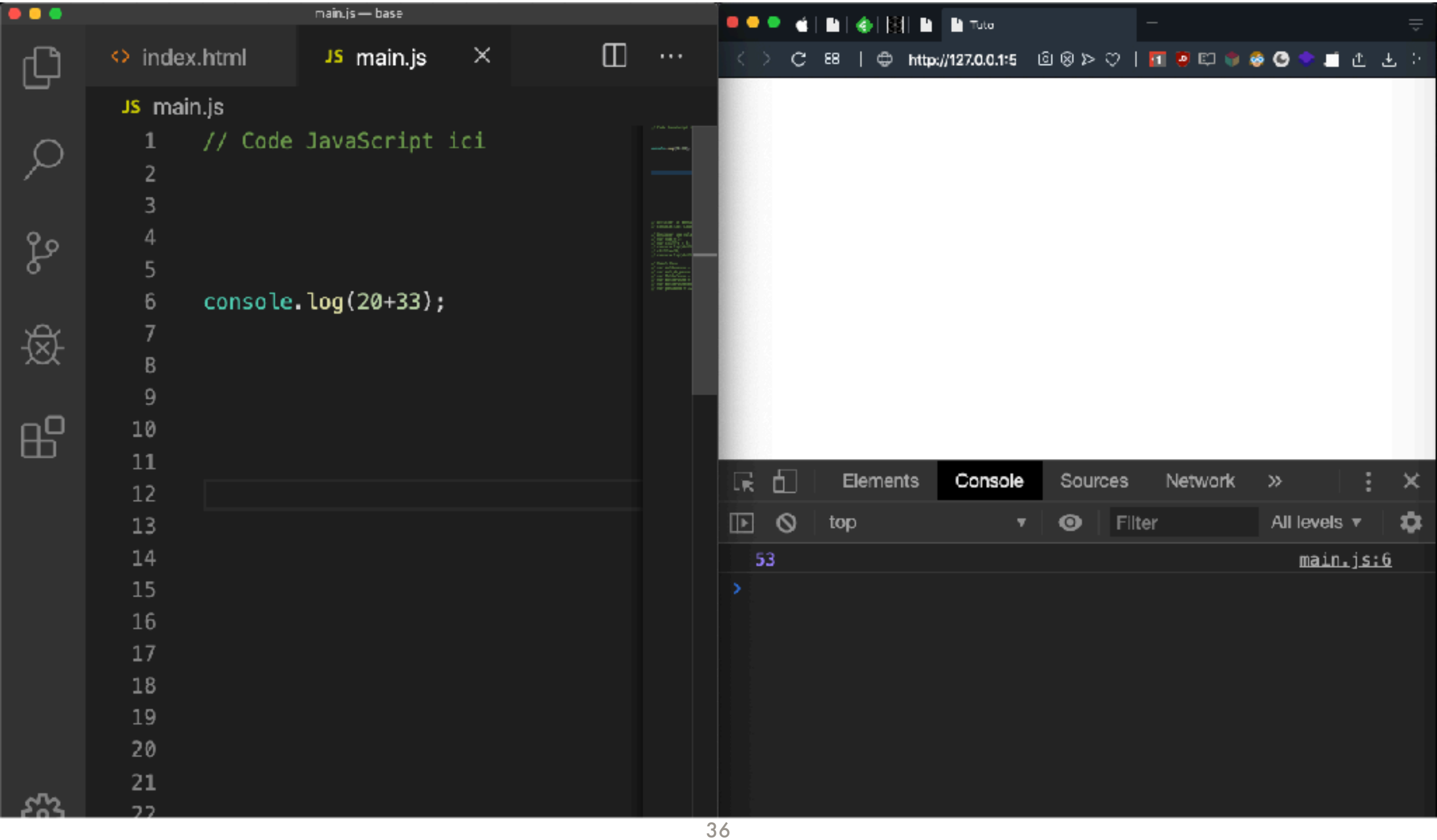
