Day 4:

**Call(), apply() and bind()**

**call()** : invoke a function immediately with a given this(current object) context and arguments passing individually.

**apply()** : here arguments are passed as array.

**bind():**

Doesn’t invoke immediately and it return a new function with this context permanently bound.

functioName.bind(thisArg,arg1,arg2,arg3…..);

**class MyComponent extends React.Component{**

**constructor(props) {**

**super(props);**

**this.state={id:100}**

**}**

**this.hello = this.hello.bind(this);**

**}**

AJAX : Asynchronous JavaScript and XML

XMLHttpRequest

ActiveXObject

Synchronous : the code execute sequentially.

Console.log(“statement1”)

Console.log(“statement2”)

Console.log(“statement3”)

Fun1()

Fun2()

Fun3();

Asynchronous : the code execute independently

Console.log(“statement1”)

Console.log(“statement2 asyn”)

Console.log(“statement3 asyn”)

Fun1()

Fun2()

Fun3();

Req1

Req2

Req3

Client Server

setTimeout(callback,time)

setInterval(callback,time)

clearInteval(referenceToStopInterval)

promise: promise is a pre defined object provided by JavaScript which help to handle asynchronous event of data of any type.

Promise can resolved or rejected or fullfill or pending etc.