

Programa académico CAMPUS



Trainer
Ing. Carlos H. Rueda C.



JAVASCRIPT

Date();

⚙ JS

```
1 var myDate = new Date('2018-08-26T19:00:00'),  
2 month = myDate.getMonth(),  
3 date = myDate.getDate(),  
4 day = myDate.getDay(),  
5 hours = myDate.getHours();
```

FECHAS EN JAVASCRIPT

Las fechas en JavaScript son **objetos**.

Internamente el tiempo es almacenado como la cantidad de milisegundos desde el 1 de enero de 1970.

TRABAJANDO CON FECHAS EN JAVASCRIPT

El objeto Date contiene la fecha y la hora

Internamente la hora está almacenada como la cantidad de milisegundos desde el 1 de enero de 1970.

Creando el objeto date

```
// ahora
const ahora = new Date();
// Tue Apr 18 2023 23:34:08 GMT-0500 (hora estándar de Colombia)

// una fcha especifica
// los meses empiezan de cero (0:ene, 1:feb, ...)
const randomdate = new Date(2015, 3, 12, 6, 25, 58);
//Sun Apr 12 2015 06:25:58 GMT-0500 (hora estándar de Colombia)

// una fcha especifica a media noche
const win95 = new Date(1995, 7, 27);
// Sun Aug 27 1995 00:00:00 GMT-0500 (hora estándar de Colombia)
```

Asignando valores

```
const ahora = new Date();  
  
ahora.setFullYear(2014); // asigna el año  
ahora.setMonth(3) // asigna Abril  
ahora.setDate(4); // asigna el día  
  
console.log(ahora);  
//Fri Apr 04 2014 23:40:54 GMT-0500 (hora estándar de Colombia)
```

Date/Time	Method	Range	Example
Year	<code>setFullYear()</code>	YYYY	1970
Month	<code>setMonth()</code>	0-11	0 = January
Day (of the month)	<code>setDate()</code>	1-31	1 = 1st of the month
Day (of the week)	<code>setDay()</code>	0-6	0 = Sunday
Hour	<code>setHours()</code>	0-23	0 = midnight
Minute	<code>setMinutes()</code>	0-59	
Second	<code>setSeconds()</code>	0-59	
Millisecond	<code>setMilliseconds()</code>	0-999	
Timestamp	<code>setTime()</code>	Milliseconds since Epoch time	

Obteniendo valores

```
const ahora = new Date();  
// Todos los metodos empieza por get  
console.log(ahora.getMonth());  
// milisegundos desde 1 enero 1970  
console.log(ahora.getTime());  
// Dia de la semana (domingo = 0)  
console.log(ahora.getDay());
```


Obteniendo valores

```
console.log(now.getFullYear());  
console.log(now.getMonth());  
console.log(now.getDate());  
console.log(now.getDay());  
console.log(now.getHours());  
console.log(now.getMinutes());  
console.log(now.getSeconds());
```

Date strings

También existen métodos que nos permiten obtener strings de fechas en diferentes **formatos**, como el método `toDateStrin()`:

```
console.log(now.toDateStrin());
```

Date strings

O el método `toISOString()`, que nos devolvería un string con la hora local actual:

```
console.log(now.toISOString());
```

Date strings

Para obtener un string con una fecha en formato corto y una hora, podemos utilizar el método `toLocaleString()`:

```
console.log(now.toLocaleString());
```

Timestamps como herramienta para comparar fechas

Una manera popular para comparar fecha es el método `getTime()`.

Este método nos devuelve los mili-segundos que han pasado desde el 01/01/1970 a las 12 am.

```
console.log(now.getTime());
```

Timestamps como herramienta para comparar fechas

Imaginemos que queremos saber la diferencia de tiempo existente entre una fecha del pasado (el 8 de Octubre de 2019) y hoy.

```
const now = new Date();  
const before = new Date('October 8 2019 16:30:00');  
const diff = now.getTime() - before.getTime();  
console.log(diff);
```

Timestamps como herramienta para comparar fechas

Esto está bien pero queremos algo más útil, como los días transcurridos entre una fecha y otra.

```
const minutes = Math.round(diff / 1000 / 60);  
const hours = Math.round(minutes / 60);  
const days = Math.round(hours / 24);  
  
console.log(minutes, hours, days); // 7323 122 5  
console.log(`This post was written ${days} ago`); // This post was written 5 ago
```

new Date(2021,
2, 28);

Year: 2021

Day: 28th

Month: March



EJERCICIO FECHAS JS

Una librería quiere calcular los días que han pasado entre la fecha de entrega pactada de un libro y su entrega, para efectos de cobrar multa.

- Construye un formulario web que reciba la fecha de entrega pactada de un libro y la fecha de entrega real.
- Calcula y muestra cuantos días han pasado y la multa o sanción.
- Si es menor igual a 3 días cobra una multa de \$7.000 pesos por día.
- Si es mayor a 5 días cobra una multa de \$10.000 pesos y una sanción de no préstamo por 6 meses.

PARA APRENDER MAS

Para facilitarnos la vida con el trabajo de fecha, lo recomendable es trabajar con librerías de fechas.

[Date-fns](#) es una librería muy recomendada para trabajar con fechas.

