Jour 5

> 💡 Pour la création de dossier/fichiers, faites comme d'habitude!

01 - Calculator

- Créez une fonction `calculate` qui recevra trois paramètres (deux nombres entiers et un opérateur `+`, `-`, `x`, `/` ou `%`)
- Créez une condition (méthode au choix) pour chaque valeur possible de l'opérateur qui retournera le résultat du calcul
- Affichez le résultat dans la console (vérifiez que pour les arguments `5`, `x` et `4` vous obtenez bien `20`)
- Vous devrez donner les arguments dans le terminal en appelant le programme alors attention aux paramètres !

  ⇒ Si vous ne donnez pas trois arguments, le programme doit afficher "error" dans la console

02 - Table

- Créez une fonction `multiply` qui recevra un paramètre (un nombre entier)
- Faites en sorte que la fonction affiche la table de multiplication de ce nombre dans la console (de 1 à 10) ligne par ligne
- Vous devrez donner l'argument dans le terminal en appelant le programme

  ⇒ Si vous ne donnez pas d'argument, le programme doit afficher "error" dans la console

03 - Separate Table

- Créez un fichier `table-utils.js`
- Mettez votre fonction `multiply` dans ce fichier, et ajouter en une deuxieme `addition` qui reprend le même principe mais avec une addition au lieu d'une multiplication
- Exportez les deux fonctions graces aux modules, et importez les dans votre fichier précedant pour refaire marcher votre code
- Affichez la table d'addition

04 - Guess

- Installez et importez le package `prompt` (n'oubliez pas d'initialiser !) et lancez les méthodes de `prompt`
- Créez une variable `mysteryNum` contenant une valeur aléatoire (un nombre entier entre 1 et 100)
- Créez une fonction `play` qui demande au joueur "Quel est le nombre mystère ?" puis :
  - si le joueur donne une valeur incorrecte (pas un nombre) on affiche "error" et on relance la fonction `play`
  - si le joueur donne un chiffre trop petit, on affiche "C'est plus !" et on relance la fonction `play`
  - si le joueur donne un chiffre trop grand, on affiche "C'est moins !" et on relance la fonction `play`

    - si le joueur donne le bon nombre, on affiche "Bravo !!"
- Appelez la fonction
- Testez le jeu

⭐ Bonus

..le mot mystère serait-il "Cassoulet" ?

Vous vous souvenez de **Motus** ? Aujourd'hui on va le coder ! Petit rappel des règles :

- Un mot mystère (ici de 5 lettres) est proposé, on ne connaît que la première lettre
- Le joueur a six tentatives pour deviner le mot mystère, et pour chaque tentative :
    - Les lettres qui sont à la bonne place sont en rouge
    - Les lettres qui sont à la mauvaise place sont en jaune

- Les lettres qui n'existent pas dans le mot mystère restent neutres

⇒ Si le joueur propose un mot qui n'a pas le bon nombre de lettres (ici 5) alors il a perdu

⇒ Si le joueur propose "BOTTE" et que le mot mystère est "BRUTE" attention : un des "T" proposés doit rester neutre !

À l'aide du package `prompt` que vous connaissez, et du package `colors` (cherchez le sur NPM et lisez la doc), tentez d'écrire un code qui peut permettre à l'utilisateur de jouer à Motus. Si vous réussissez : bravo ! Sinon : dommage...