

# KSI Centrale Marseille

La Web Track

# Projet 2 : le Taquin



# Se former

On continue avec les tutoriels du NetNinja, pour JavaScript cette fois-ci :

[https://www.youtube.com/watch?v=qoSksQ4s\\_hg&list=PL4cUxeGkcC9i9Ae2D9Ee1RvylH38dKuET](https://www.youtube.com/watch?v=qoSksQ4s_hg&list=PL4cUxeGkcC9i9Ae2D9Ee1RvylH38dKuET)

Et toujours MDN pour la doc si vous avez besoin d'infos sur certains objets de base :

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

Pour comprendre le jeu, rien de tel que de jouer un peu :

<https://taquin.net/>



# Le projet

système de 9 div dans une div  
les placer avec CSS  
Pour l'aspect dynamique, échanger les class des div pour varier les CSS propre à chaque div  
- voir pour trouver fonction qui compte les click

Rendu : un jeu du Taquin minimal, fonctionnel et esthétique (en mettant en pratique ce vous avez appris sur le niveau 1 avec HTML et CSS) sur une grille 3x3 ou 4x4.

Au minimum : les cases seront numérotées, on pourra à tout moment recommencer la partie grâce à un bouton *Restart*, un compteur de nombre de coups est disponible, une fois l'état final atteint un message de victoire s'affiche et le plateau de jeu se verrouille tant qu'on ne recommence pas une partie.

En bonus : ajouter des images sur chaque case sur le modèle de <https://taquin.net/> qui constituent un puzzle.



# Projet 3 : une calculatrice en TypeScript



# Se former

TypeScript est très similaire à JavaScript et en vous amusant avec, vous allez monter très vite en compétences mais quelques articles pour comprendre à quoi ça sert et comment installer l'environnement de développement ne seront pas de trop. N'ayez pas peur, ce n'est pas vraiment un nouveau langage, simplement une petite surcouche.

Je vous conseille de lire au moins les 3 premiers chapitres de ce guide :

<https://medium.com/@wittydeveloper/typescript-essentials-b7ae85b0f561>

Puis cet article qui explique comment mettre en place un environnement de développement TypeScript basique :

<https://matthewhorne.me/setup-typescript-project/>

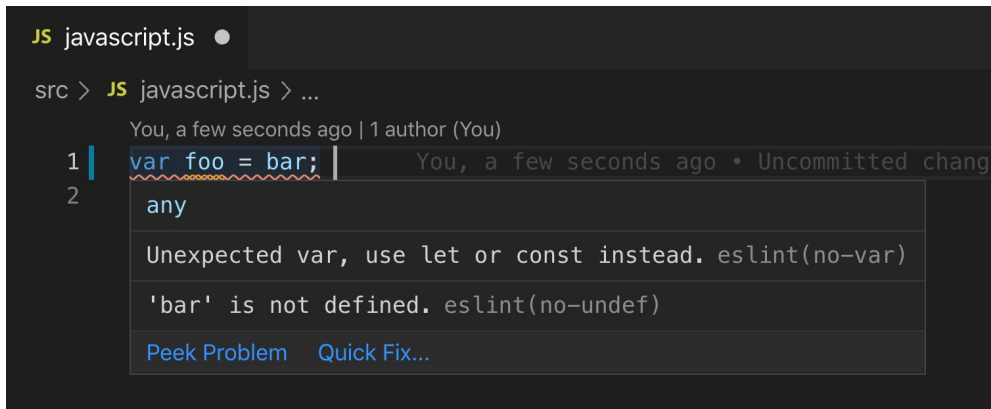
Puisque TypeScript est une surcouche langage qui se compile en JavaScript, il va falloir un environnement capable de compiler votre code `.ts` en `.js`.



# Se former

ESLint est un outil qu'on appelle *linter*. Il vous prévient quand votre code ne respecte pas certaines règles. Ces règles peuvent venir d'un ensemble de règles open-source comme ceux de Google ou Airbnb ou alors être écrites par vous-mêmes. L'intérêt étant d'assurer une *codebase* propre et uniforme.

Il peut aussi se configurer pour reformater automatiquement votre code au moment de la sauvegarde, ce qui est bien pratique.



The screenshot shows a code editor with a file named 'javascript.js'. The code contains the line 'var foo = bar;' on line 1. An ESLint error is displayed below the code, indicating that 'var' is unexpected and 'let' or 'const' should be used instead. The error also notes that 'bar' is not defined. The editor interface includes a line number column on the left, a status bar at the top right showing 'Uncommitted changes', and a bottom panel with 'Peek Problem' and 'Quick Fix...' options.

```
JS javascript.js ●
src > JS javascript.js > ...
You, a few seconds ago | 1 author (You)
1 | var foo = bar; | You, a few seconds ago • Uncommitted chang
2 | any
Unexpected var, use let or const instead. eslint(no-var)
'bar' is not defined. eslint(no-undef)
Peek Problem Quick Fix...
```

# Se former

Pour le configurer, il faut installer la dépendance grâce à NPM (comme pour TypeScript) et configurer votre IDE pour l'intégrer. Pour commencer : <https://eslint.org/docs/user-guide/getting-started> ; La liste des plugins à installer en fonction de votre IDE : <https://eslint.org/docs/user-guide/integrations>.

Vous aurez aussi besoin d'utiliser *typescript-eslint* pour faire fonctionner eslint avec TypeScript, plus d'infos ici :

<https://github.com/typescript-eslint/typescript-eslint/tree/master/packages/eslint-plugin>

Plus d'infos sur eslint en lui-même avec cet article :

<https://www.freecodecamp.org/news/the-essentials-eslint/>

Un bon tutoriel qui vous explique comment tout mettre en place de A à Z :

<https://www.robertcooper.me/using-eslint-and-prettier-in-a-typescript-project>





# Le projet

Rendu : une calculatrice avec les 4 opérations de base —÷×+

Les contraintes seront plus sur le set-up du projet que sur le résultat d'où la simplicité du rendu. Le projet devra être codé en TypeScript et un environnement eslint devra être configuré (règles du plugin eslint-typescript au minimum).

Deadline : 1er décembre 2019