**Context API**

Context API

Сэдвээсээ хазайхаас өмнө эхэлцгээе!

**Илэрхийллийг JSX дотор шигтгэх(embed)**

Доорх жишээнд бид name нэртэй хувьсагч зарлан түүнийгээ JSX дотор угалзан хаалтууд хүрээлүүлэн ашигласан байна:

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

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

Та ямар ч зөв [Жаваскрипт илэрхийлэл](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators#Expressions) угалзан хаалтаар хүрээлүүлэн JSX дотор ашиглаж болно. Жишээлбэл 2 + 2, user.firstName, эсвэл formatName(user) нь бүр зөв Жаваскрипт илэрхийллүүд юм.

Доорх жишээнд бид formatName(user) Жаваскрипт функцийн утгийг <h1> элемент рүү шигтгэж өгсөн байна.

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')

);

[**Try it on CodePen**](https://mn.reactjs.org/redirect-to-codepen/introducing-jsx)

Бид кодыг илүү унших байдлыг нь бодон JSX-ээ олон мөр болгон хуваасан. Энэ зайлшгүй хийх ёстой зүйл биш ч хийж байгаа тохиолдолд [автомат цэг таслал нэмэгдэх](https://stackoverflow.com/q/2846283)-ээс сэргийлж дугуй хаалтаар хүрээлүүлэхийг зөвлөж байна.

**JSX бол бас илэрхийлэл юм**

Хөрвүүлэлт хийгдсний дараа JSX илэрхийллүүд нь энгийн Жаваскрипт функцийн дуудалтууд болон объектууд болно хөрвөдөг.

Энэ нь if нөхцөл болон for давталт, хувьсагчуудийн утга олголтууд, функцийн аргументууд хүлээн авах, функцээс утга буцаах үйлдлийг JSX дотор хийж болно гэсэн үг:

function getGreeting(user) {

if (user) {

return <h1>Hello, {formatName(user)}!</h1>; }

return <h1>Hello, Stranger.</h1>;}

**JSX-р аттрибут тодорхойлох нь**

Та аттрибут дээр квот ашиглан тэмдэгт төрлөөр өгөх байх:

const element = <div tabIndex="0"></div>;

Мөн та угалзан хаалтаар Жаваскрипт илэрхийлэл аттрибутад шигтгэж болно:

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

Жаваскрипт илэрхийлэл угалзан хаалт ашиглан аттрибутад шигтгэж өгж байгаа бол квот ашиглахгүй. Та квот (тэмдэгт утгийн хувьд) эсвэл угалзан хаалтын (илэрхийлэлд) аль нэгийг ашиглаж болох ч зэрэг нэг аттрибутад хэрэглэж болохгүй.

**Анхааруулга:**

JSX нь HTML илүүтэй Жаваскрипттэй төстэй учир React DOM camelCase шинж чанарын нэрлэх стандарт HTML аттрибут нэрүүдийн оронд хэрэглэдэг.

Жишээлбэл class нь JSX дээр [className](https://developer.mozilla.org/en-US/docs/Web/API/Element/className), tabindex нь [tabIndex](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/tabIndex) болдог.

**JSX-д дэд(children) элемент тодорхойлох нь**

Хэрэв таг хоосон бол яг XML шиг /> хаах болно:

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

JSX тагууд нь дэд элемент агуулж болдог:

const element = (

<div>

<h1>Hello!</h1>

<h2>Good to see you here.</h2>

</div>

);

**JSX Injection халдлагаас сэргийлдэг**

JSX дотор хэрэглэгчийн оролтыг шигтгэхэд аюулгүй байдаг:

const title = response.potentiallyMaliciousInput;

// This is safe:

const element = <h1>{title}</h1>;

Анхнаасаа React DOM JSX дотор шигтгэгдсэн утгийг дүрслэхийн өмнө [escape хийдэг](https://stackoverflow.com/questions/7381974/which-characters-need-to-be-escaped-on-html). Ингэснээр та таны програмд тусгагдаагүй зүйлийг тарилга байдлаар халдах боломжийг хаадаг. Бүх зүйл дэлгэцэнд дүрслэгдэхээсээ өмнө тэмдэгт болон хувирдаг. Энэ нь [XSS (cross-site-scripting)](https://en.wikipedia.org/wiki/Cross-site_scripting) халдлагуудаас сэргийлдэг.

**JSX Объектуудыг төлөөлөх нь**

Babel JSX-ыг React.createElement() дуудалтууд болгон хөрвүүлдэг.

Доорх хоёр жишээнүүд ижилхэн:

const element = (

<h1 className="greeting">

Hello, world!

</h1>

);

const element = React.createElement(

'h1',

{className: 'greeting'},

'Hello, world!'

);

React.createElement() хэдэн шалгалтууд гүйцэтгэн таньд алдаагүй код бичихэд тусалдаг ч дараах шиг объект үүсгэдэг:

// Note: this structure is simplified

const element = {

type: 'h1',

props: {

className: 'greeting',

children: 'Hello, world!'

}

};

Эдгээр объектууд нь “React элементүүд” гэж дуудагддаг. Та үүнийг дэлгэц дээр харахийн хүссэн зүйлийн чинь тайлбар гэж бодож болно. React эдгээр объектуудыг уншаанд DOM шинэчилж байдаг.

Бид [дараагийн бүлэгт](https://mn.reactjs.org/docs/rendering-elements.html) React элементүүдийг DOM руу хэрхэн дүрслэх талаар судлах болно.

**Зөвлөмж:**

Бид [“Babel” хэлний тодорхойлолт](https://babeljs.io/docs/editors)-ийг код засварлагчийн сонголт дээрээ ашиглахийг зөвлөдөг ES6 болон JSX код нь хоёулаа зөв тодордог(highlight). Энэ вебсайт [Oceanic Next](https://labs.voronianski.com/oceanic-next-color-scheme/) өнгөний схем хэрэглэдэг зохицолдолгоо сайтай.