

Bài 2

JS = Javascript

JS: chạy trên trình duyệt + Server (Nodejs) + Mobile (React Native)

=====

ReactJS: DOM, alert(), Document

React Native: Không có DOM, Alert của RN, Không có Document

=> JS dùng để:

- .xử lý dữ liệu
- .Điều khiển UI
- .Call API

=====

Biến:

var x=5; //không dùng

Dùng biến thế nào trong RN

let – biến có thể thay đổi được, hay dùng trong vòng lặp
ví dụ:

```
let count=0;
```

```
count=1;
```

const – biến không đổi

Không đổi không có nghĩa là không thay đổi giá trị bên trong

=====

Kiểu dữ liệu:

```
const number=5;//number
```

```
const text="xin chao" //string
const isLogin = true; //boolean
const a= null; //null
let undef; //undefine
=====
```

Kiểu Object => kiểu tham chiếu:

là tập hợp dữ liệu có cấu trúc

```
const user = {
  id:1,
  name: "an",
  email: "an@gmail.com"
};
```

Cách truy cập:

user.name

user["email"]

=====

Ví dụ kiểu tham chiếu

```
const a = {x:1};
```

```
const b=a;
```

```
b.x = 2;
```

```
console.log(a.x);//2
```

=====

Array: Danh sách

```
const products = [
  {id:1, name:"Áo"},
  {id:2, name: "quần"},
];
```

```
];  
duyet mảng (vòng lặp)  
products.map(item => item.name);  
=====
```

Hàm mũi tên:

```
const sum = (a,b) => a+b;
```

Hàm mũi tên không tạo this riêng

```
=====
```

Destructuring: tách thuộc tính khỏi đối tượng

Có object:

```
const user = {name:"An",age:20};
```

```
const {name,age}=user; //khai báo destructuring
```

ví dụ:

```
function hello ({name}){  
    console.log(name);  
}
```

```
=====
```

Toán tử ...: copy 1 thuộc tính mới vào đối tượng

ví dụ1:

```
const user = {name:"an"};
```

```
const newUser = {...user, age:20};
```

ví dụ2:

```
const arr = [1,2];
```

```
const newArr = [...arr,3]; //copy thêm 1 thành phần vào  
arr
```

```
=====
```

Thay đổi trạng thái:

state.count++; //sai

setState(**{...state, count: state.count+1}**);

=> React không theo dõi sự thay đổi trạng thái, mà chỉ render khi có object mới

=====

if....else

isLoggedIn ? <Home/> : <Login />

tương đương

if(isLoggedIn){

 showHome();

}

=====

async/await

async function fetchData(){

 const res = await fetch(url);

 const data = await res.json();

}

Mọi API trong react native đều dùng async/await

=====

Object => Props

Array.map => Render list

Async/await => API

Destructuring => component props

=====

