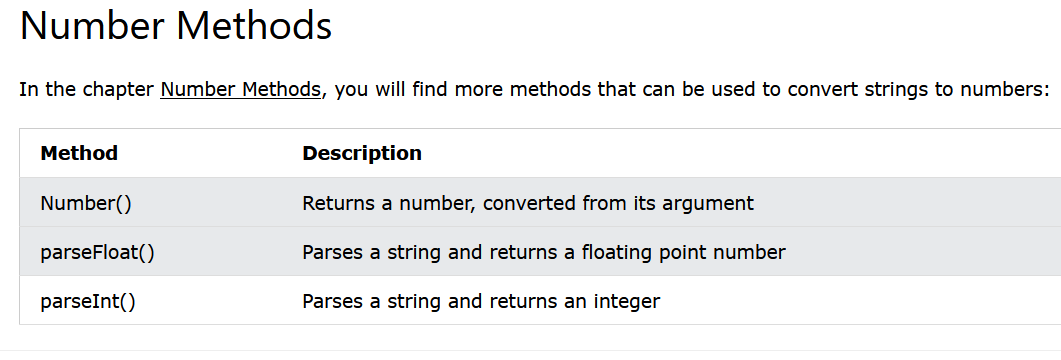
**Strings to Number**



Number( )

It converts any string Number passed in to it to Number type, number can ne int ot float

parseInt ( ) or Number.parseInt( )

It converts any Number passed in to it to integer number, for example : if 256.32 is passed than it only return us 256

“123ABC” as 123

basically it parse the passed string and returns us the Number ( int or float depending upon the method ) if there are any

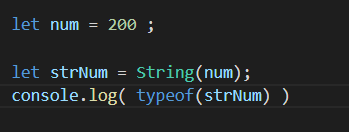
Similar thing could be achieved by Math.floor( )

parseFloat ( ) or Number.parseFloat( )

It converts any Number in string form passed in to it to floating point number, for example : if 256.01 is passed than it only return us 256.01

**Number to String**

**String( )**

Like Number( ) converts the passed string type of number to Number type

String( ) converts the Number type to String type

**.toString( ) also does the same thing**

**Actually .toString( ) is used to convert many data types to String type**

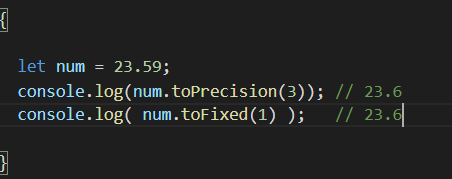
**toFixed( )**

Converts Number type to string type but we can specify the fix value and it will give us the rounded number value in string type

**toPrecision()**

here we pass overall length of number that we need in string form

If the passed length is less than the actual length of number it rounds off the number



Here both gives same result, so we can achieve desired task with only one also

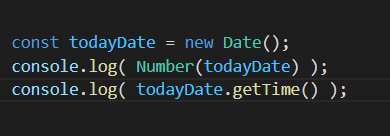
**Date ( ) to Number( )**

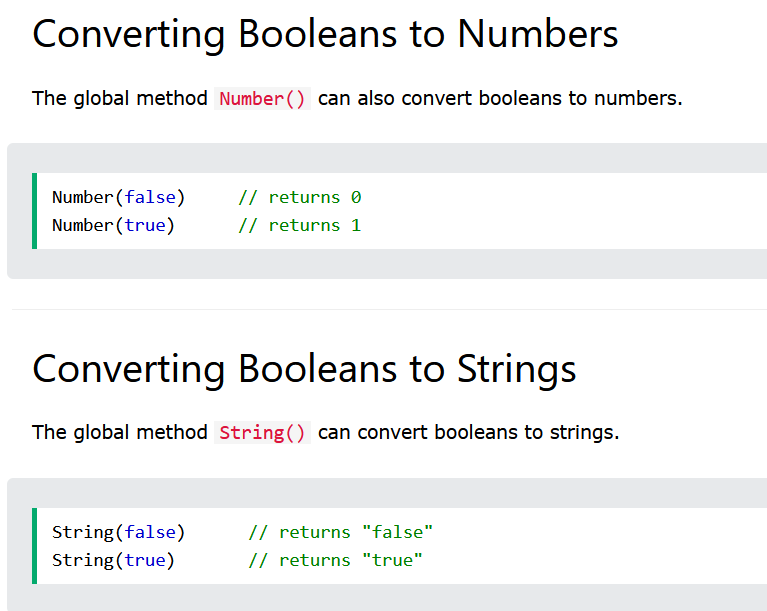
Number ( )

it converts everything passed it to Number type

So when we pass date we get the total milliseconds since Jan 1 1970

.getTime( ) also does similar thing



See Number ( ) converts everything passed to it to Number type of that data

String( ) converts everything passed it to String type equivalent of that data

here .toString( ) also does similar thing that String() does