

Introduction au JavaScript

<https://fr.javascript.info>

Attribution - Pas d'Utilisation Commerciale 4.0 International (CC BY-NC 4.0)

Comprendre les bases de la syntaxe JavaScript

- L'importance de la ponctuation
- Javascript vs ECMAScript
- Les instructions en Javascript
- Les commentaires en Javascript
- ES5 vs ES6 : “use strict”
- Qui est derrière le standard ?

L'importance de la ponctuation

Rien de bien catastrophique !

Rien de bien, catastrophique !

On mange, les enfants !

On mange les enfants !

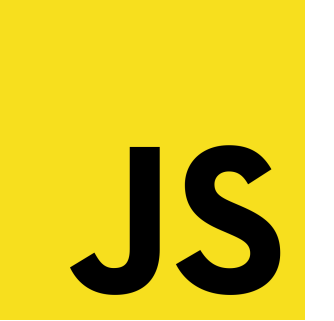




```
document.addEventListener("click", () => alert("Merci d'avoir cliqué"));
```

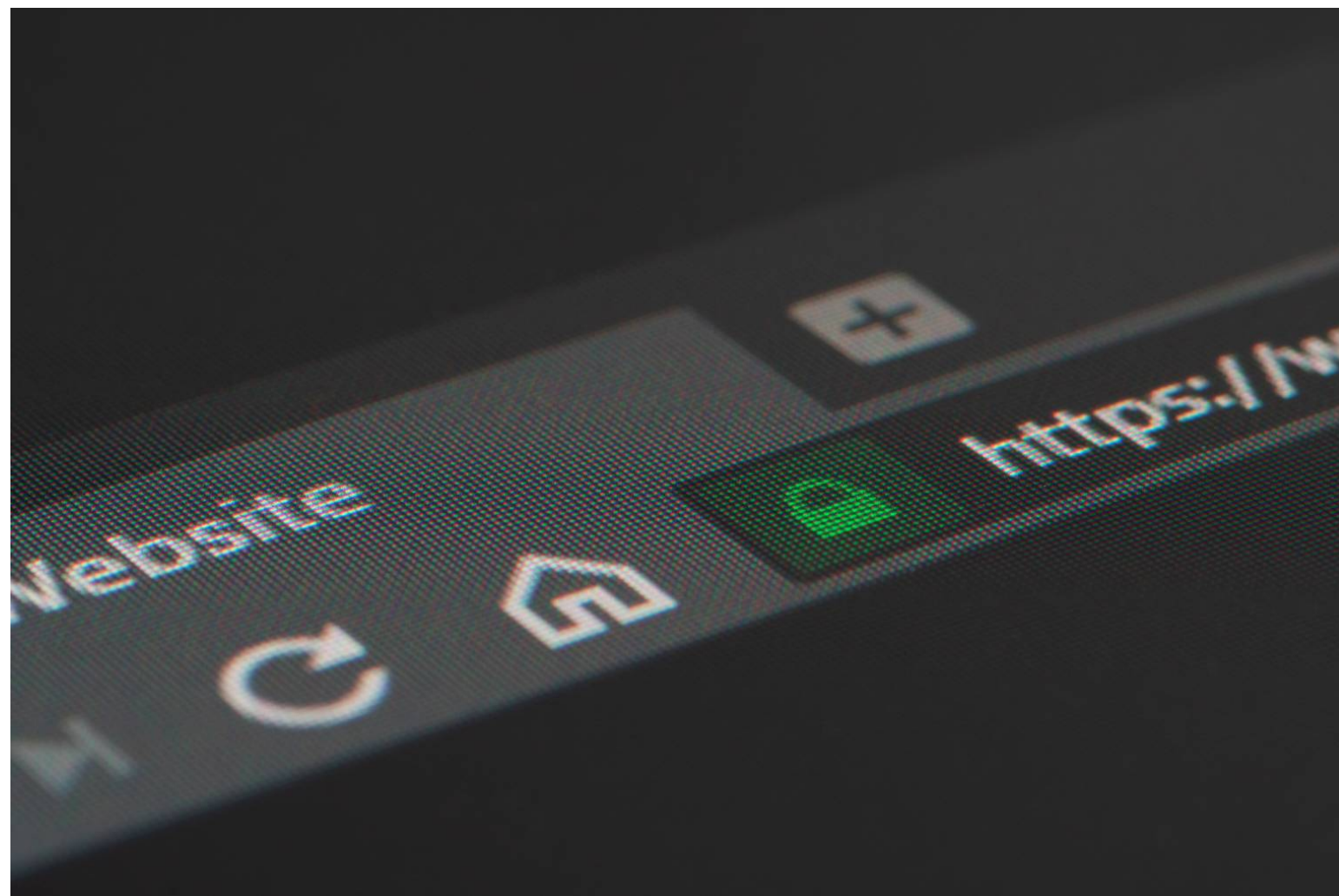
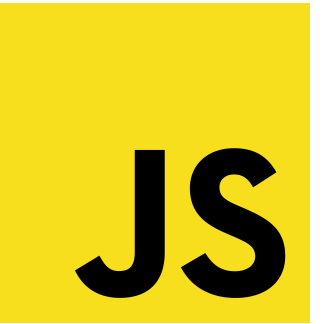
The diagram consists of five green arrows. One arrow points from the top-left towards the word 'addEventListener' in the code. Another arrow points from the top-left towards the word 'click'. A third arrow points from the top-left towards the opening parenthesis of the arrow function. A fourth arrow points from the top-left towards the word 'alert'. A fifth arrow points from the top-left towards the string 'Merci d'avoir cliqué'.

