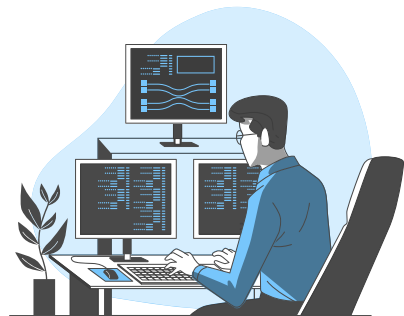


# Advanced JS



## Khóa học Backend

### Bài 13: Javascript nâng cao (Tiết 3)



# Nội dung

01  
...

Storage API

03  
...

Closure (bao đóng)

02  
...

So sánh localStorage,  
sessionStorage, cookies





# 01

## Storage API



## 1.1. localStorage

- **localStorage** là kho lưu trữ ở phía người dùng.
- Lưu trữ dữ liệu vô thời hạn, dữ liệu sẽ được lưu trữ cho tới khi người dùng xóa lịch sử duyệt web.
- localStorage có **5 phương thức**:
  - **setItem()** : Thêm dữ liệu vào localStorage.
  - **getItem()** : Lấy dữ liệu từ localStorage.
  - **removeItem()** : Xóa dữ liệu ra khỏi localStorage.
  - **clear()** : Xóa toàn bộ dữ liệu ra khỏi localStorage.
  - **key()** : Lấy tên key của dữ liệu đang lưu trữ trong localStorage.

## 1.1. localStorage

- `localStorage.setItem()`
  - Dùng để **thêm dữ liệu** vào `localStorage`.
  - Dữ liệu sẽ được lưu trữ ở **dạng key và value**.
  - Cú pháp:

```
localStorage.setItem(key, value);
```

## 1.1. localStorage

- `localStorage.getItem()`
  - Dùng để **lấy dữ liệu**.
  - Truyền vào tên key muốn lấy.
  - Nếu không tồn tại thì trả về null.
  - Cú pháp:

```
localStorage.getItem(key);
```

## 1.1. localStorage

- `localStorage.removeItem()`
  - Dùng để **xóa dữ liệu**.
  - Truyền vào tên key muốn xóa.
  - Cú pháp:

```
localStorage.removeItem(key);
```

## 1.1. localStorage

- `localStorage.clear()`
  - **Xóa toàn bộ dữ liệu** của `localStorage`.
  - Truyền vào tên key muốn xóa.
  - Cú pháp:

```
localStorage.clear();
```



## 1.1. localStorage

- `localStorage.key()`
  - Lấy ra key của localStorage.
  - Nếu không có trả ra null.
  - Cú pháp:

```
localStorage.key(index);
```

- Trong đó: **index** là vị trí của key đó, có thể là 0, 1, 2, ...

## 1.2. sessionStorage

- **sessionStorage** là **kho lưu trữ theo phiên**.
- Lưu trữ dữ liệu cho một phiên làm việc, có nghĩa **dữ liệu** sẽ bị **mất khi** bạn **tắt browser**.
- sessionStorage có **5 phương thức**:
  - **setItem()** : Thêm dữ liệu vào sessionStorage.
  - **getItem()** : Lấy dữ liệu từ sessionStorage.
  - **removeItem()** : Xóa dữ liệu ra khỏi sessionStorage.
  - **clear()** : Xóa toàn bộ dữ liệu ra khỏi sessionStorage.
  - **key()** : Lấy tên key của dữ liệu đang lưu trữ trong sessionStorage.

## 1.2. sessionStorage

- sessionStorage.**setItem()**
  - Dùng để **thêm dữ liệu** vào sessionStorage.
  - Dữ liệu sẽ được lưu trữ ở **dạng key và value**.
  - Cú pháp:

```
sessionStorage.setItem(key, value);
```

## 1.2. sessionStorage

- `sessionStorage.getItem()`
  - Dùng để **lấy dữ liệu**.
  - Truyền vào tên key muốn lấy.
  - Nếu không tồn tại thì trả về null.
  - Cú pháp:

```
sessionStorage.getItem(key);
```

## 1.2. sessionStorage

- `sessionStorage.removeItem()`
  - Dùng để **xóa dữ liệu**.
  - Truyền vào tên key muốn xóa.
  - Cú pháp:

```
sessionStorage.removeItem(key);
```

## 1.2. sessionStorage

- `sessionStorage.clear()`
  - **Xóa toàn bộ dữ liệu** của `sessionStorage`.
  - Truyền vào tên key muốn xóa.
  - Cú pháp:

```
sessionStorage.clear();
```

## 1.2. sessionStorage

- `sessionStorage.key()`
  - Lấy ra key của sessionStorage.
  - Nếu không có trả ra null.
  - Cú pháp:

```
sessionStorage.key(index);
```

- Trong đó: **index** là vị trí của key đó, có thể là 0, 1, 2, ...

# 02

**So sánh localStorage,  
sessionStorage, cookies**



## ... 02. So sánh localStorage, sessionStorage, cookies

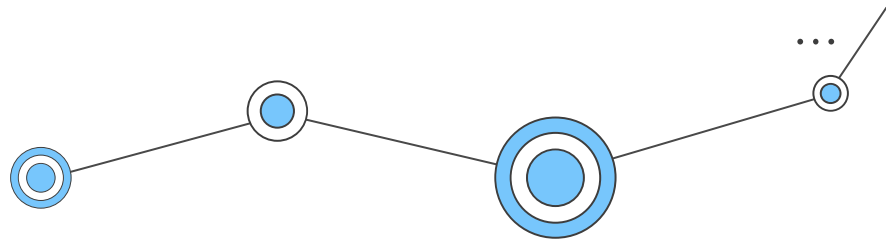
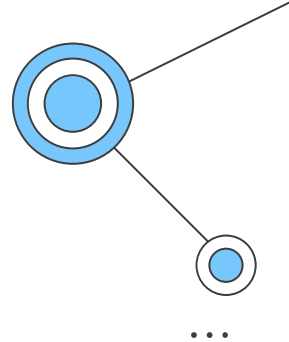
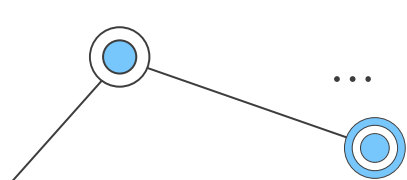
	cookies	localStorage	sessionStorage
<b>Hết hạn (Expires)</b>	Sau khi tắt trình duyệt hoặc đặt thủ công	Vô thời hạn	Sau khi tắt trình duyệt
Có thể truy cập/Lưu trữ	Server side hoặc Client side	Client side	Client side
Kích thước lưu trữ	4 KB	5 MB	10 MB

# 03

## Closure (bao đóng)

### 03. Closure (bao đóng)

- **Closure** là **một hàm ở bên trong hàm khác**.
- Nó có thể sử dụng các biến toàn cục, biến cục bộ của hàm cha và biến cục bộ của chính nó.
- Nếu có các biến trùng tên thì closure sẽ ưu tiên theo thứ tự: biến cục bộ của chính nó, biến cục bộ của hàm cha, biến toàn cục.
- Nếu muốn return nhiều hàm closure thì bạn phải sử dụng một object, trong đó mỗi phần tử sẽ là một hàm closure.
- Closure thay đổi được giá trị biến toàn cục lẫn cục bộ.



# Bài tập

- **Link bài tập:** <https://dacavn.notion.site/B-i-t-p-b-i-13-Javascript-n-ng-cao-Ti-t-3-1da44b750a0e4d6b97e7d0584719ec38?pvs=4>

