Clarification isNaN()

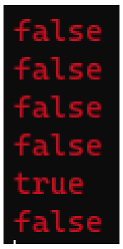
isNaN() vs Number.isNaN()

- **Number.isNaN(value):** A strict check that returns true if the value is exactly NaN. It doesn't coerce the value to a number, which means it only returns true if the value is already of the type Number and equals NaN.
- **isNaN(value):** A more lenient check that first tries to convert the value to a number before checking if it's NaN. This means it can return true for values that are not of type Number but can't be converted to a valid number.

In other words: **isNaN() converts the value to a number before testing it.**

Example

```
var numberString = '12';

var string = 'A';

var emptyString = '';


console.log(Number.isNaN(numberString));

console.log(Number.isNaN(string));

console.log(Number.isNaN(emptyString));


console.log(isNaN(numberString)); //global function

console.log(isNaN(string)); //global function

console.log(isNaN(emptyString)); //global function
```



```
false
false
false
false
true
false
```

Explanation

- Number.isNaN(numberString) checks if '12' is NaN. '12' is a string representing a number but not of type Number, so Number.isNaN returns false.
- Number.isNaN(string) checks if 'A' is NaN. Since 'A' is a string and not of type Number, Number.isNaN returns false.
- Number.isNaN(emptyString) checks if '' (an empty string) is NaN. Since it's a string and not a NaN value of type Number, Number.isNaN returns false.
- isNaN(numberString) checks if '12' can be converted to a number. Since '12' can be converted to the number 12, isNaN returns false.
- isNaN(string) checks if 'A' can be converted to a number. Since 'A' cannot be converted to a valid number, isNaN tries to convert it and fails, so it returns true.
- isNaN(emptyString) checks if '' (an empty string) can be converted to a number. An empty string is considered as 0 when converted to a number, so isNaN returns false.