DOM => Navigateur



```
document.addEventListener("click", () $\Rightarrow$  alert("Merci d'avoir cliqué"));
```

ECMAScript = Standard



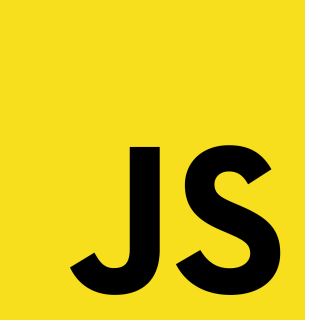
<https://fr.javascript.info>

Attribution - Pas d'Utilisation Commerciale 4.0 International (CC BY-NC 4.0)

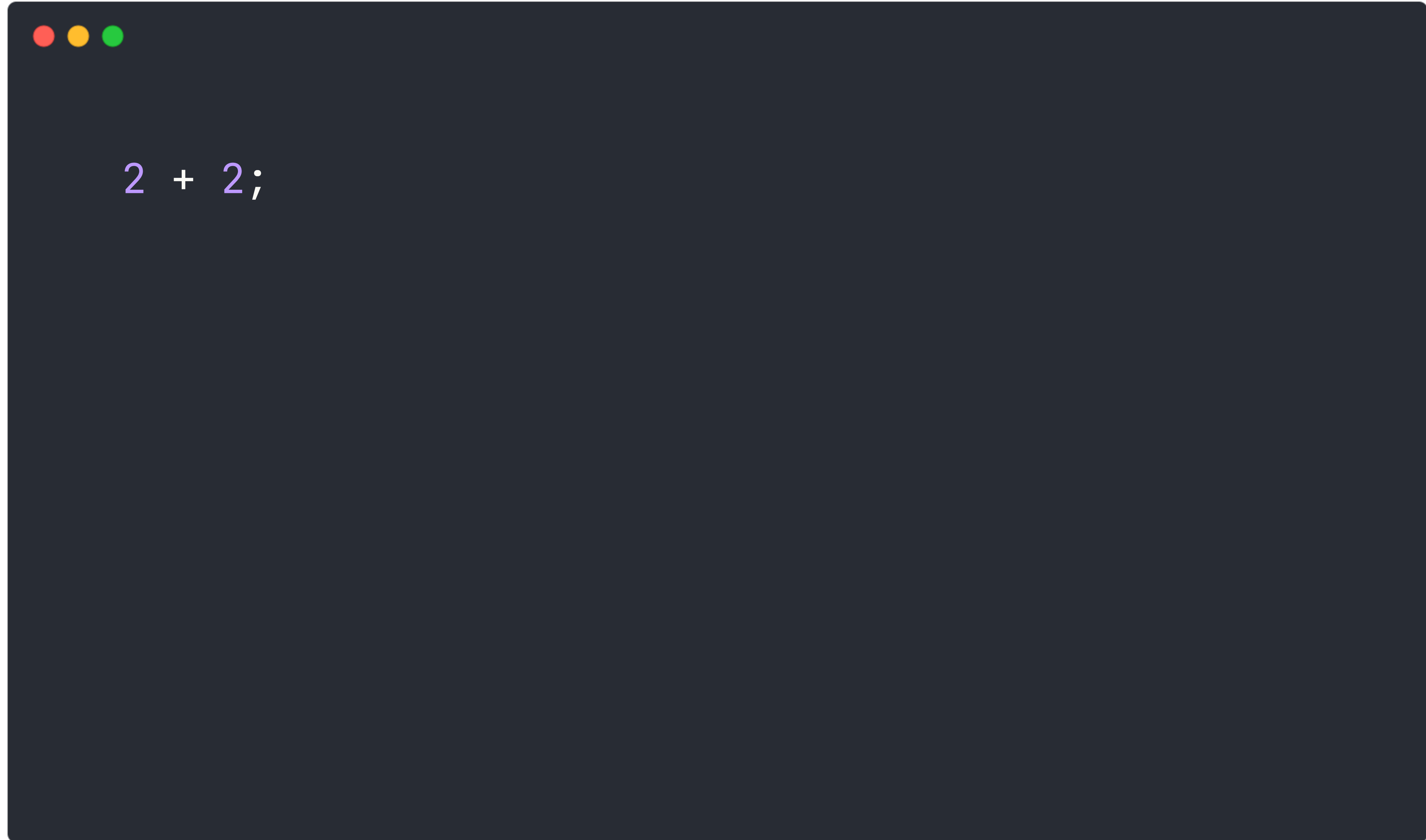
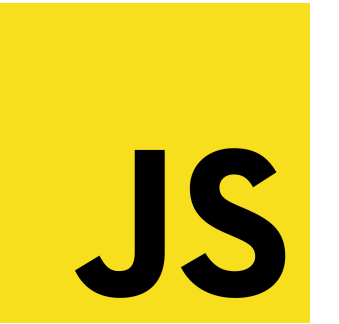


```
alert("Hello world!");
```