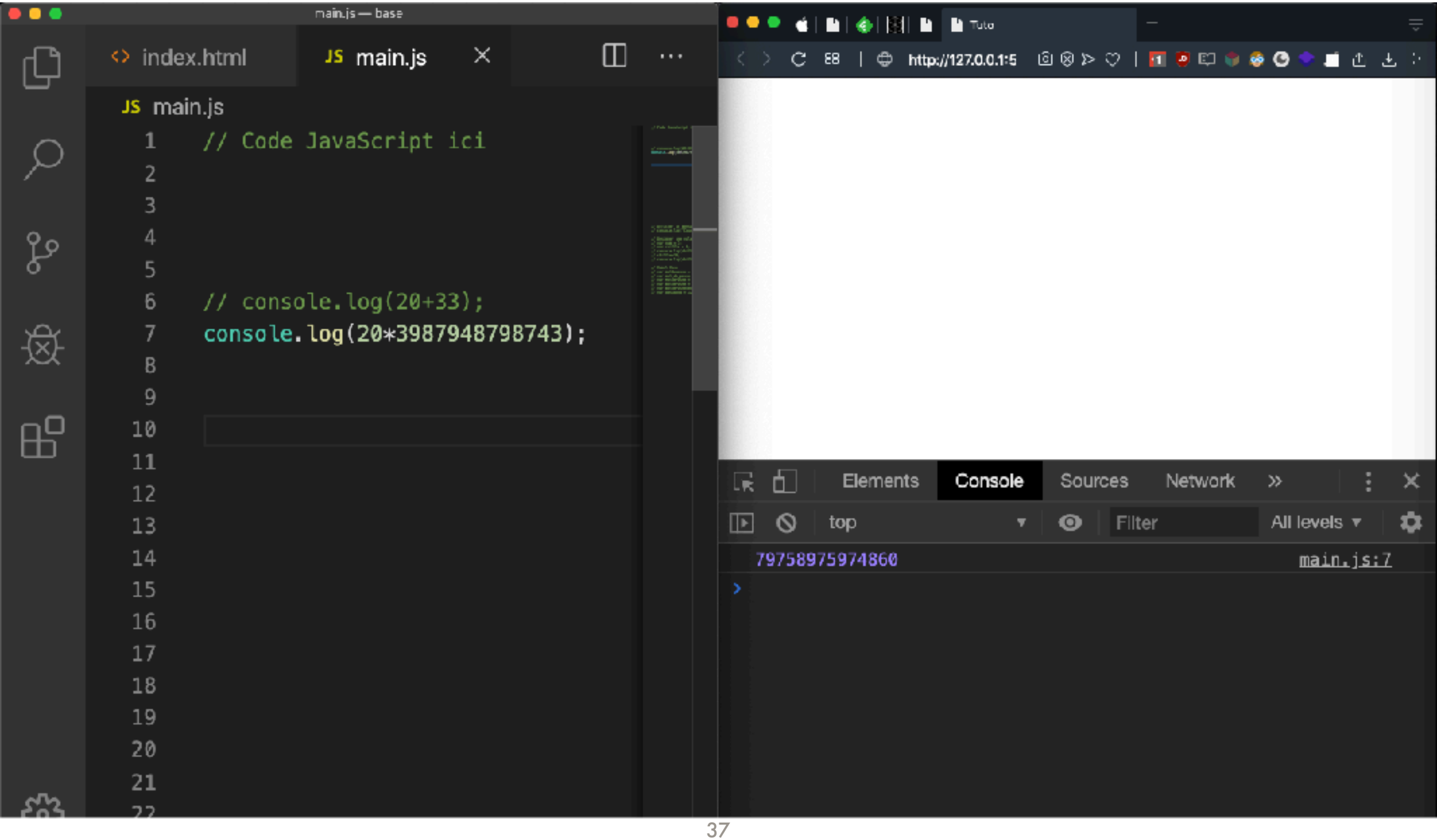
BASE JAVASCRIPT

