Métodos estáticos
Objetos Date y Math

Métodos estáticos

- Son métodos como cualquier otro, pero que no necesitamos instanciar una clase para llamarlos.
- Es decir, no necesitamos crear una instancia de una clase con el operador **new**, podemos llamar al método directamente.

- Un ejemplo que hemos visto en el curso:
- const array = Array.of(‘a’, ‘b’, ‘c’)

Como crear métodos estáticos

- Usamos la palabra reservada static.
-
- Un ejemplo:

```
class Persona {
    constructor(nombre) { this.nombre = nombre; }

    static toString() { return `Nombre: ${this.nombre}`; }
}
```

Llamando a un método estático

- Con el ejemplo anterior:

- Persona.toString();

- Persona es el nombre de la clase
- toString() el método estático el cual queremos llamar.

Manejo de Fecha y Hora
El objeto Date

- Date es un objeto que nos permite trabajar con fechas y horas en JavaScript.
- Podemos hacer cálculos de fechas, crear una fecha en un formato determinado, obtener la fecha y hora actual, etc.
- Instancia del objeto Date:

```
> const date = new Date()
<- undefined
> date
<- Sun May 30 2021 14:57:25 GMT-0300 (hora estándar de Uruguay)
> |
```

- new Date() nos retorna una fecha representando la fecha actual (del sistema)

Crear una fecha personalizada

- Para crear una fecha personalizada hay varias formas, nosotros lo haremos pasándole al constructor de Date, una representación de la fecha en string.
- Ejemplo: Crearemos la fecha 23 de Abril de 2018, hora 13:45

  Lo haríamos de la siguiente forma:

- let fecha = new Date('Apr 23 2018 13:45:00');
- Debemos indicar el mes abreviado en inglés, seguido del día, el año, y como dato opcional la hora en el formato hh:mm:ss

  También podemos incluir milisegundos: hh:mm:ss:ms

Obtener día, mes y año

- A partir de una instancia del objeto date, podemos usar los métodos getDate(), getMonth() y getFullYear()

```
> date.getDate()
< 30
> date.getMonth()
< 4
> date.getFullYear()
< 2021
>
```

- getDate() nos retorna el dia del mes (de 1 a 31)
- GetMonth() nos retorna el numero del mes, pero del 0 al 11, siendo 0 el mes de enero y 11 el de diciembre.
- getFullYear() nos devuelve el numero del año de la fecha en formato completo, es decir, los 4 dígitos.

Cómo puedo mostrar la fecha:

```
> const d = new Date()
<· undefined
> d.getDate() + "/" + (d.getMonth() + 1) + "/" + d.getFullYear()
<· "30/5/2021"
>
```

Otra forma de mostrar la fecha toLocaleString()

```
> const hoy = new Date()
<- undefined
> hoy.toLocaleString()
<- "30/5/2021 18:35:42"
> hoy.toLocaleDateString()
<- "30/5/2021"
> hoy.toLocaleTimeString()
<- "18:35:42"
>
```

toLocaleString() : nos devuelve una cadena de la representación de la fecha y hora
toLocaleDateString() : sólo la fecha
toLocaleTimeString() : sólo la hora

Configurar toLocaleString()

- Este método de la clase Date admite parámetros **opcionales** de configuración:
- toLocaleString(idioma, opciones)
- Idioma:
- Será un string donde debemos establecer el idioma de la fecha.
- Ej: “en-US” (inglés americano)
- “es-UY” (español uruguayo)

Configurar toLocaleString()

- Parámetro de Opciónes:
- Será un objeto donde debemos establecer la configuración del string:

- { day: 'numeric', month: 'numeric', year: 'numeric', weekday: 'long' })
- Cada una de las propiedades es opcional.
- Numeric indica que el dato se mostrará en forma de número, y long en formato largo (palabra)
- La única propiedad que puede variar entre long y numeric es month

Configuración de toLocaleString() Ejemplos:

```
> const hoy = new Date()
< undefined
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'numeric', year: 'numeric', weekday: 'long' })
< "domingo 30/5/2021"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'long', year: 'numeric', weekday: 'long' })
< "domingo, 30 de mayo de 2021"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'long', year: 'numeric' })
< "30 de mayo de 2021"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'numeric', year: 'numeric' })
< "30/5/2021"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'numeric', weekday: 'long' })
< "domingo, 30/5"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'long', weekday: 'long' })
< "domingo, 30 de mayo"
> hoy.toLocaleString("es-UY", { day: 'numeric', month: 'long', hour: 'numeric', minute: 'numeric' })
< "30 de mayo 18:44"
>
```

Modificar una fecha ya creada

- Para esto utilizamos los métodos setDate(), setMonth() y setFullYear()
- También podemos modificar la hora, minutos, segundos, etc con setHours(), setMinutes() ....

```
> const fecha = new Date("apr 21 2018")
<- undefined
> fecha.toLocaleString()
<- "21/4/2018 00:00:00"
> fecha.setDate(15)
<- 1523761200000
> fecha.setMonth(0)
<- 1515985200000
> fecha.setFullYear(2008)
<- 1200362400000
> fecha.toLocaleString()
<- "15/1/2008 00:00:00"
>
```

El objeto Math

- Math es un objeto que tiene propiedades y métodos estáticos para resolver cálculos matemáticos.
- Nos servirá cada vez que queramos hacer algunos cálculos típicos, como calcular un logaritmo, una raíz cuadrada, una exponencial, hallar el seno, coseno, tangente, etc.
- Con Math también podremos generar números aleatorios.

```
> Math.trunc(2.8131)    // Parte entera
< 2
> Math.log10(100)    // logaritmo en base 10
< 2
> Math.log(10)    // logaritmo base e
< 2.302585092994046
> Math.max(0, 2, 4, 2)   // el máximo de todos sus argumentos
< 4
> Math.min(3, 2, 4, 2)   // el mínimo de todos sus argumentos
< 2
> Math.pow(2, 4)    // potencia base 2 exponente 4
< 16
```

Mas ejemplos de las métodos de Math

```
> Math.round(3.555)    // numero redondeado al entero mas cercano
< 4
> Math.sign(-2)    // devuelve f(x): Función matemática Signo
< -1
> Math.sqrt(2)    // Raíz cuadrada
< 1.4142135623730951
> Math.atan(3)    // arcotangente de un numero
< 1.2490457723982544
> Math.random()    // numero aleatorio entre 0 y 1
< 0.9918526095719671
>
```

Se recomienda ir a la documentación de JavaScript y consultar todos los métodos (son muchos, en el curso veremos algunos pocos)

- https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Math

Algunas propiedades de Math

- Estas son algunas de las propiedades, pero hay mas

```
> Math.PI    // Numero PI
< 3.141592653589793
> Math.E    // Numero e
< 2.718281828459045
> Math.LN2    // Valor de log(2) [Logaritmo Natural de 2]
< 0.6931471805599453
> Math.SQRT2    // Raíz cuadrada de 2
< 1.4142135623730951
>
```

Próxima clase:

- Aprenderemos a generar números aleatorios en un intervalo con Math.random()
- Trabajaremos con los métodos de los Strings.