JavaScript = ECMAScript + Environnement

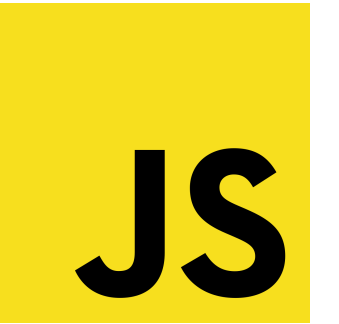


Les instructions en JavaScript

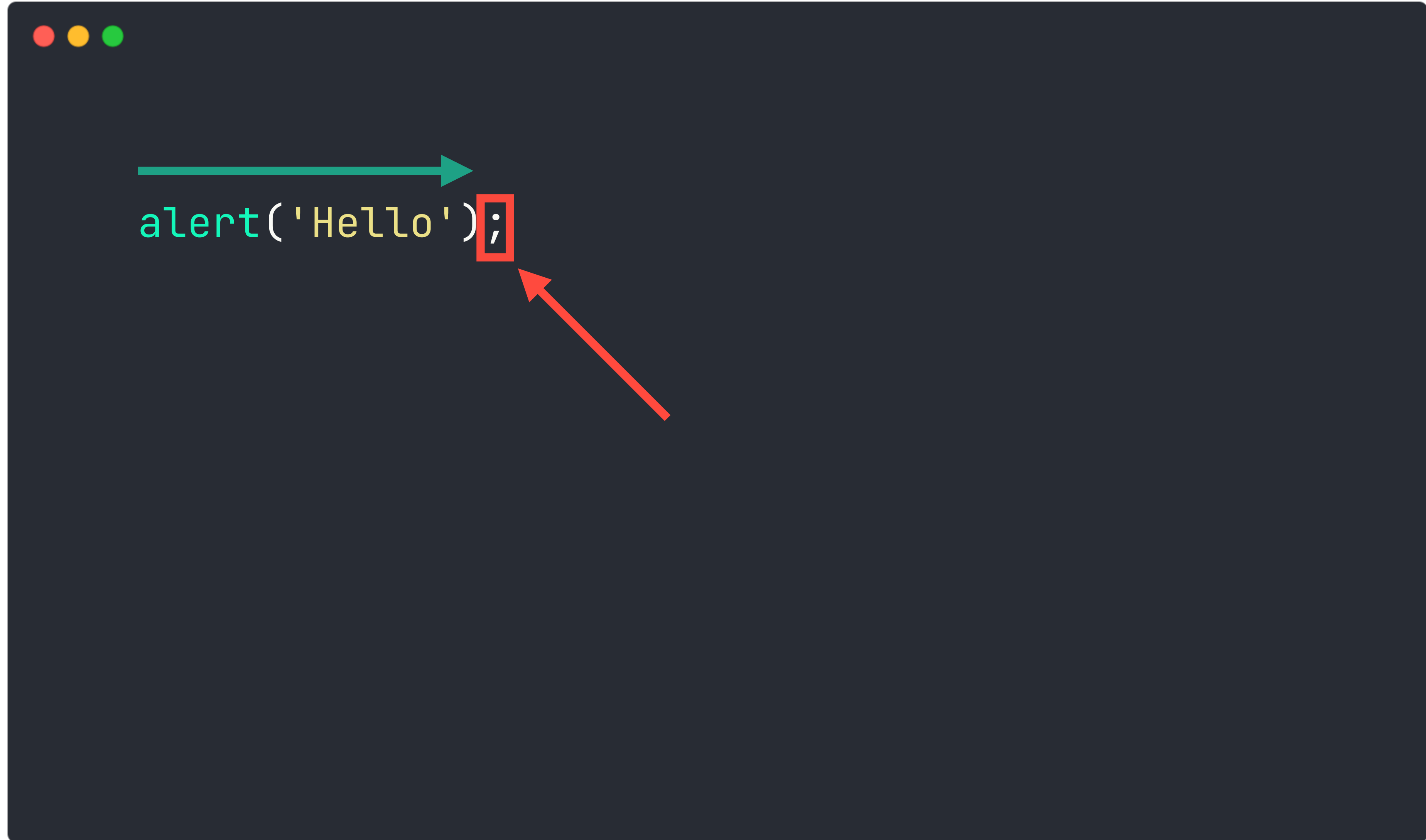


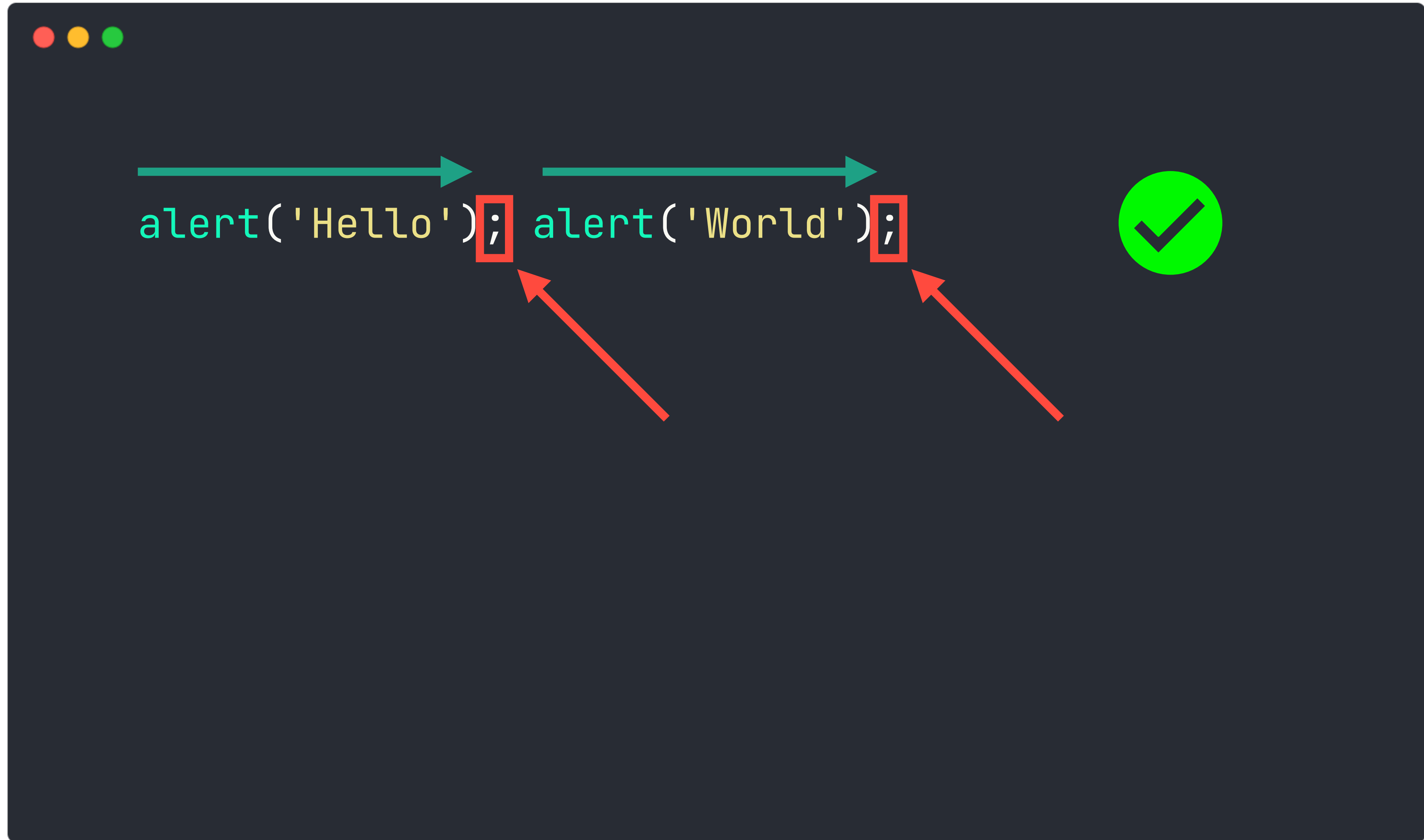
<https://fr.javascript.info>