EXO: CALCUL



BASE JAVASCRIPT

EXO: CALCUL



BASE JAVASCRIPT

EXO: CALCUL

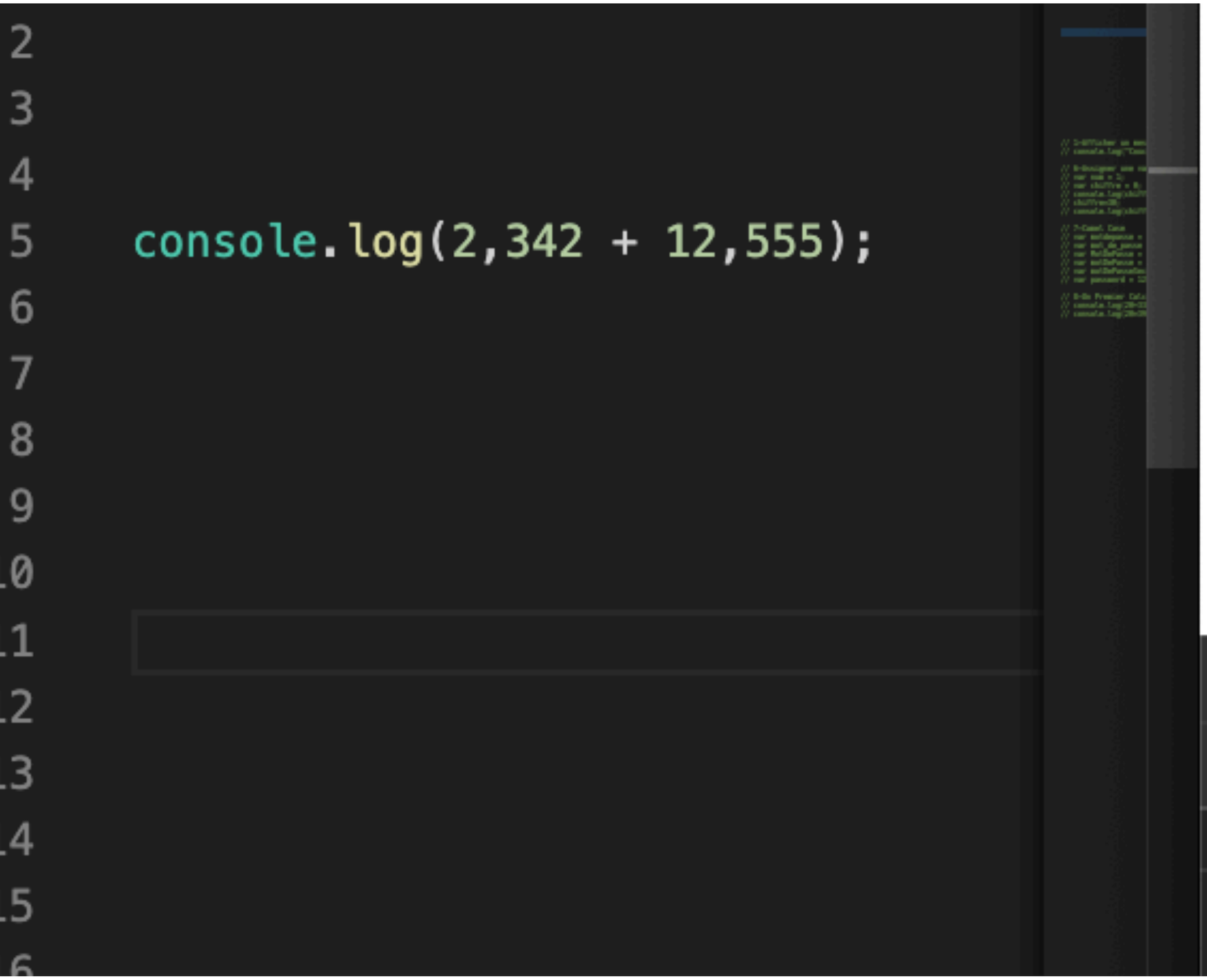
- On a plusieurs opérateurs pour faire des calculs
- + Additionner
- - Soustraire
- * Multiplier
- / Diviser

