

Brief :20 exercices pratiques

let's start coding !

Exercice 1 : Inverser les valeurs de deux variables

Exercice 2 : Afficher des variables

A partir des variables suivantes :

Nom = toto;

Age = 30;

Homme = true;

Exercice 3 : Puissance de 2

Réaliser un programme permettant à l'utilisateur de saisir la puissance de 2 qu'il souhaite afficher

Exercice 3:

Créez une fonction qui prend deux nombres comme arguments et retourne leur somme.

Exemple:

addition(1, 2) → 3

addition(-2, -4) → -6

Exercice 3:

Écrivez une fonction qui prend un nombre entier de minutes et le convertit en secondes.

Exemple:

minuteToSecond(4) → 240

minuteToSecond(3) → 180

Exercice 4:

Créez une fonction qui prend un nombre en argument, incrémente le nombre +1 et renvoie le résultat.

Exemple:

`increment(1) → 2`

`increment(4) → 5`

Exercice 5:

Écrivez une fonction qui prend la base et la hauteur d'un triangle et retourne sa surface. Notez que la surface d'un triangle est: $(base * hauteur) / 2$

Exemple:

`getSurface(8, 2) → 8`

`getSurface(7, 3) → 10.5`

Exercice 6:

Écrivez un programme JavaScript pour inverser une chaîne de caractères.

Exemple:

`strReverse('WayToLearnX') → XnraeLoTyaW`

`strReverse('Hello') → olleH`

Exercice 7:

Écrivez un programme JavaScript pour trouver le plus grand des trois entiers donnés.

Exemple:

`getMax(5, 9, 1) → 9`

`getMax(2, 3, 10) → 10`

Exercice 8:

Créez une fonction qui prend un tableau et renvoie le premier élément. Notez que le premier élément d'un tableau commence toujours par l'index 0.

Exemple:

`getFirst([1, 2, 3]) → 1`

`getFirst([50, 60, 70]) → 50`

Exercice 9:

Écrivez un programme JavaScript pour récupérer l'URL d'un site Web

Exercice 10:

Écrivez un programme JavaScript pour renvoyer le reste de deux nombres. Il existe un seul opérateur en JavaScript, capable de fournir le reste d'une division. Deux nombres sont transmis comme paramètres. Le premier paramètre divisé par le deuxième paramètre.

Exemple:

`resteDiv(1, 3) → 1`

`resteDiv(2, 4) → 2`

`resteDiv(3, 3) → 0`

Exercice 11: < 100

Avec deux nombres, retournez TRUE si la somme des deux nombres est inférieure à 100. Sinon, retournez FALSE.

Exemple:

`check(5, 20) → true`

`// 5 + 20 = 25`

`check(60, 55) → false`

`// 60 + 55 = 115`

Exercice 12:

Écrivez une fonction qui convertit les heures en secondes. Notez qu'il y a 60 secondes en une minute et 60 minutes en une heure.

Exemple:

`heureSeconde(1) → 3600`

`heureSeconde(5) → 18000`

Exercice 13:

Créez une fonction qui prend un nombre et renvoie TRUE s'il est inférieur ou égal à zéro, sinon retournez FALSE.

Exemple:

`checkNbr(1) → false`

`checkNbr(-1) → true`

`checkNbr(0) → true`

Exercice 14:

Créez une fonction qui renvoie TRUE lorsque `nbr1` est égal à `nbr2`; sinon retournez FALSE.

Exemple:

`isEqual(3, 6) → false`

`isEqual(1, '1') → false`

`isEqual(1, 1) → true`

Exercice 15:

Créez une fonction qui renvoie TRUE si un entier est divisible par 5, sinon retournez FALSE.

Exemple:

`isDivisible(5) → true`

`isDivisible(10) → true`

`isDivisible(6) → false`

Exercice 16:

Écrivez une fonction qui prend deux entiers (heures, minutes) et les convertit en secondes.

Exemple:

`hmTos(1, 10) → 4200`

`hmTos(0, 59) → 3540`

`hmTos(0, 0) → 0`

Exercice 17:

Écrivez une fonction pour inverser un tableau. Attention: N'utilisez pas la méthode intégrée `reverse()`;

Exemple:

`reverseTab([1, 2, 3]) → [3, 2, 1]`

`reverseTab([1, 1, 2, 2, 3]) → [3, 2, 2, 1, 1]`

Exercice 18:

Créez une fonction qui prend un tableau et renvoie le dernier élément du tableau.

Exemple:

```
getLastElem([1, 2, 3, 4]) → 4
```

```
getLastElem(['Alex', 'Bob', 'Emily']) → 'Emily'
```

Exercice 19:

Créez une fonction qui vérifie l'égalité de deux paramètres a et b. La valeur et le type de paramètres doivent être testés afin d'avoir une égalité. La fonction doit renvoyer TRUE si les paramètres sont égaux et FALSE s'ils sont différents.

Exemple:

```
checkEq(5, 5) → true
```

```
// number = number: leur type et leur valeur sont égaux
```

```
checkEq(5, 0) → false
```

```
// number = number: leur type sont égaux et leur valeur sont différent
```

```
checkEq(5, false) → false
```

```
// number <> boolean: leur type est différent
```

```
checkEq(5, '5') → false
```

```
// number <> string: leur type est différent
```

Exercice 20:

Créez une fonction qui renvoie TRUE si une chaîne est vide et sinon renvoie FALSE.

Exemple:

`isEmpty("1") → false`

`isEmpty("WayToLearnX") → false`

`isEmpty(" ") → false`

`isEmpty("") → true`