Attribution - Pas d'Utilisation Commerciale 4.0 International (CC BY-NC 4.0)



```
2 + 2 * 4 - 3;
```







```
alert('Hello') alert('World')
```

 **ERROR**

```
alert('Hello')  
alert('World')
```



insertion automatique de point-virgule

```
alert(3 +  
1  
+ 2);
```



```
[1, 2].forEach(alert)
```



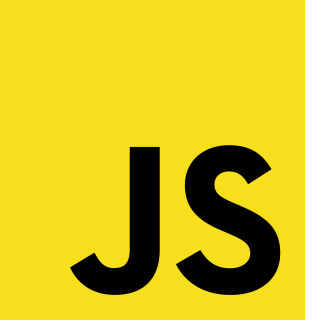
```
alert("There will be an error")
```

```
[1, 2].forEach(alert)
```

 **ERROR**

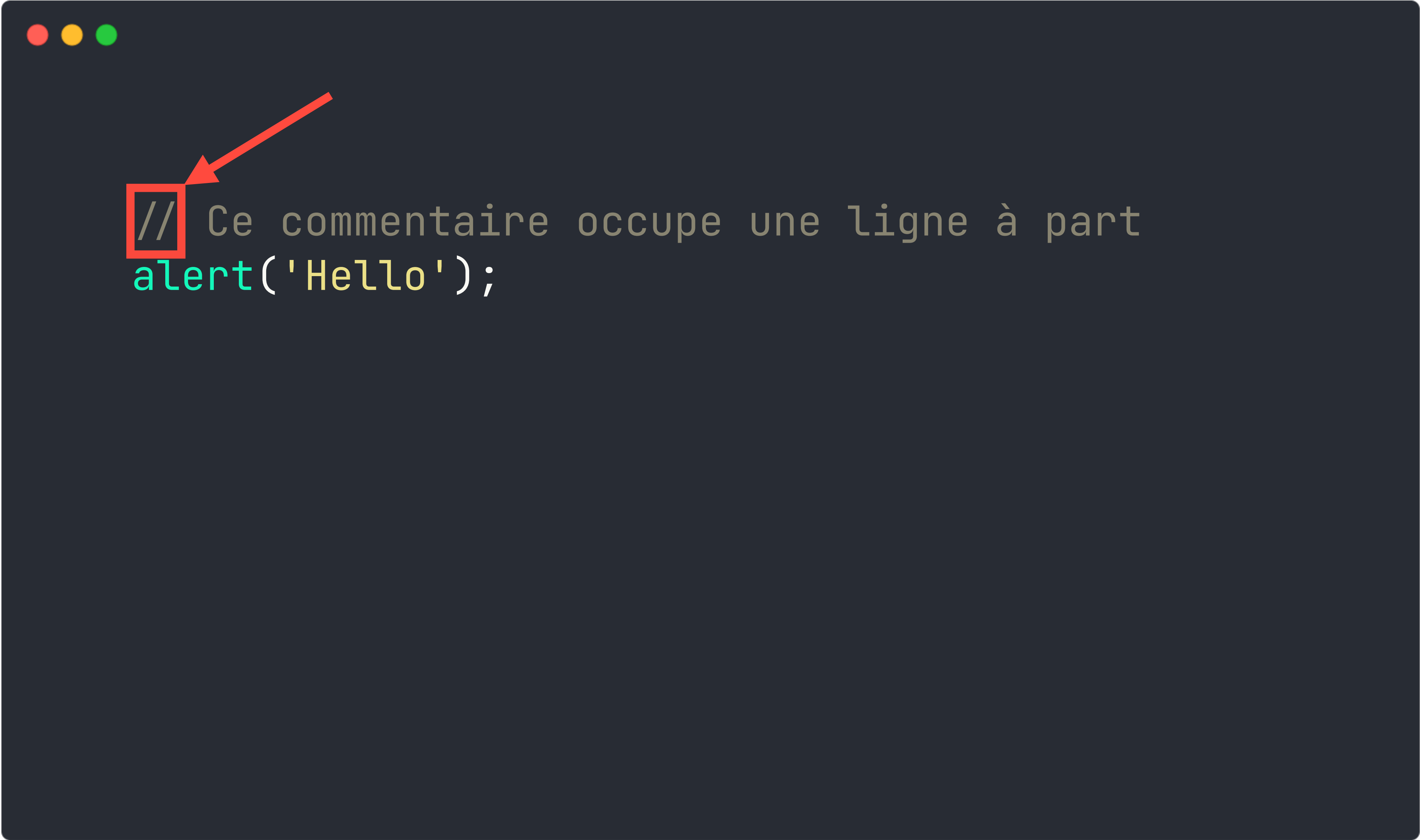
```
alert("There will be an error")[1, 2].forEach(alert)
```



