

Logging

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en Programmation par contrainte (IA)
Ingénieur en Génie logiciel

`elmouelhi.achref@gmail.com`

4 type de messages de journalisation avec JavaScript

- `console.log()` : pour affiché un message non formaté
- `console.info()` : pour affiché un message en bleu
- `console.warn()` : pour affiché un message d'avertissement
- `console.error()` : pour affiché un message d'erreur

Exemple

```
console.log("log");  
// affiche un message en noir dans la console  
  
console.info("info");  
// affiche un message en bleu dans la console  
  
console.warn("warn");  
// affiche un message + une icône en orange dans la  
  console  
  
console.error("error");  
// affiche un message + une icône en rouge dans la  
  console
```

On peut aussi personnaliser les couleurs en utilisant des propriétés CSS et le paramètre %c (compatible avec chrome firefox)

```
console.log("%chello world", "background:red; color: white");
```

On peut aussi personnaliser les couleurs en utilisant des propriétés CSS et le paramètre %c (compatible avec chrome firefox)

```
console.log("%chello world", "background:red; color:white");
```

Pour afficher un messages avec des propriété CSS différente

```
console.log("%chello " + "%cworld", "background:red; color:white", "background:blue; color:red" );
```

Il est possible de créer un groupement de messages

```
console.group("Premier groupe de message");  
console.log("première ligne");  
console.log("deuxième ligne");  
console.log("dernière ligne");  
console.groupEnd();
```

Il est possible de créer un groupement de messages

```
console.time("Test process");  
// on ajoute du code JS ici  
console.timeEnd("Test process");
```

Il est possible de créer un groupement de messages

```
console.time("Test process");  
// on ajoute du code JS ici  
console.timeEnd("Test process");
```

Le contenu de `console.time` et `console.timeEnd` doit être le même. Ça correspond au nom du bloc pour lequel on veut calculer le temps d'exécution.

Pour éviter d'écrire chaque fois `console.log`, on peut créer une variable contenant le nom de la fonction

```
var cl = console.log;
```

Pour éviter d'écrire chaque fois `console.log`, on peut créer une variable contenant le nom de la fonction

```
var cl = console.log;
```

Puis utiliser ce raccourci pour afficher de messages

```
cl("hello");  
// affiche hello
```