

# Reactivity - Signals

createSignal()

# Signals?

What are they?



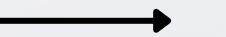


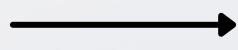
# What is Reactive Programming

To declaratively express the  
relationship between values that  
change over time

$$a = b + c$$

where the value of **a** updates  
whenever the value of **b** or **c** changes





# fine grained reactivity

simple event emitter that hold a value and represent that value over time

Declarative (desc behavior rather than implementation)

Composable

Simple Model consists of only 3 concepts:

1. Signals
2. Derivations -- computed values (update - notify - trigger)
3. Effects (side effects)



# signals

# let's dive deeper

Signals are the most primary part of a reactive system.

They consist of a getter, setter, and a value. Although often referred to as Signals in academic papers, they also have been called Observables, Atoms, Subjects, or Refs.

# Reactivity in frameworks



KNOCKOUT JS 2010

Mobx + React 2015 ([@observable](#))

Svelte (`let count, $: double = count * 2`) 2019

Vue JS Composition 2020 ([ref](#), [reactive](#) apis)

all of these frameworks use those 3 concepts  
as the base for everything.



# React?

Is React Reactive????

it does have [useState](#), [useEffect](#), [useMemo](#)

**will let u think about it**

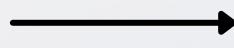


**now let's code them by  
ourselves from scratch**



## Useful links

1. <https://dev.to/ryansolid/a-hands-on-introduction-to-fine-grained-reactivity-3ndf>
2. <https://dev.to/ryansolid/building-a-reactive-library-from-scratch-1i0p>
3. <https://www.youtube.com/watch?v=80L3Ts1t7v8>
4. <https://www.youtube.com/watch?v=O6xtMrDEhcE>
5. <https://www.youtube.com/watch?v=jfZdl3QckMM>
6. <https://frontendmasters.com/courses/reactivity-solidjs/>
7. <https://github.com/tc39/proposal-signals>
8. <https://refactoring.guru/design-patterns/observer>



# Thank You