BASE JAVASCRIPT

NOMBRE DÉCIMAUX

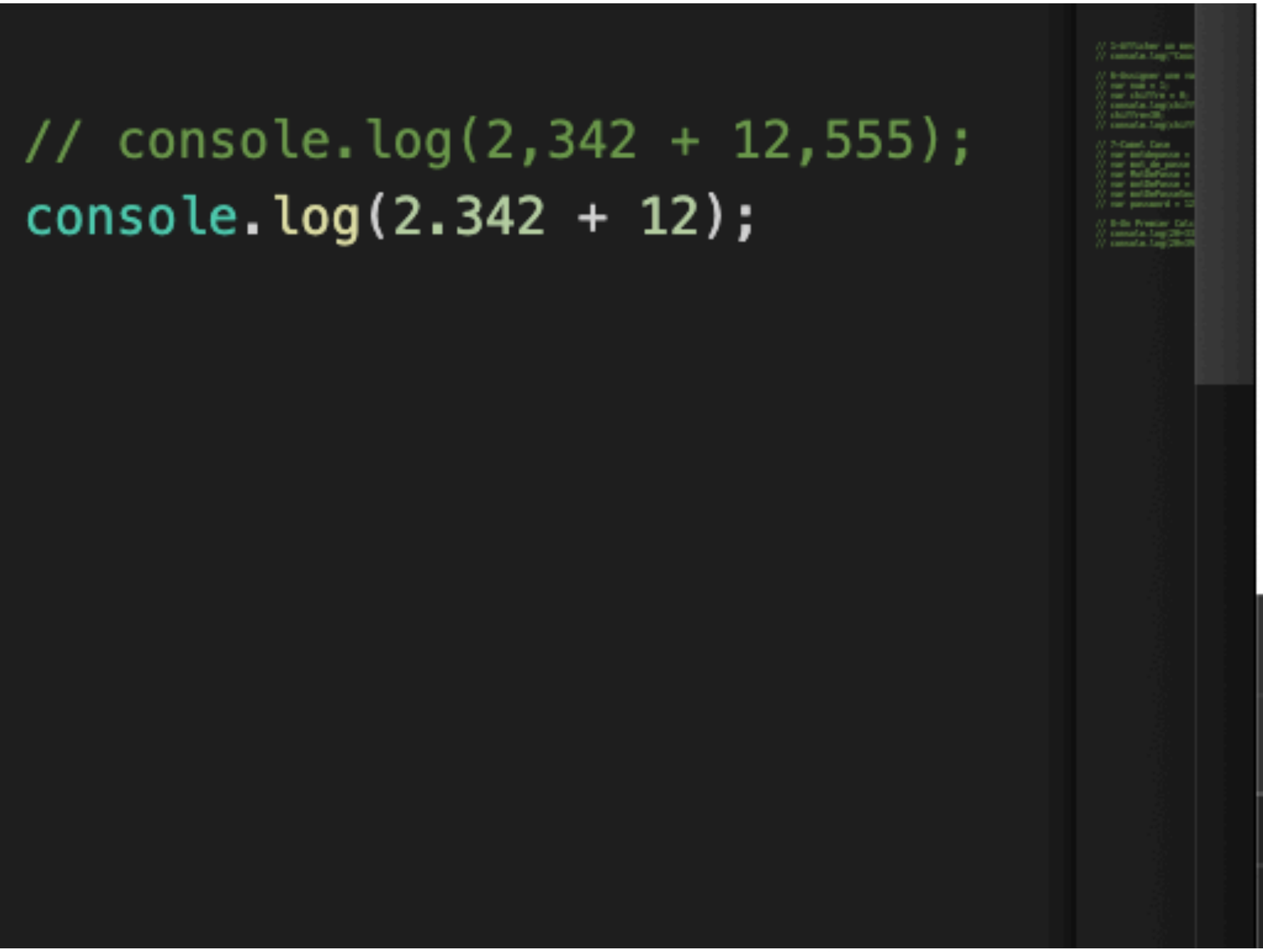
BASE JAVASCRIPT

NOMBRE DÉCIMAUX



BASE JAVASCRIPT

NOMBRE DÉCIMAUX



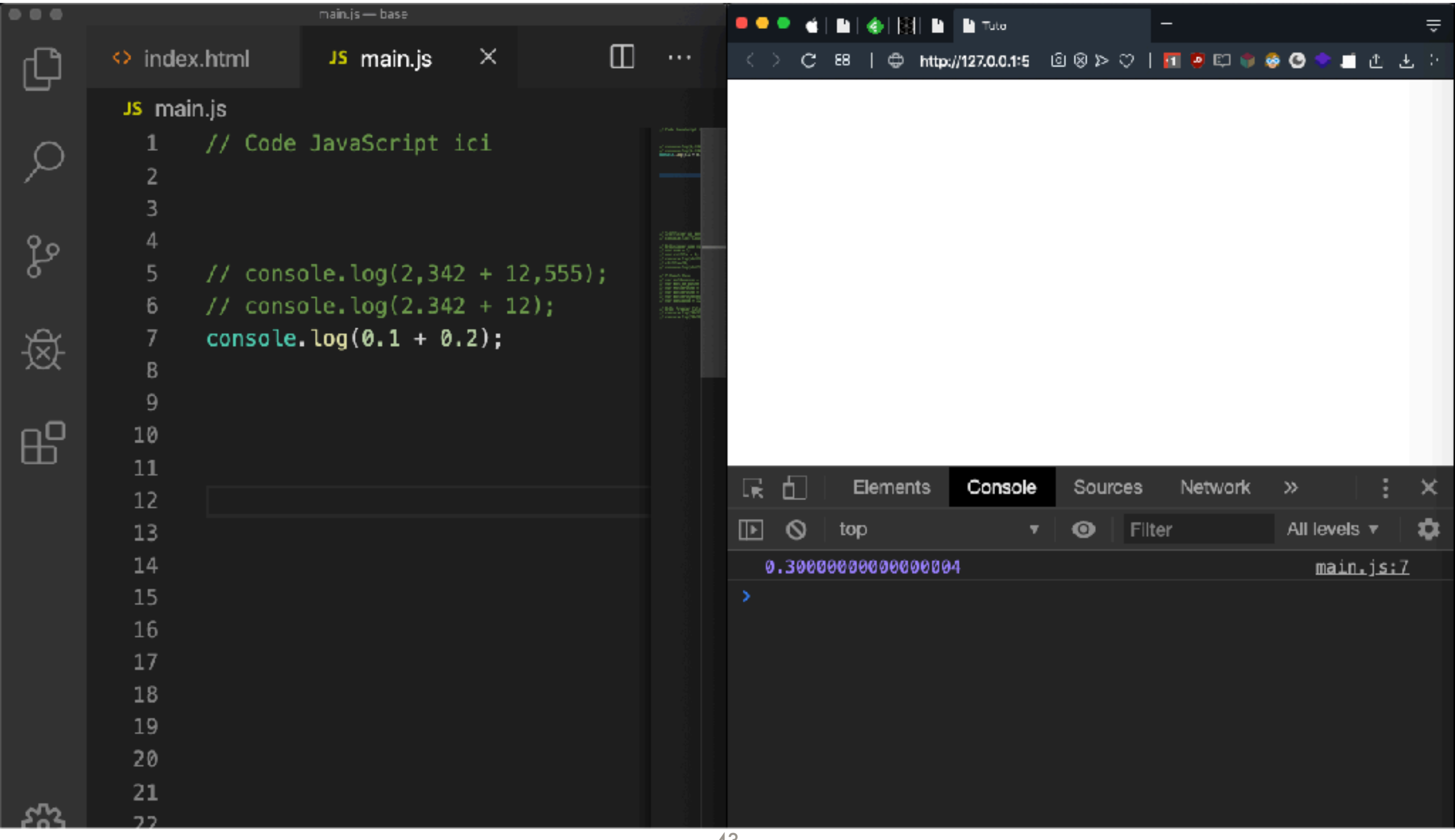
BASE JAVASCRIPT

NOMBRE DÉCIMAUX

- Le « BUG » de JS avec les calculs décimaux.
- À la base quand JS a été créé les capacités mémoire des machines étaient inférieures comparé à aujourd'hui.
- Pour gérer les décimaux les concepteurs du langage on fait le choix d'un certain algorithme de calcul
  (Pour résoudre les Pb de mémoire)
- JS utilise une précision de virgule Flottante
  (un système d'arrondi particulier)
- Pour faire du Jeux vidéo, des Apps Web ou Mobile, des animations cela ne pose pas de soucis.
- Pas pertinent si vous travaillez sur une app scientifique avec des données hyper précises.

