**Résumons !**

* \* Une variable est une **zone de stockage** d'information. Chaque variable possède un nom, une valeur et un type. En JavaScript, le type d'une variable est déduit implicitement de sa valeur, et non pas défini explicitement. Il s'agit d'un langage à **typage dynamique.**
* On déclare une variable JavaScript avec le mot-clé  let  suivi du nom de la variable. Si la valeur initiale d'une variable n'est pas destinée à changer, on utilise de préférence le mot-clé  const  pour déclarer une variable constante.
* L'opérateur d'affectation  =  permet de donner une valeur à une variable. Dans le cas d'une variable de type nombre, on peut utiliser l'opérateur  +=  pour augmenter et l'opérateur  ++  pour incrémenter (augmenter de 1) sa valeur.
* La **portée** (scope) d'une variable représente la portion du code source dans laquelle cette variable est utilisable. Les variables définies avec  let  et  const  ont une portée de type bloc : elles ne sont utilisables qu'à l'intérieur du bloc de code dans lequel elles sont définies. Un bloc de code est délimité par une paire d'accolades ouvrante et fermante.
* Une expression est un morceau de code combinant des variables, des valeurs et des opérateurs. L'évaluation d'une expression produit une valeur et correspond à un certain type.  
  On peut inclure des expressions dans une chaîne de caractères délimitée par une paire d'accents graves seuls (   `...`  ). Ces chaînes sont appelées modèles de libellés (template literals).
* Des conversions de type peuvent avoir lieu implicitement lors de l'évaluation d'une expression, ou bien explicitement avec les instructions  Number()  et  String()  qui convertissement respectivement une expression en un nombre et en une chaîne.
* Les instructions   prompt()  et   alert()  permettent respectivement de faire saisir et d'afficher une information sous la forme d'une boîte de dialogue. Elles ne fonctionnent que dans le contexte d'un navigateur web.
* Il est essentiel de bien nommer ses variables et d'adopter une convention de nommage, comme par exemple camelCase.
* Le mot-clé  if  définit une instruction conditionnelle, appelée également test. Les instructions associées au  if  ne sont exécutées que si la condition est vérifiée (vraie). Une condition est une expression dont l'évaluation produit une valeur booléenne (  true  ou  false  ).
* Les instructions associées à une instruction  if  sont regroupées dans un bloc de code délimité par une paire d'accolades ouvrante et fermante. Pour plus de lisibilité, le contenu d'un bloc de code être indenté (décalé vers la droite) par rapport à l'instruction à laquelle il est associé.
* Les opérateurs  ===  ,   !==  ,   <  ,   <=  ,  >  et   >=  peuvent être utilisés pour comparer des nombres au sein d'une condition. Ils renvoient tous un résultat booléen.
* Associée à un  if  , l'instruction  else  permet d'exprimer une alternative. Selon la valeur de la condition, le bloc de code associé au  if  ou celui associé au  else  sera exécuté, mais jamais les deux. On peut imbriquer sans limite des instructions  if/else   à l'intérieur d'autres instructions   if/else  .
* Les opérateurs logiques  &&  (ET),  ||  (OU) et  !  (NON) permettent de créer des conditions composées.
* L'instruction  switch  permet d'exécuter un bloc de code parmi plusieurs selon la valeur d'une expression.
* On utilise une **boucle** afin d'exécuter plusieurs fois un bloc d'instructions. Chaque exécution est appelée un tour de boucle ou encore une **itération**. Le bloc d'instructions associé à une boucle est appelé le corps de la boucle.
* La boucle  while  permet de répéter des instructions tant qu'une condition est vérifiée. La boucle  for  ajoute la possibilité d'effectuer un traitement à l'entrée dans la boucle (initialisation) et après chaque tour de boucle (étape).
* La variable utilisée dans l'initialisation, la condition et l'étape d'une boucle  for  est appelée le compteur de la boucle.
* Il faut toujours que la condition d'une boucle  while  puisse devenir fausse afin d'éviter le risque d'une boucle infinie.
* On s'interdit de manipuler le compteur d'une boucle  for  à l'intérieur du corps de la boucle.  
  Toutes les boucles peuvent s'écrire avec un  while  . La boucle  for  est à privilégier lorsque le nombre d'itérations est connu à l'avance.