NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

Name : HARSHRAJSINH ZALA
Reg no :22BCE2238

All About Not A Number (NaN)
CODE :

```
const prompt = require('prompt-sync')()
let x = prompt("Enter  a number: ");

if(isNaN(x)){
    console.log("its not a number");
}
else{
    console.log("its a valid number");
}
```

OUTPUT :

```
harsh@Harshs-MacBook-Air JS % cd "/Users/harsh/weblab/JS"
harsh@Harshs-MacBook-Air JS % node "/Users/harsh/weblab/JS/try.js"
Enter  a number: 12
its a valid number
harsh@Harshs-MacBook-Air JS %
```

AGE :
CODE :

```
const prompt = require('prompt-sync')()
let x = prompt("Enter  a number: ");
if(x>120 || x<0){
    if(isNaN){
    console.log("its not a valid age");
}
}
```

OUTPUT :

```
harsh@Harshs-MacBook-Air LAB2 % cd "/Users/harsh/weblab/JS/LAB2"
harsh@Harshs-MacBook-Air LAB2 % node "/Users/harsh/weblab/JS/LAB2/age.js"
Enter  a number: -2
its not a valid age
harsh@Harshs-MacBook-Air LAB2 %
```

NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

USES OF NAN

When converting char to int or else
Illegal math expressions
Infinity /infinity
Arithmetic operations involving a NaN
Relational operators
NaN is never equal to another NaN

CODE :

```
const prompt = require('prompt-sync')()
let x=12;
console.log("x is "+x);

let y=Math.sqrt(-1);
console.log("y is "+y);

let i = Infinity/Infinity;
console.log(i);

let z=7*NaN;
console.log("z is "+z);

if(7>NaN){
    console.log("TRUE");
}
else{
    console.log("Relation Operators with NaN are always FALSE");
}

if(NaN == NaN){
    console.log("nan == nan are always equal")
}
else{
    console.log("nan == nan are always unequal")
}

if(NaN != NaN){
    console.log("they are equal")
}
else{
    console.log("nan == nan are always unequal")
}
```

NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

Output ;

```
harsh@Harshs-MacBook-Air JS % cd "/Users/harsh/weblab/JS"
harsh@Harshs-MacBook-Air JS % node "/Users/harsh/weblab/JS/try.js"
x is 12
y is NaN
NaN
z is NaN
Relation Operators with NaN are always FALSE
nan == nan are always unequal
they are equal
harsh@Harshs-MacBook-Air JS %
```

ARRAY in JAVA SCRIPT :

CODE :

```
const prompt = require('prompt-sync')()
//declaration of arrays using const
//cannot redeclare or reinitialise
const subjects = ["java","python","DAA","OS"];
{
    const subjects= ["java","python","DAA","OS"];
    console.log(subjects[0]);
}
console.log(subjects[1]);

//also declaring and assigning should be done at the same time
// const sub;
// sub= ["java","python","DAA","OS"];           // NOT ALLOWED


//method 2 to declare array
const sub = [];
sub[0] = "java";
sub[1] = "s1";
console.log(sub[1]);

//method 3
const s = new Array("szero","sone","stwo");
console.log(typeof(s));

//method to check if an array
if(Array.isArray(sub)){
    console.log("ITs an Array using instance of Array.isArray ");
}
else{
    console.log("not an array");
}
```

NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

```
//method 2 to check if an array
if(sub instanceof Array){
    console.log("ITs an Array using instance of");
}
else{
    console.log("not an array");
}
```

OUTPUT :

```
harsh@Harshs-MacBook-Air Array % cd "/Users/harsh/weblab/JS/LAB2/Array"
harsh@Harshs-MacBook-Air Array % node "/Users/harsh/weblab/JS/LAB2/Array/methods.js"
java
python
s1
object
ITs an Array using instance of Array.isArray
ITs an Array using instance of
harsh@Harshs-MacBook-Air Array %
```

NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

ARRAY METHODS IN JAVASCRIPT :

CODE :

```
//Array methods

let subjects = ["java", "os", "python"];
for (i in subjects) {
    console.log(subjects[i]);
}
//LENGTH
console.log(subjects.length);

//toString
let res = subjects.toString();
console.log(res);


//PUSH POP
subjects.push("PUSHED");
console.log(subjects);
console.log(subjects.pop());

//JOIN
let x = subjects.join("|");
console.log(x);

console.log("\n");

//SHift and Unshift  QUEUE can be made using PUSH POP SHIFT UNSHIFT
subjects.shift();
console.log(subjects);

subjects.unshift("UNSHIFT");
console.log(subjects);

//CONCAT
{
    const s = ["java", "os", "python"];
    const u1 = ["t1", "t2"];
    const u2 = ["t11", "t22"];
    const result = s.concat(u1, u2);
    console.log(result);
}
```

NAME : HARSHRAJSINH ZALA REG NO : 22BCE2238

OUTPUT:

```
harsh@Harshs-MacBook-Air Array % cd "/Users/harsh/weblab/JS/LAB2
harsh@Harshs-MacBook-Air Array % node "/Users/harsh/weblab/JS/LA
java
os
python
3
java,os,python
[ 'java', 'os', 'python', 'PUSHED' ]
PUSHED
java|os|python


[ 'os', 'python' ]
[ 'UNSHIFT', 'os', 'python' ]
[ 'java', 'os', 'python', 't1', 't2', 't11', 't22' ]
harsh@Harshs-MacBook-Air Array %
```

SOME MORE METHODS

CODE :

```
//copywithin
//to copy within the array from the array itself
{
    const sub = ["java", "python", "os", "networks"];
    sub.copyWithin(1, 0);
    console.log(sub);
}
{ //using range
    const sub = ["java", "python", "os", "networks","hello","why"];
    sub.copyWithin(4,0,3);
    console.log(sub);
}

//SLICE AND SPLICE
{
    const sub = ["java", "python", "os", "networks"];
    const res =sub.slice(1,3); //ending index not included and returns an array
    console.log(res);
    const res2 = sub.slice(1); //from index 1 to end
    console.log(res2);
}
//SPLICE --> remove/add any number of elements from an index
{
    const sub = ["java", "python", "os", "networks","why","Hello"];
    let res = sub.splice(1,2); //splice(start index , how many elements)
    console.log(res);
```

```
    sub.splice(1,0,"physics","chemistry");
    console.log(sub);
    sub.splice(4,2,"physics","chemistry");
    console.log(sub);
}
```

OUTPUT :

```
harsh@Harshs-MacBook-Air Array % cd "/Users/harsh/weblab/JS/LAB2/Array"
harsh@Harshs-MacBook-Air Array % node "/Users/harsh/weblab/JS/LAB2/Array/arr-methods2.js"
[ 'java', 'java', 'python', 'os' ]
[ 'java', 'python', 'os', 'networks', 'java', 'python' ]
[ 'python', 'os' ]
[ 'python', 'os', 'networks' ]
[ 'python', 'os' ]
[ 'java', 'physics', 'chemistry', 'networks', 'why', 'Hello' ]
[ 'java', 'physics', 'chemistry', 'networks', 'physics', 'chemistry' ]
harsh@Harshs-MacBook-Air Array %
```