BASE JAVASCRIPT

NOMBRE DÉCIMAUX



BASE JAVASCRIPT

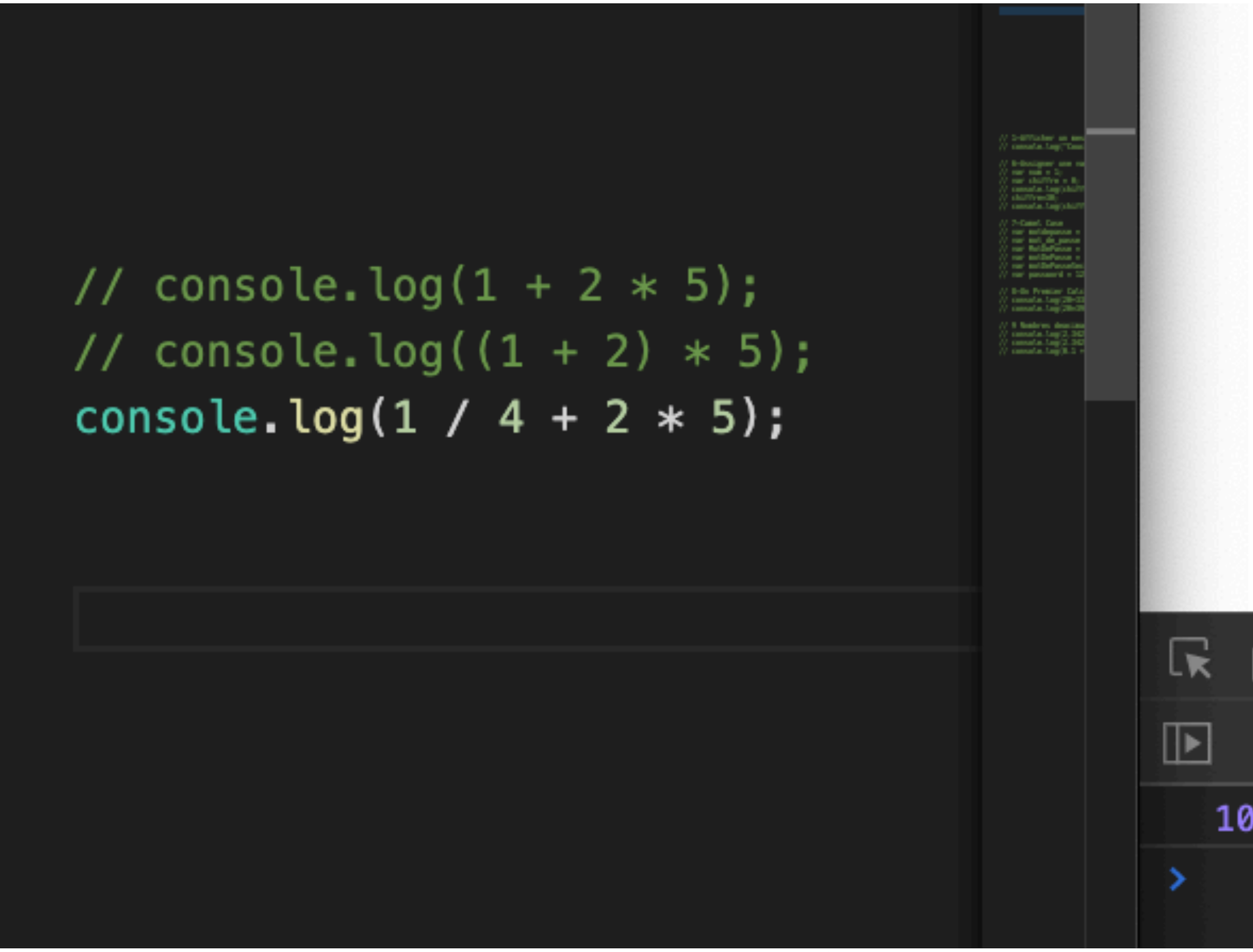
PRIORITÉS DE CALCUL

BASE JAVASCRIPT

PRIORITÉS DE CALCUL

BASE JAVASCRIPT

