

TP et labo de la semaine 5 + solutions

1) Nous sommes intéressés à vérifier à quel point la fonction `Math.random()` donne des nombres aléatoires. Pour cela, nous voulons simuler une série de tirs à pile-ou-face. Nous voulons compter le nombre de fois qu'il faut tirer pour que 5000 des tirs aient donné pile. En principe le nombre de tirs devrait être proche de 10000 si `Math.random` est aléatoire. Codez un programme qui effectue cette simulation dans un premier temps avec une boucle `while`, et ensuite avec une boucle `for`.

Note : pour tirer à pile-ou-face on peut se servir du test `Math.random() < 0.5`. Un résultat `"true"` signifie "pile", et un résultat `"false"` signifie "face".

Solution :

```
var pile = 0; // Nombre de fois qu'on tombe sur pile
var total = 0; // Nombre de tirs de pièce

while(pile < 5000) {

    if(Math.random() < 0.5) {
        pile++;
    }

    total++;
}

print("Total de tirs pour avoir 5000 'pile' : ", total);
```

2) Codez un programme qui, étant donné une variable `n` qui contient un entier ≥ 0 , calcule et affiche la somme des `n` premiers nombres impairs (par exemple, pour `n=4` : $1 + 3 + 5 + 7$). Essayez d'exécuter votre programme avec codeBoot avec différentes valeurs de `n`. En mots expliquez la relation entre la somme obtenue et la valeur de `n`. Trouvez une optimisation (meilleur codage) de votre programme pour qu'il calcule la même valeur en moins de pas d'exécution.

Solution :

```
var n = 4;
var somme = 0;

for(var i=0; i < n; i++) {
    somme += i * 2 + 1;
}

print(somme);
```

Autre façon de procéder :

```
var n = 5;
var somme = 0;
```

```

    for(var i=0; i < 2 * n; i += 2) {
        somme += i + 1;
    }

    print(somme);

```

Le programme nous donne les valeurs suivantes :

```

Si n = 4 => 16
Si n = 5 => 25
Si n = 6 => 36
Si n = 7 => 49
Si n = 8 => 64
Si n = 9 => 81
...

```

Le programme calcule donc n^2 .

Le programme suivant est donc une version simplifiée du même programme :

```

var n = 5;
    var resultat = Math.pow(n, 2);

    print(resultat);

```

3) Dans un programme nous voulons obtenir de l'utilisateur un nombre positif. Si l'utilisateur entre un nombre invalide (≤ 0 ou bien du texte non-numérique) nous voulons continuer à lui demander un nombre positif jusqu'à ce qu'il entre un nombre positif. Codez cette partie de programme.

Solution :

On peut procéder avec un while :

```

var input = prompt("Veuillez entrer un nombre positif");

    while(input == "" || isNaN(input) || input < 0) {
        input = prompt("Veuillez entrer un nombre positif");
    }

    print("Vous avez entré le nombre ", input);

```

Mais un do...while permet d'éliminer le prompt redondant :

```

var input = 0;

    do {

        input = prompt("Veuillez entrer un nombre positif");
    }

```

```

    } while(input == "" || isNaN(input) || input < 0);

    print("Vous avez entré le nombre ", input);

```

Une troisième façon de faire est d'initialiser la variable `input` à une valeur qui forcera à entrer dans le `while` :

```

var input = "";

while(input == "" || isNaN(input) || input < 0) {
    input = prompt("Veuillez entrer un nombre positif");
}

print("Vous avez entré le nombre ", input);

```

4) Utilisez la réponse de l'exercice #3 pour coder un programme qui affichera la somme de tous les nombres positifs qui ont été entrés par l'utilisateur. L'utilisateur indiquera qu'il a fini d'entrer des nombres en entrant le nombre 0. Le programme doit indiquer l'index du nombre qui est demandé (i.e. le premier, le second, etc) avec le message "Entrez le nombre positif #1", "Entrez le nombre positif #2", etc

Solution :

```

var input = "";

var somme = 0;
var index = 1;

do {
    input = prompt("Entrez le nombre positif #" + index);

    if(!isNaN(input) && input > 0) {
        somme += +input;
        index++;
    }
} while(input != 0);

print("La somme des nombres positifs entrés est ", somme);

```

5) Modifiez le programme de l'exercice #4 pour qu'il affiche la moyenne des nombres positifs entrés par l'utilisateur.

Solution :

```

var input = "";

```

```

var somme = 0;
var index = 1;

do {
    input = prompt("Entrez le nombre positif #" + index);

    if(!isNaN(input) && input > 0) {
        somme += +input;
        index++;
    }
} while(input != 0);

print("La moyenne des nombres positifs entrés est ",
somme/(index - 1));

```

6) Affichez un damier de taille N x N. Expliquez d'abord votre algorithme en mots, puis exprimez-le en JavaScript. (Source: Eloquent JavaScript)

Un damier 8x8

```

# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #

```

Solution:

```

for(var i=0; i<n; i++) {
    var str = "";
    for(var j=0; j<n; j++) {
        if((i+j) % 2 == 0)
            str += '#';
        else
            str += ' ';
    }
    print(str);
}

```