🕵🏾‍♂️

NaN, isNaN() and Number.isNaN()

- `NaN` is a read only property on the `window` object.
- `window.hasOwnProperty(”isNaN”)` returns `true`
- It is a **number** that represents “Not A Number” value. Hence: `typeof NaN //number`
- It has global scope.
- It is a falsy value `Boolean(NaN)` returns `false`
- We get `NaN` when we do following operations:
    - Number cannot be parsed.

      ```
      parseInt(”abcd”) //NaN
      ```

    - Math operation when the result is not real number

      ```
      Math.sqrt(-1);//NaN
      ```

    - Operand of an argument is `NaN`

      ```
      7*NaN //NaN
      ```

    - Indeterminate form

      ```
      Infinity*0//NaN
      undefined+undefined//NaN
      ```

    - Operations that involve string(and not numerical string like “1”), and doesn’t involve + operator

```
"abcd"/5 //NaN
"hi"*5 //NaN
```

- `NaN` is the only value in JS that is not equal to itself.

```
const x = 10;
x===NaN //false
x==NaN //false
NaN===NaN //false
NaN!==NaN //true
```

`window.isNaN()` VS `Number.isNaN()`

- `isNaN()` is a property on the window object.
- `isNaN()` returns true if passed value **coerces** to `NaN`

```
function isNaN(value){
  let value = Number(value);

  //since NaN is the only value that is not equal to itself.
  return value!==value;
}
```

```
//isNaN
isNaN(true) //false, bacause true converts to 1, which is a number
isNaN(null) //false, because flase converts to 0, which is a number
isNaN(undefined) //true because Number(undefined) returns NaN
isNaN("hello")//true because Number("hello") is NaN
isNaN(NaN) //true, since it is NaN
```

- But this coercion creates confusion. We want a function that returns true if the value **IS** `NaN` and not coerces it and then checks.
- `Number.isNaN()` comes into picture.
- `Number.isNaN()` doesn't coerce the value passed.
- It simply checks if the type of the value passed is a number, if not return false

```
//isNaN
Number.isNaN(true) //false, bacause true is not equal to NaN
Number.isNaN(null) //false, bacause null is not equal to NaN
Number.isNaN(undefined) //false bacause undefined is not equal to NaN
Number.isNaN("hello")//false bacause "hello" is not equal to NaN
Number.isNaN(NaN) //true bacause NaN is equal to NaN
```

- See how simple and intuitive it became.
- Polyfill for `Number.isNaN()`

```
Number.isNaN = Number.isNaN || function isNaN(input) {
    return typeof input === 'number' && input !== input;
}
```

- `Number.isNaN()` is relatively new as compared to `isNaN()` . It was added in ES6. Hence less browser support.
- If a browser doesn’t have `Number.isNaN()`use pollyfill.

Extra

`Object.is()` can be used to check if two values are same or not. Also `Object.is(NaN,NaN)` returns true.

References

- https://www.youtube.com/watch?v=GW6Qiblc5JQ(video by Steve Griffith)
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/NaN (MDN)
- https://www.codementor.io/@diegopalacios/nan-isnan-number-isnan-1agz7vzs08(article at CodeMentor by Diego Placias Lepore)
- https://dev.to/sedighian/everything-you-need-to-know-about-nan-in-javascript-113n(article by Sam Sedhigian)