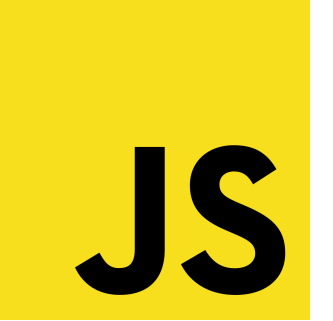


Les commentaires

Les commentaires sont des **portions du code source ignorées**
par le compilateur ou l'interpréteur,
car **destinées en général à un lecteur humain**
et non censées influencer l'exécution du programme.



```
// Ce commentaire occupe une ligne à part  
alert('Hello');
```



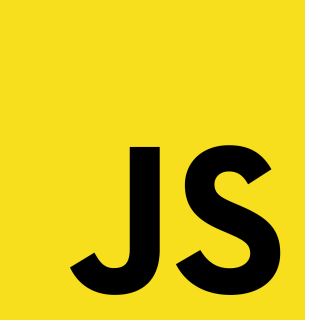
```
alert('World'); // Ce commentaire suit l'instruction
```



```
/* Un exemple avec deux messages.  
   C'est un commentaire multiligne.  
*/  
alert('Hello');  
alert('World');
```

Début du commentaire

Fin du commentaire



```
/* Commenter le code  
alert('Hello');  
*/  
alert('World');
```

Le mode moderne, "use strict"



ECMAScript

L'apparition du **mode strict**

ES5

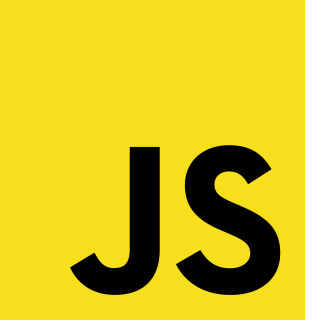
AVANT Juin 2015

ES6

APRÈS Juin 2015

Modules, classes, let/const,
destructuration, asynchrone, etc ...

