

# Reactive Context

- Any location in you code where Angular **follows** the invocation of signals
  - Inside the body of computed

```
readonly derived = computed(() => firstSignal() * 2);
```

- Inside the body of effect

```
effect(() => {  
  console.log('The first signal value is:', this.firstSignal());  
  console.log('The second signal value is:', this.secondSignal());  
});
```

# When in a Reactive Context...

- It's ok to

- Read from signals



```
const x = mySignal()
```

- But don't

- Modify a signal



```
computed(() => {  
  x.update(v => v + 1);  
  return x() * 10;  
})
```

- Create a new signal



```
computed(() => {  
  const total = x() + y();  
  return {sum: signal(total)}  
})
```

- Create an effect



```
effect(() => {  
  if (sum() > 10) {  
    effect(() => console.log(y()))  
  }  
})
```