PRIORITÉS DE CALCUL



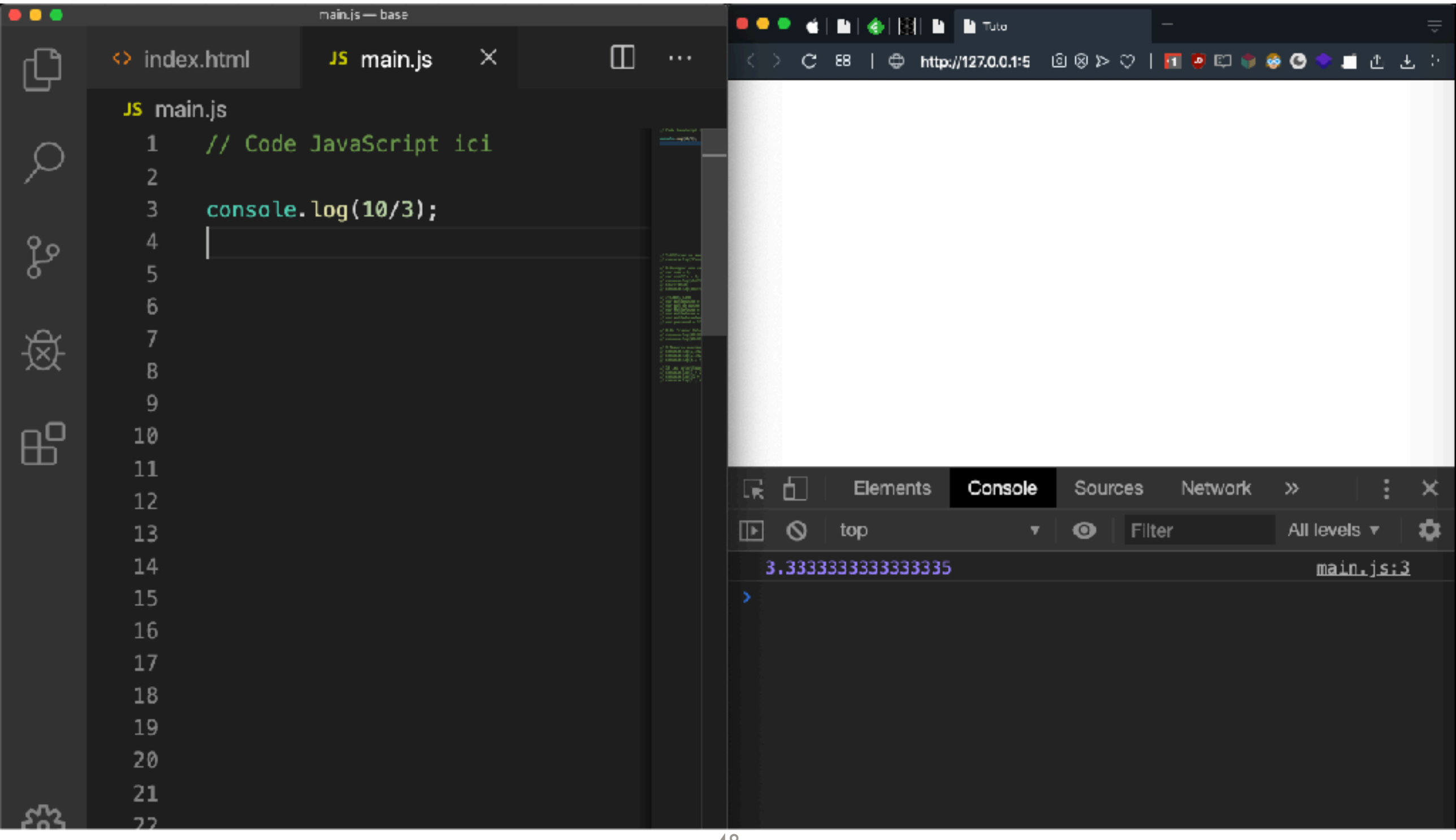
BASE JAVASCRIPT

EXO: OPÉRATION

- On a vu Presque tous les opérateur utiles pour faire des opérations.
- Sauf 1.
- Modulo

