Hello everyone, My name is Yan and theme of my lecture is React, so let’s start.  
  
What is React?  
React is a component-based library which is used to develop interactive User Interfaces. It is currently one of the most popular JavaScript front-end library.  
  
Why React so fast? It is fast Because of two things: first of all is Virtual DOM and the second is Effective Updating Virtual DOM  
So what is virtual DOM? Virtual DOM is the in-memory representation of Real DOM. It is lightweight JavaScript object which

is copy of Real DOM. You must notice that React maintains two virtual DOMs, first is updated virtual DOM and the second is previous virtual DOM.  
Effective Updating Virtual DOM include steps like  
Efficient diff algorithm  
Batched update operations  
Efficient update of sub tree only  
Uses observable’s instead of dirty checking to detect change  
  
What is observables? It means ReactJS watch components and notice when component change it’s props or state. When it changes React rerender this component.  
Batch Update operation, what does this mean?  
This mean, that react try to minimize operations and communications with real DOM, so React at first updates all states of components and only swen all states is updated start to render it in real dom.  
  
lets continue. Virsual dom elements versus dom elements  
on this image we see representation of react element, it has small amount of values, and now lets view on the dom element.  
dom element is huge, and this pictures represent not all values of dom object. react components lightweight therefore operations with these parts are fast.  
  
States to find the difference in both the Virtual DOM’s  
on the picture bellows picture left is virsual dom representation and picture right is real dom element representation.  
As you can see virsual dom has small amount of properties it means that it lightweight and operation with it is fast instead of big object that represents real dom element. This element has more properties and operation with it is slower.  
okay, lets continue.

What is the steps to find all differencies in virsual doms?  
If the state of a component has changed, then React re-renders all the child components even if child components are not modified.  
the second step is React traverse the tree using Breadth First Search. What does this mean?  
Look at the below tree. Two elements B and H have changed. So when React goes through the tree it notice that element B is changed, so it will change all child elements of B, and it mean that element H will rerender also.  
the third step is reconciliation. What does this mean? Here we see two virsual doms, this is initial virsual dom and target virsual dom. here react start to matches this lists components and notice, that only element New York is new in this list. React match this list component by the keys value and as you can see key with zh and ld is already existed in the initial virtual dom. And that is mean that react will update only one component, not all of those.  
  
What is JSX?  
JSX is syntax that represents react elements. As we can notice we can put some variables on the attributes and also on the tags. And second what do you must know about this JSX that, we wrap all our elements into one div or another element. This is need to correct react rendering. And third is React has some differencies with attributes, in HTML we have attribute class and instead of JSX we have attribute className. The second exception is in HTML we have attribute for for label, in JSX this attribute calls htmlFor.  
  
Here we see React component, stateless component and statefull component.  
All this components are building blocks of React application. We can see basic JSX syntax and ReactDOM.render() function. This function is start to parse react virsual DOM to real Dom on the browser. What is the difference between stateless and statefull component? The difference that statefull component has its own property this property is the state. And in this property we can set some values that represet out class andour component. Second is we have props it is outside value that is need to work inside the component.  
  
Lets move on and see what is component life circle.  
here are three phases: first is mounting, seconds is update , third is unmounting  
in vaunting stage we first of all goes to constructor and initialize it then we render our element and then react updates DOM and after it componentDidMount function is immediately invoking.  
Okay, update phase. When it start: it starts on three situations: first of all when props of component os change, second is when setState is execute and third is when parent node is changed, then component starts updating phase whitch os start from render and when element allready rerender react calls component didUpdate, and unmounting this is phase when component is destroyed on real DOM  
  
to better understand lets view on example of class Clock and how it represents on the browser.   
what is going on here?

When <Clock /> is passed to ReactDOM.render(), React calls the constructor of the Clock component. Clock needs to display the current time, so it initializes this.state with an object including the current time.   
React then calls render() method. React then updates the DOM and browser show current time.   
When the Clock output is inserted in the DOM, React calls the componentDidMount() lifecycle method. Inside it, the Clock component asks the browser to set up a timer to call the component’s tick() method once a second.   
Every second the browser calls the tick() method. Inside it, the Clock component update setState(). When the setState() call, React knows the state has changed, and calls the render() method again to learn what should be on the screen. This time, this.state.date in the render() method will be different, and so the render output will include the updated time. React updates the DOM accordingly.   
If the Clock component is ever removed from the DOM, React calls the componentWillUnmount() lifecycle method so the timer is stopped.   
Notice important thing from the second image: browser update only string with time, other html stay unchanged.

we must change state only use setState function  
Because this.props and this.state may be updated asynchronously, you should not rely on their values for calculating the next state.