DATE-FNS

SO HOT RIGHT NOW

EJEMPLO DE FECHAS JS

El usuario ingresa un fecha (dd-mm-yyyy) y ud debe hacer un programa web que día de la semana cayó.

```
function obtenerDiaSemana(fecha) {  
    const diasSemana = ['domingo', 'lunes', 'martes', 'miércoles', 'jueves', 'viernes', 'sábado'];  
    const partesFecha = fecha.split('-');  
    const fechaObjeto = new Date(partesFecha[2], partesFecha[1] - 1, partesFecha[0]);  
    const diaSemana = diasSemana[fechaObjeto.getDay()];  
    return diaSemana;  
}  
console.log(obtenerDiaSemana('19-04-2023')); // Ejemplo de salida: "miércoles"
```

EJEMPLO DE FECHAS JS

El usuario ingresa dos fechas y se le propone que construya un programa que calcule la diferencia en días entre estas dos.

```
function calcularDiferenciaEnDias(fecha1, fecha2) {  
  const unDia = 24 * 60 * 60 * 1000; // milisegundos en un día  
  const fechaInicial = new Date(fecha1);  
  const fechaFinal = new Date(fecha2);  
  const diasTranscurridos = Math.round(Math.abs((fechaFinal - fechaInicial) / unDia));  
  return diasTranscurridos;  
}  
  
// Ejemplo de uso:  
const fecha1 = '2023-01-01';  
const fecha2 = '2023-04-19';  
const diferenciaEnDias = calcularDiferenciaEnDias(fecha1, fecha2);  
console.log(`La diferencia en días entre ${fecha1} y ${fecha2} es ${diferenciaEnDias}.`);  
//Resultado 108
```

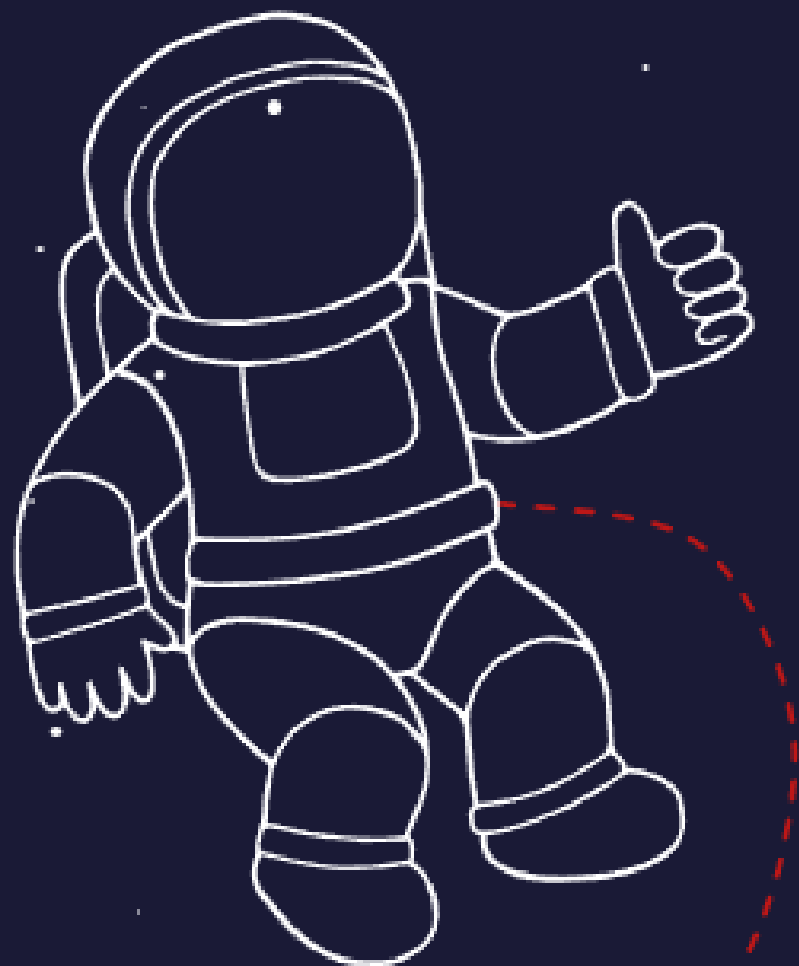
EJERCICIOS CON FECHAS

1. El usuario ingresa dos fechas y se le propone que construya un programa que calcule la diferencia en días entre estas dos.
2. La oficina de cartería desea que le construya un programa web al cual se le ingresa una fecha y que este calcule y muestre que cual es la fecha que cae después de 90 días hábiles.
3. Hacer un función que reciba la hora actual y la que regrese esta misma pero en diferente zona horaria. *Ejemplo* de función:

```
function calcZonaHor(hora, 'America/New_York')
```

EJERCICIOS CON FECHAS

4. Un servicio bancario desea poner a disposición de sus usuarios una aplicación web donde estos puedan ingresar la fecha de corte de su tarjeta de credito y la aplicación le muestre cuál es la fecha límite de pago. Esta fecha límite se calcula, sumando 15 días calendario a la fecha de corte más tres días de gracia.
5. Hacer un programa web que reciba una fecha (19-04-2023) y le de el formato de: "lunes, 19 de abril de 2023"



Programa académico CAMPUS



Trainer
Ing. Carlos H. Rueda C.

