* JSON.stringify : chuyển đổi object thành JSON.
* JSON.parse : chuyển đổi JSON trở lại object.

\*\*\*\*\* JSON.stringify \*\*\*\*\*

* let json = JSON.stringify(object).
* let json = JSON.stringify(value,[, replacer, space])
* Trong đó : value là value cần chuyển dổi

Replacer : là array chứa các properties hoặc 1 hàm mapping có dạng function(key, value).

Space : khoảng trắng dùng cho format.

**Lưu ý: phải có dấu []**

* Json lúc này được gọi là JSON-encoded hoặc stringified…
* JSON-encoded sẽ có 1 số khác biệt so với object như sau:

+ Strings sử dụng dấu ngoặc kép(double quotes) : không dùng ngoặc đơn hay backticks trong JSON.

+ Properties name của object cũng sẽ được đặt trong dấu ngoặc kép,đó là bắt buộc. age: 30 => “age”: 30.

* JSON.stringfy cũng có thể dùng cho primitives
* JSON.stringify sẽ không áp dụng với các dạng sau :

+ function properties (methods)

+ Symbolic properties

+ Properties có giá trị là undefined.

* **Các object lồng vào nhau vẫn có thể được chuyển đổi trừ các object có circular references.**
* **Để khắc phục, ta có thể dùng replacer như sau :**

et room = {

number: 23

};

let meetup = {

title: "Conference",

participants: [{name: "John"}, {name: "Alice"}],

place: room // meetup references room

};

room.occupiedBy = meetup; // room references meetup

alert( JSON.stringify(meetup, ['title', 'participants']) );

// {"title":"Conference","participants":[{},{}]}

* Ở ví dụ trên cho thấy sự too strict khi khai báo 2 properies là title, participants thì kết quả ở participants là 2 dấu ngoặc do name không được khai báo trong replacer.
* **Ngoài việc truyền properties, ta có thể dùng hàm có dạng key, value như sau :**

let room = {

number: 23

};

let meetup = {

title: "Conference",

participants: [{name: "John"}, {name: "Alice"}],

place: room // meetup references room

};

room.occupiedBy = meetup; // room references meetup

alert( JSON.stringify(meetup, function replacer(key, value) {

alert(`${key}: ${value}`);

return (key == 'occupiedBy') ? undefined : value;

}));

/\* key:value pairs that come to replacer:

: [object Object]

title: Conference

participants: [object Object],[object Object]

0: [object Object]

name: John

1: [object Object]

name: Alice

place: [object Object]

number: 23

\*/

* Replacer function sẽ nhận cặp key/value bao gồm cả nested object và array items.

\*\*\*\*\* JSON.parse() \*\*\*\*\*

* let value = JSON.parse(str, [reviver]);
* Trong đó : reviver là function với cặp key/value sẽ được gọi cho mỗi cặp key/value. Và thay đổi value.
* Để chuyển đổi thì str cần có dạng hay chuẩn như của JSON xem ở trên.
* Sử dụng reviver để chuyển đổi key/value vd như sau :

let str = '{"title":"Conference","date":"2017-11-30T12:00:00.000Z"}';

let meetup = JSON.parse(str, function(key, value) {

if (key == 'date') return new Date(value);

return value;

});

alert( meetup.date.getDate() ); // now works!

Hoặc nested object

let schedule = `{

"meetups": [

{"title":"Conference","date":"2017-11-30T12:00:00.000Z"},

{"title":"Birthday","date":"2017-04-18T12:00:00.000Z"}

]

}`;

schedule = JSON.parse(schedule, function(key, value) {

if (key == 'date') return new Date(value);

return value;

});

alert( schedule.meetups[1].date.getDate() ); // works!

* Ở hai hàm trên, nếu chỉ sử dụng parse thì sẽ chỉ là string ko phải là object date.