



Comunicación Persona-Máquina

Internacionalización en Javascript

¿Qué es internacionalización?

- Internacionalización (**I18n**) en interfaces consiste en adaptar el contenido y la presentación a diferentes idiomas y localizaciones.
- En este tema se va introducir la internacionalización en Javascript usando la API de internacionalización que provee el objeto global **Intl**.

¿Qué se puede hacer con Intl?

- Intl puede crear objetos para tratar diferentes temas de l18n, pero en este tema se van a ver 2:
 - `Intl.DateTimeFormat`: para formatear fechas y horas.
 - `Intl.NumberFormat`: para formatear números.

Idioma y país: Locales

- Para utilizar objetos `Intl` hay que indicar un idioma y una localización (`Locales`).
- Si no se indica explícitamente se tomarán las del sistema.
- Se indica con una cadena formada por abreviaturas separadas por '-'.
 - 'es' : español
 - 'en-UK' : inglés del reino unido

Formato de fechas

- Hay que crear un objeto `Intl.DateTimeFormat` para poder convertir un objeto `date` en un texto

`new Intl.DateTimeFormat(Locales[,Options])`

Algunas opciones:

dateStyle / timeStyle:	weekday :
"full"	"long" (e.g.,
"long"	Thursday)
"medium"	"short" (e.g., Thu)
"short"	"narrow" (e.g., T).

Ejemplo:

```
var dateFormat=
```

```
    new Intl.DateTimeFormat(MyLocal,{dateStyle: 'medium'});
```

Formato de fechas

- El método `format(date)` convierte la fecha en una cadena con las especificaciones indicadas en `DateTimeFormat`.

Ejemplo:

```
> var dateFormat=  
    new Intl.DateTimeFormat('fr',{dateStyle: 'medium'});  
console.log(dateFormat.format(new Date()));
```

15 avr. 2020

< undefined

```
> var dateFormat=  
    new Intl.DateTimeFormat('es-ES',{dateStyle: 'medium'});  
console.log(dateFormat.format(new Date()));
```

15 abr. 2020

< undefined

```
> var dateFormat=  
    new Intl.DateTimeFormat('en-US',{dateStyle: 'medium'});  
console.log(dateFormat.format(new Date()));
```


Apr 15, 2020

Formato de números

- Hay que crear un objeto `Intl.NumberFormat` para poder convertir un numero en un texto

new Intl.NumberFormat (Locales [, Options])

Algunas opciones:

Style: "decimal" "currency" "percent" "unit"	 Must be defined	currency: "USD" "EUR" ... <u>List</u>
---	---	--

Ejemplo:

```
var dateFormat=  
    new Intl.NumberFormat(MyLocal,{Style: 'decimal'});
```

Formato de números

- El método `format` (`Number`) convierte el número en una cadena con las especificaciones indicadas en `NumberFormat`.

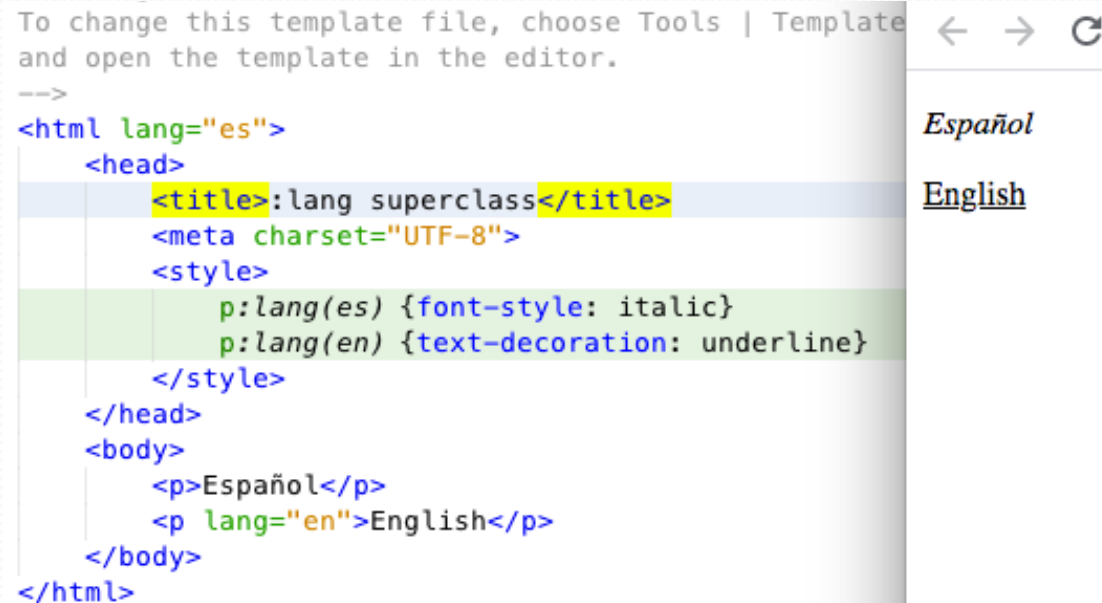
Ejemplo:

```
> var numberFormat=  
    new Intl.NumberFormat('en',{style: 'currency',  
                                currency: 'USD'});  
console.log(numberFormat.format(0.128));  
$0.13  
< undefined  
  
> var numberFormat=  
    new Intl.NumberFormat('es-ES',{style: 'currency',  
                                currency: 'EUR'});  
console.log(numberFormat.format(0.128));  
0,13 €  
< undefined  
  
> var numberFormat=  
    new Intl.NumberFormat('en',{style: 'currency',  
                                currency: 'GBP'});  
console.log(numberFormat.format(0.128));  
£0.13
```


CSS dependiente del idioma

- La pseudo-clase **:lang** permite especificar reglas CSS para elementos según el idioma indicado en *lang*.
- El parámetro lang se hereda de un componente a todos los hijos. Si se quiere indicar que toda la página es de un determinado idioma se ha de indicar en `<HTML>`.

Ejemplo:



```
To change this template file, choose Tools | Template
and open the template in the editor.
-->
<html lang="es">
  <head>
    <title>:lang superclass</title>
    <meta charset="UTF-8">
    <style>
      p:lang(es) {font-style: italic;}
      p:lang(en) {text-decoration: underline;}
    </style>
  </head>
  <body>
    <p>Español</p>
    <p lang="en">English</p>
  </body>
</html>
```

← → ↻

Español

English