Programmation procédurale

Bases

Structure de base TypeScript console

1. Séquence d'instructions

Exercice: Écrire un programme qui effectue les actions suivantes :

- Déclare trois variables : a, b, et c.
- Assigne les valeurs 5 à a, 10 à b, et la somme de a et b à c.
- Affiche la valeur de c dans la console.

2. Assignation ou affectation

Exercice: Écrire un programme qui fait les opérations suivantes :

- Déclare une variable x avec la valeur 8.
- Multiplie x par 2 et assigne le résultat à une nouvelle variable y.
- Assigne à x la valeur de y divisée par 4.
- Affiche x et y dans la console.

3. Instruction conditionnelle

Exercice 1 :

Écrire un programme qui fait les opérations suivantes :

- Demande à l'utilisateur d'entrer un nombre via prompt().
- Si le nombre est supérieur à 10, affiche “Grand nombre” dans la console.
- Sinon, affiche “Petit nombre”.

Exercice 2 :

Écrivez un programme qui détermine le plus grand de trois nombres et affiche un message approprié selon le cas.

Instructions : Déclarez trois variables x, y, et z et assignez-leur des valeurs numériques. Utilisez des instructions conditionnelles (if, else if, else) pour déterminer lequel des trois nombres est le plus grand. Affichez le résultat avec un message, par exemple : “Le plus grand nombre est x.”

4. La boucle

Exercice 1 :

Écrire un programme qui affiche les nombres de 1 à 5 en utilisant une boucle for.

Exercice 2 :

Écrivez un programme qui génère les 10 premiers nombres de la séquence de Fibonacci et les affiche dans la console.

Instructions :

1. Utilisez une boucle for pour itérer 10 fois.
2. Initialisez deux variables a et b avec les valeurs 0 et 1.
3. À chaque itération, affichez la valeur actuelle de a.
4. Mettez à jour les valeurs de a et b pour obtenir le prochain nombre de Fibonacci.

Exercice 3 :

Écrivez un programme qui demande à l'utilisateur de deviner un nombre secret entre 1 et 10. L'utilisateur a un nombre limité de tentatives (par exemple, 3). Le programme doit indiquer si la réponse est correcte ou non après chaque tentative, et arrêter si le nombre est trouvé ou si les tentatives sont épuisées.

Instructions :

1. Déclarez une variable pour le nombre secret (par exemple const secret = 7).
2. Utilisez une boucle while pour limiter le nombre de tentatives (par exemple, 3).
3. Utilisez un branchement (if) pour vérifier si la réponse de l'utilisateur est correcte ou non.
4. Affichez un message après chaque tentative indiquant si le nombre est correct ou incorrect.
5. Si le nombre est trouvé, sortez de la boucle.

5. Les branchements

Exercice 1 :

Écrire un programme qui fait les opérations suivantes :

- Déclare un tableau numbers contenant les nombres 1, 2, 3, 4, 5.
- Parcourt le tableau à l'aide d'une boucle.
- Si le nombre est 3, sort de la boucle immédiatement.
- Sinon, affiche le nombre dans la console.

Exercice 2 :

Écrivez un programme qui fonctionne comme une calculatrice simple. Il doit demander à l'utilisateur deux nombres, puis l'opération à effectuer (addition, soustraction, multiplication, division) et afficher le résultat. Le programme doit continuer jusqu'à ce que l'utilisateur choisisse de quitter.

Instructions :

1. Utilisez une boucle while pour que le programme continue à fonctionner jusqu'à ce que l'utilisateur saisisse une commande pour quitter (par exemple “q”).
2. Demandez à l'utilisateur d'entrer deux nombres et l'opération à effectuer.
3. Utilisez des conditions pour effectuer l'opération demandée.
4. Affichez le résultat.

5. Gérez les erreurs (par exemple, division par zéro).

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/cnam/utc503/td0

Last update: **2026/08/24 14:26**