JSX

JSX

> *JavaScript eXtension JS*
>
> *UI React .*
>
> *JS .*
>
> *React , UI*

JSX

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;

ReactDOM.render(
  element,
  document.getElementById('root')
);
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}

const user = {
  firstName: 'Harper',
  lastName: 'Perez'
};

const element = (
  <h1>
    Hello, { formatName(user) }!
  </h1>
);

ReactDOM.render(
  element,
  document.getElementById('root')
);
```

name {} .

JS .

) `{ 2+2 }`, `{ user.firstName }`, `{ formatName(user) }`

JSX , , .

```
function getGreeting(user) {
  if (user) {
    return <h1>Hello, { formatName(user) }!</h1>;
  }
  return <h1>Hello, Stranger.</h1>;
}
```

Babel

Babel JSX React.createElement() .

```
const element = (
  <h1 className="greeting">
    Hello, world!
  </h1>
);
const element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, world!'
);
```

.

.

```
const element = {
  type: 'h1',
  props: {
    className: 'greeting',
    children: 'Hello, world!'
  }
};
```

- https://babeljs.io/repl/#?browsers=defaults%2C%20not%20ie%2011%2C%20not%20ie_mob%2011&build=&builtIns=false&corejs=3.6&spec=false&loose=false&code_lz=GYVwdgxgLglg9mABACwKYBt1wBQEpEDeAUIogE6pQhlIA8AJjAG4B8AEhlogO5xnr0AhLQD0jVgG4iAXyJA&debug=false&forceAllTransforms=false&shippedProposals=false&circleciRepo=&evaluate=false&fileSize=false&timeTravel=false&sourceType=module&lineWrap=true&presets=react&prettier=false&targets=&version=7.15.6&externalPlugins=&assumptions=%7B%7D

JSX

attribute "" .

```
const element = <div className="my-class"></div>
```

attribute JS .

```
const element = <img src={ user.avatarUrl }></img>
```

=> attribute .

, .

=> class `class` `className` (class JS)

=> HTML attribute camelCase .

- (`onclick` => `onClick`)
- (style { `margin-top` x => `marginTop` o })

Tag, Fragment

HTML , `input` `br` .

`<div />` .

return .

(Vue component template)

`React.Fragment` .

```
function App () {
  return (
    <React.Fragment>
      <Hello />
      <div></div>
    </React.Fragment>
  )
}

//     Fragment  .
function App () {
  return (
    <>
      <Hello />
      <div></div>
    </>
  )
}
```

```
function Glossary (props) {
  return (
    <dl>
      { props.items.map(item => {
        <React.Fragment key={item.id}>
          <dt>{ item.term }</dt>
          <dd>{ item.descriptioin }</dd>
        </React.Fragment>
      }) }
    </dl>
  )
}
```

JSX JS if .

JSX if {} .

```
import React from 'react'

function App () {
  const name = 'react'
  return (
    <div>
      { name === 'react' ? (<h1>This is React.</h1>) : (<h2>This is not
React</h2>) }
    </div>
  )
}
```

v-if && .

```
function App () {
  const name = 'react'
  return (
    <div>
      { name === 'react' && <h1>It's correct!</h1> }
    </div>
  )
}
```

<div> <span> , .

```
import React from 'react';

//  !     .
function hello(props) {
  //  !  <div>   HTML   .
  return <div>Hello {props.toWhat}</div>;
}

function HelloWorld() {
  //  ! React <hello />    HTML   .
  return <hello toWhat="World" />;
}
```

```
import React from 'react';
import { PhotoStory, VideoStory } from './stories';

const components = {
  photo: PhotoStory,
  video: VideoStory
};


// 1
function Story(props) {
  //  ! JSX     .
  return <components[props.storyType] story={props.story} />;
}

// 2
function Story(props) {
  //  !    JSX    .
  const SpecificStory = components[props.storyType];
  return <SpecificStory story={props.story} />;
}
```

```
const element = (
  <h1 className="greeting">
    Hello, world!
    <Child />
  </h1>
);
```

```
const element = /*#__PURE__*/React.createElement( "h1", { className:
"greeting" }, "Hello, world!",
/*#__PURE__*/React.createElement(Child, null));
```

boolean, null, undefined

`false`, `null`, `undefined`, `true` .

.

```
<div />

<div></div>

<div>{false}</div>

<div>{null}</div>

<div>{undefined}</div>

<div>{true}</div>
```

`0` .

```
// props.messages.length = 0
<div>
  {props.messages.length &&
    <MessageList messages={props.messages} />
  }
</div>

///
0


//  &&   Boolean  .
<div>
  {props.messages.length > 0 &&
    <MessageList messages={props.messages} />
  }
</div>
```

JSX

`JSX` `props.children` .

```
// App.js
return (
 <MyComp>
  <div>
    123412342
    <div>
      555
    </div>
  </div>
  <span>nnnnn</span>
  <span></span>
  {5+2}
  <MyChild></MyChild>
</MyComp>
)

/////////
// MyComp.js
function MyComp(props) {
  console.log(props)
  return (
      <div>Hello
          {props.children}
      </div>
  );
}

export default MyComp;


// MyChild.js
function MyChild(props) {
    return (
        <div>Child!!</div>
    );
  }

  export default MyChild;
//

Hello
123412342
555
nnnnn7
Child!!
```

.

```
//   numTimes    .
function Repeat(props) {
  let items = [];
  for (let i = 0; i < props.numTimes; i++) {
    items.push(props.children(i));
  }
  return <div>{items}</div>;
}

function ListOfTenThings() {
  return (
    <Repeat numTimes={10}>
      {(index) => <div key={index}>This is item {index} in the list<
/div>}
    </Repeat>
  );
}
```

JSX

{/* */} .