



Les formats des Dates en chaîne de caractères

Le format par défaut : chaîne de caractères `toString()`

Lorsque vous appelez la méthode `toString()` ou que vous utilisez la console, c'est le format utilisé par défaut, elle affiche la date dans l'heure locale du système en précisant le fuseau horaire :

```
const date = new Date();  
date; // Fri Jan 10 2020 18:16:49 GMT+0100  
date.toString(); // Fri Jan 10 2020 18:16:49 GMT+0100
```



La différence entre les deux est que `date` contient un objet `Date` et c'est simplement le navigateur qui affiche par commodité la date au format chaîne de caractères.

En revanche, `date.toString();` retourne une chaîne de caractères.

Ce format n'est pas du tout recommandé pour le stockage car il n'est pas standardisé entre les navigateurs. Il sert simplement à l'affichage lors du développement.

Le format chaîne de caractères `toDateString()`

La méthode `toDateString()` renvoie une chaîne de caractères contenant la date en anglais.

```
const date = new Date();  
date.toDateString(); // "Fri Jan 10 2020"
```



Les trois premières lettres indiquent le jour en anglais.

Les trois lettres d'après indiquent le mois en anglais.

Les deux chiffres suivant indiquent le jour du mois.

Enfin, les quatre derniers chiffres indiquent l'année.

Ce format est standardisé mais est recommandé uniquement pour l'affichage en développement.

En effet, il ne contient pas le fuseau horaire de référence et ne peut donc être utilisé pour le stockage.

Et il ne convient pas à l'affichage aux utilisateurs car il contient des abréviations et affiche uniquement une chaîne de caractères en anglais.

Le format chaîne de caractères UTC avec `toUTCString()`

La méthode `toUTCString()` convertit une date en une chaîne de caractères au fuseau horaire UTC

```
const date = new Date();  
date.toUTCString(); // "Fri, 10 Jan 2020 17:40:08 GMT"
```



Ce format est également standardisé. Il permet d'inclure également le fuseau horaire ce qui en fait une date fiable. Cependant, elle ne convient pas particulièrement au stockage. Elle n'est pas assez précise (pas de millisecondes).

Le format ISO avec `toISOString()`

Une date ISO est une date qui respecte le standard international ISO 8601 .

Ce standard permet de spécifier la représentation numérique d'une date et d'une heure. Il utilise le calendrier grégorien et le système horaire de 24 heures.

Elle est parfaite pour le stockage car elle contient le fuseau horaire et est utilisée dans quasiment tous les langages informatiques.

La méthode `toISOString()` renvoie une chaîne de caractères au format ISO .

Le format est le suivant :

```
YYYY-MM-DDTHH:mm:ss.sssZ
```



Les parties de la date sont séparées par des tirets - .

YYYY est l'année.

MM le mois et pas l'index du mois, donc janvier est 01.

DD le jour.

T pour `time` , le séparateur avant l'heure. Les parties de l'heure sont séparées par des : .

HH pour l'heure.

mm pour les minutes.

ss pour les secondes.

sss pour les millisecondes. Elles sont séparés des secondes par un . .

Le suffixe Z signifie zéro décalage avec UTC .

Exemple :

```
const date = new Date();  
date.toISOString(); "2020-01-10T17:55:32.094Z"
```

