**Thuật ngữ**

Conditional Statement: Câu lệnh điều kiện

Reference

Primitive Data type

Non-primitive Data type

Closure: Khép kín

Asynchoronous: Bất đồng bộ

Synchronous: Đồng bộ

Hoisting

Prototype: Nguyên mẫu

**Phương thức**

**1. String**

let myString = "Hello World"

length

charAt(0)

concat()

indexOf("World")

lastIndexOf("o")

includes("World")

substring(0,5)

slice(0,5)

search("World") // 7: Tìm vị trí của một chuỗi con hoặc biểu thức chính quy. Không tìm thấy trả về -1

replace("World", "Hi")

toUpperCase()

toLowerCase()

trim()

spilit()

startWith()

endWith()

padStart()

padEnd()

**2. Number**

toString()

toFixed(n)

toPrecision(n): Chyển số thành chuỗi với độ dài n

parseInt()

parseFloat()

Number.isNaN()

Number.isFinite()

Math.round()

Math.floor()

Math.ceil()

Math.abs():

Math.random()

Math.min()

Math.max()

**3. Object**

Object.create(proto, [propertiesObject]): Tạo một đối tượng mới với đối tượng nguyên mẫu (proto) đã được chỉ định.

Object.assign(target, ...sources

Object.keys(obj)

Object.values(obj)

Object.entries(obj)

Object.freeze(obj)

Object.seal(obj)

Object.getPrototypeOf(obj): Trả về nguyên mẫu (prototype) của đối tượng.

Object.setPrototypeOf(obj, proto): Đặt một đối tượng khác làm nguyên mẫu cho đối tượng hiện tại.

Object.defineProperty(obj, prop, descriptor): Xác định hoặc sửa đổi một thuộc tính của đối tượng với mô tả chi tiết.

Object.defineProperties(obj, props): Xác định nhiều thuộc tính cùng lúc cho một đối tượng.

Object.is(value1, value2): So sánh hai giá trị có bằng nhau hay không

Object.preventExtensions(obj): Ngăn không cho thêm bất kỳ thuộc tính mới nào vào đối tượng.

**4. Array**

Các hàm không sửa mảng gốc: slice, concat, flat

push()

pop()

unshift()

shift()

indexOf()

lastIndexOf()

join()

toString()

splice(i,del, “x”)

slice(x, y)

concat()

sort()

filter()

map():

reduce()

forEach():

find():

findIndex():

includes()

reverse():

every():

some():

strucrturedClone(): Sao chép mảng sâu