JavaScript Numbers - Interview Cheat Sheet

1. Primitive vs Object Numbers

```
let num1 = 231;                 // Primitive number
let num2 = new Number(231);     // Number object
let num3 = new Number(231);     // Another Number object

console.log(num1 == num2); // true  (type coercion)
console.log(num2 == num3); // false (different memory)

typeof num1  "number"
typeof num2  "object"
```

2. parseInt() vs parseFloat() vs Number()

```
parseInt("123.45px")  123
parseFloat("123.45px")  123.45
Number("123.45px")  NaN

parseInt("abc123")  NaN
Number("123")  123 (strict)
parseInt("123abc")  123 (flexible)
```

3. isNaN() vs Number.isNaN()

```
isNaN("hello")  true  (due to coercion)
Number.isNaN("hello")  false  (no coercion)

Always prefer: Number.isNaN() for accuracy
```

4. toString(radix)

```
let num = 15;
num.toString(2)  "1111"     (binary)
num.toString(8)  "17"       (octal)
num.toString(16)  "f"       (hex)
```

5. toFixed(n)

```
let num = 5.6789;
num.toFixed(2)  "5.68"  (returns string)
num.toFixed(0)  "6"
```

6. toPrecision(n)

```
let num = 123.456;
num.toPrecision(2)  "1.2e+2"
num.toPrecision(4)  "123.5"
num.toPrecision(6)  "123.456"

let small = 0.00000012345;
small.toPrecision(1)  "1e-7"
```

7. valueOf()

```
let obj = new Number(42);
obj.valueOf()  42  (primitive value)
```

8. toExponential()

```
let num = 123456;
num.toExponential()  "1.23456e+5"
num.toExponential(2)  "1.23e+5"
```

9. Binary, Octal, Hex in JS

```
Binary: let bin = 0b1010;  10
Octal: let oct = 0o755;   493
Hex:   let hex = 0x1A3F;  6719
```

10. Binary Octal Conversion

```
Binary  Octal:
let bin = "101011";
let dec = parseInt(bin, 2);
let oct = dec.toString(8); // "53"

Octal  Binary:
let oct = "53";
let dec = parseInt(oct, 8);
let bin = dec.toString(2); // "101011"
```