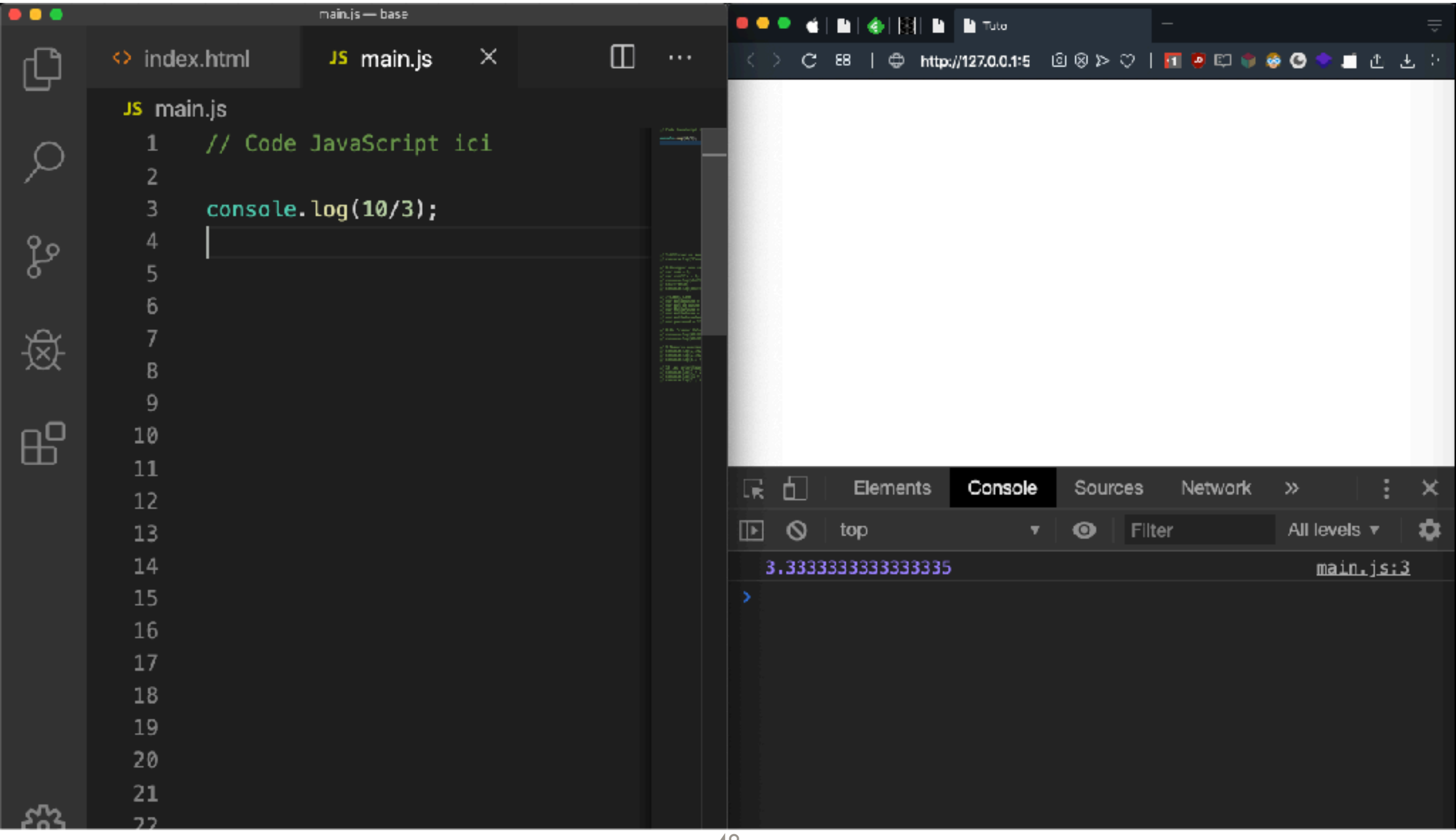
BASE JAVASCRIPT

EXO: OPÉRATION



BASE JAVASCRIPT

EXO: OPÉRATION



BASE JAVASCRIPT

EXO: OPÉRATION

Voici une division posée :

$$\begin{array}{r|l} 35 & 12 \\ \cline{2-2} -\,24 & 2 \\ \cline{1-1} 11 & \end{array}$$

La division euclidienne correspondante va s'écrire :

$$35 = 12 \times 2 + 11$$

BASE JAVASCRIPT

EXO: OPÉRATION