"use strict"



```
"use strict";
```

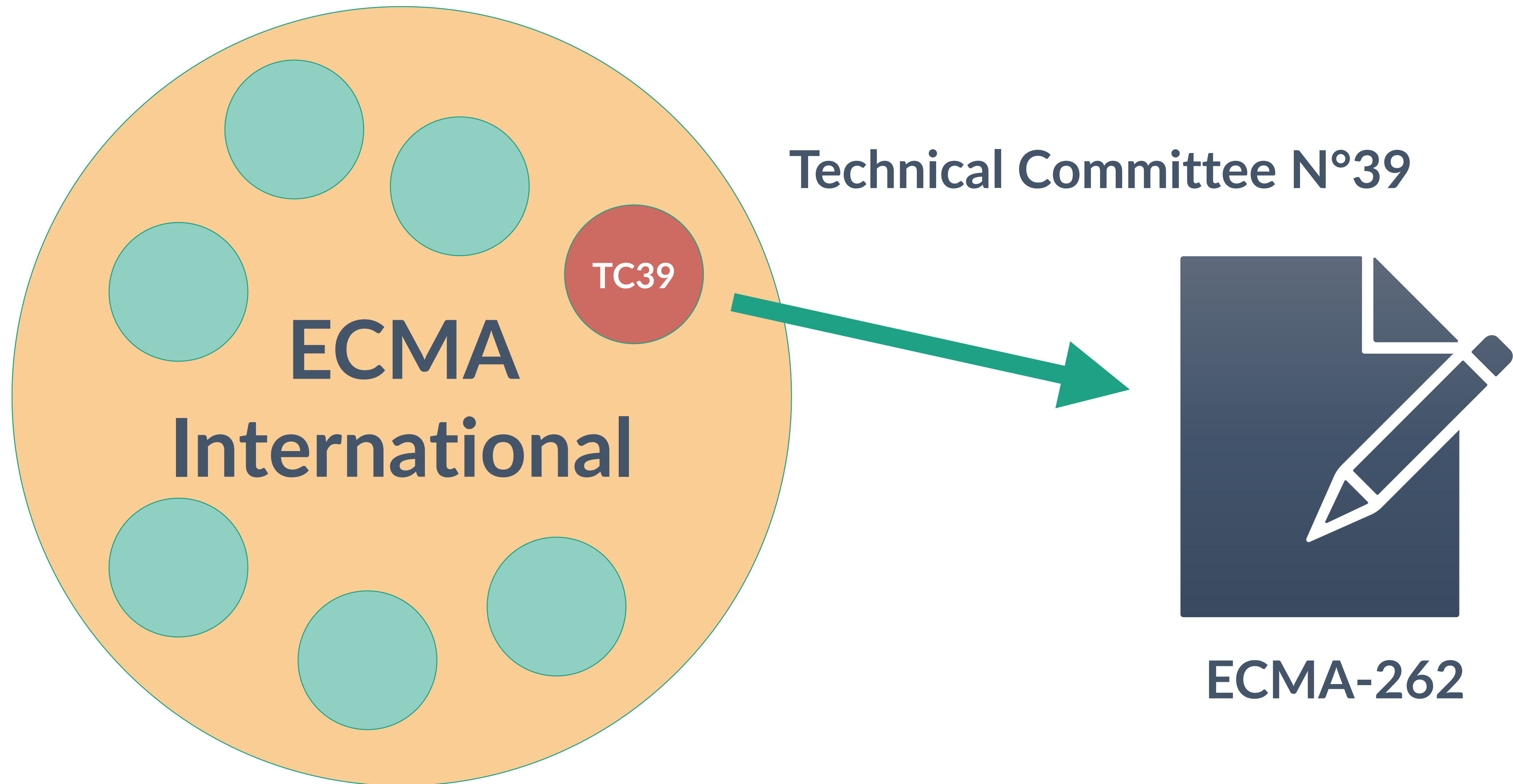
```
// ce code fonctionne de manière moderne (ES6+)
```

```
...
```

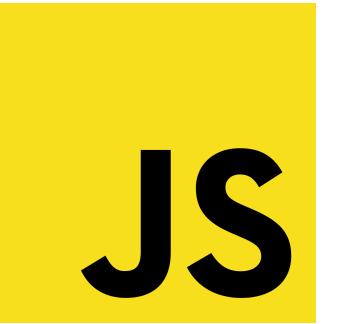
```
alert("some code");  
/* "use strict" ci-dessous est ignoré  
il doit être en haut */  
  
"use strict";  
  
// le mode strict n'est pas activé
```


Par défaut la console du navigateur
n'utilise pas le "use strict"

ECMAScript, TC39, ECMA-262 ?



ECMA-262 = Document **ECMAScript**



ECMA-262



ECMA International



Javascript



TC39

<https://fr.javascript.info>

Attribution - Pas d'Utilisation Commerciale 4.0 International (CC BY-NC 4.0)

