value which instantly occurs to have a

synchronous data stream or then it's not

a stream I guess or we have multiple

values whatever the case may be we have

our observer with the three methods I

mentioned where we can handle any values

and can handle one or multiple values at

the end we might have an endpoint when

you observable it's done and as I

mentioned at before that might never

occur some observables are never done if

we do complete the observable though so

if we have some data source which

eventually finishes then we can call end

and we will execute complete if the

observer provides it on the observer

object we also might have a stream which

does not only emit values but which also

throws an error at some point of time

think about a stream or an observable I

should say wrapping an HTTP request we

know that we will eventually get back a

response but that response could be an

error either a timeout or maybe a

server-side error in this case the

observable would throw the error and we

could handle it in the error function of

the observer now seeing that on slides

is nice but how do we actually use these

concepts where can we see this concept

in code well let's take a look I'm on

jsfiddle here and I simply chose JSON

because I kind of liked the way it looks

and how we can work with it and what I'm

doing is I'm importing the rxjs package

from their CDN which you can find on the

official documentation on install it at

the very bottom and then I added a

button which I can click

I listen to any leg clicks on the button

with my observable here so I get an

access to the button here and then I

simply wrap the button click and I

create a new observable with the from

event helper method and as a side note

if you visit the official documentation

at reactive X dot IO and then you click

on observable here you see there are a

lot of methods you can use to create a

new observable and a lot of these

methods are actually nice methods to

conveniently create observables which do

something specific like here wrapping an

event average would emit a new value

every X seconds later in this video

we'll also see how we can create an

observable from scratch but back to this

one first

I'm wrapping or I can create an

observable wrapping this click event and

therefore let's open the console

whenever I click the button here we see

that the value the the x position of my

cursor is emitted because that's what

I'm getting here so that is how this

observable works and what happens here

behind the scenes is that this

observable has an infinite stream of