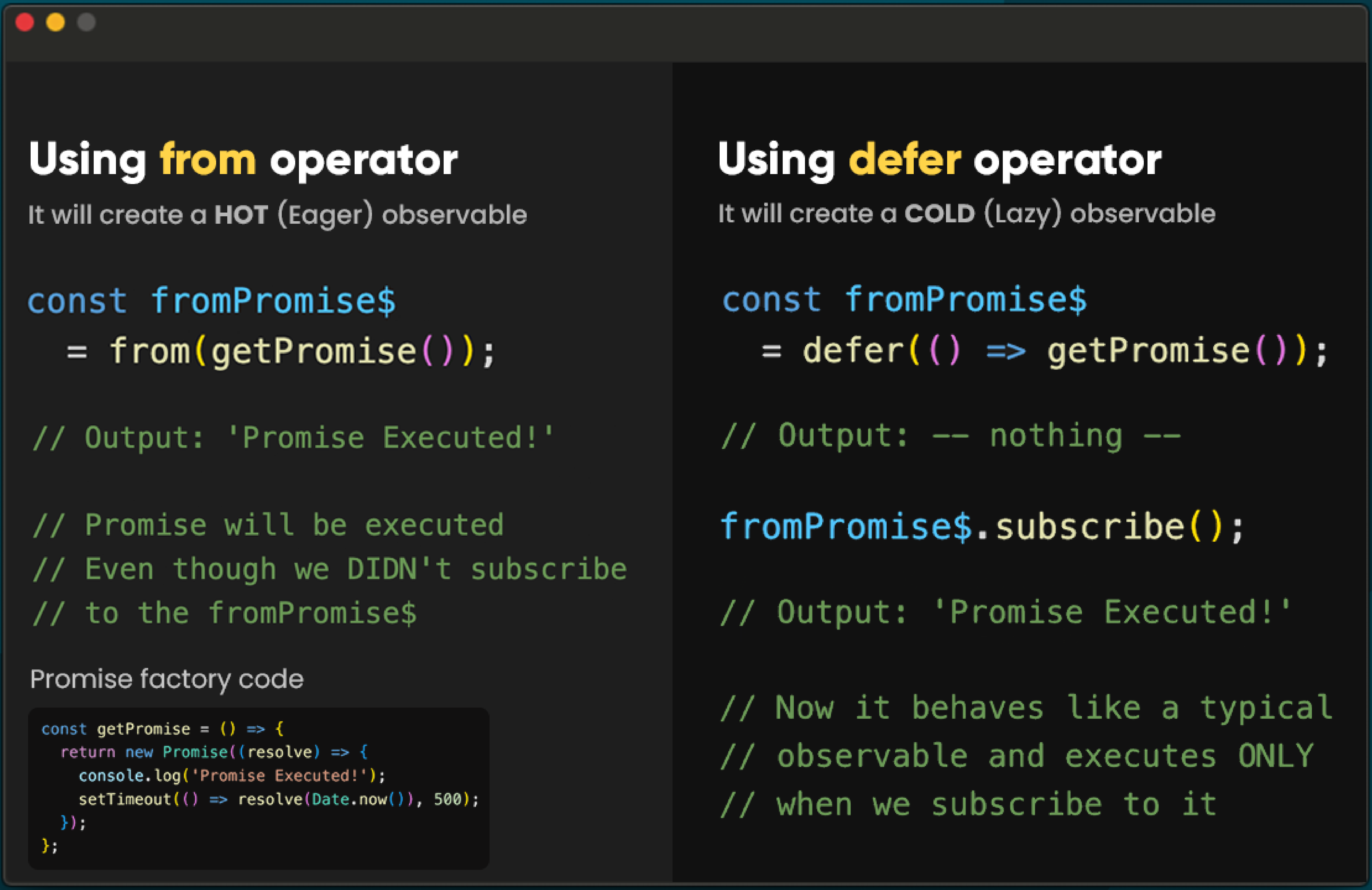
2 Ways to convert Promise into Observable 👇👇



Using from operator

It will create a **HOT** (Eager) observable

```
const fromPromise$
  = from(getPromise());

// Output: 'Promise Executed!'

// Promise will be executed
// Even though we DIDN't subscribe
// to the fromPromise$
```

Promise factory code

```
const getPromise = () => {
  return new Promise((resolve) => {
    console.log('Promise Executed!');
    setTimeout(() => resolve(Date.now()), 500);
  });
};
```

Using defer operator

It will create a **COLD** (Lazy) observable

```
const fromPromise$
  = defer(() => getPromise());

// Output: -- nothing --

fromPromise$.subscribe();

// Output: 'Promise Executed!'

// Now it behaves like a typical
// observable and executes ONLY
// when we subscribe to it
```

DecodedFrontend

decoded frontend

Advanced Angular Tutorials

Like().Subscribe().Share()

DecodedFrontend

decoded
frontend