# Bonus:Meteor_and_React_Developer_Tools

- We can use the keyword debugger; in our program where we want it to stop

```
import React from 'react';
import {Players} from './../api/players';



export default class AddPlayer extends React.Component{
    handleSubmit(e){
        let playerName = e.target.playerName.value;
        e.preventDefault();

        debugger;

        if(playerName){
          e.target.playerName.value = '';
          Players.insert({
            name: playerName,
            score: 0
          });
        }
    }
```

- Once the program stops you can use the console to check the variables at any specific time
- we are going to be getting  two debugging tools by react and meteor.  They are available at chrome.google.com/webstore and type meteor dev tools and add the one by The Bakery and the other one is react developer tools by Facebook
- Restart your browser to have to those two extensions available
- by selecting React on your console, you can inspect all of your components and can change things on the fly
- you can also check meteor on your dev tools on chrome, and you can view miniMongo (for some reason I wasn't able to toggle between React And Meteor on the Dev tools so I have to swith to another option on toolbar like 'Memory' and then switch to the one you want to view and that works)
- another feature on Meteor is DDP (Distributed Data Protocol) it lets us view the whats happening between the client and the server, the communication that's getting send over those web sockets - DDP it's what syncs up our local MiniMongo with the Meteor server side with the Mongo server side database, and then it syncs that up with all other connected clients that need to know about updates to data.  In our app you can clearly view what is going on when removing a player, the requests and messages