

Une erreur est survenue lors du chargement de la version complète de ce site. Veuillez vider le cache de votre navigateur et rafraîchir cette page pour corriger cette erreur.

Apprendre le Lua : Session #2

iThorgrim

Apprendre le Lua : Session #2

Qu'est ce qu'une condition ? Pseudo-Code ? If ? C'est quoi un case ?

Le principe de ses Session est de pouvoir apprendre le Lua directement depuis Open-Wow.eu.

Au cours de cette Session nous verrons donc les bases des sujets suivants :

- Le Pseudo-Code
 - Apprendre à coder dans sa tête
- Qu'est ce qu'une structure conditionnelle ?
 - If
 - Elseif
 - Else
- Qu'est ce qu'un Case ?
 - Créer son propre Case

Pseudo-Code

Avant de commencer à parler de structure conditionnelle et donc de condition, je pense qu'il est important de comprendre la base de certaine chose.

Alors nous ne parlerons pas d'Algorithmique etc. mais nous allons parler de Pseudo-Code.

Nous sommes Français, normalement si vous lisez ceci c'est que votre langue natale est celle de notre beau pays ou que tout du moins vous comprenez et parlez Français.

Pour tout Français qui le parle nativement, réfléchir en Français est donc plus simple, plus rapide, plus intuitif.

Le Pseudo-Code c'est une manière de faire un brouillon de votre code en Français.

Vous utilisez déjà du Pseudo-Code tout les jours (ainsi que des bases en Algorithmique).

Lorsque que vous suivez une recette de cuisine vous suivez des instructions (qui sont l'algorithme), vous les suivez dans l'ordre et des conditions sont posés.

Exemple :

- Remplir une casserole d'eau
- Placer votre casserole sur votre gazinière pendant 10 minutes à feu doux.
- Si l'eau bouille, verser 100 grammes de pâtes.
- Laissez les pâtes cuir pendant 10 minutes supplémentaire.
- Si les pates vous plaise au bout de 10 minutes, retirez les.
 - Sinon laissez encore 2 à 3 minutes selon vos goûts.

Vous pouvez déjà repérer quel sont les mots clés utilisés en Pseudo-Code :

- SI
- SINONSI
- SINON
- FINSI

Apprendre à coder dans sa tête

Admettons demain vous souhaitez développer un robot qui à comme fonction de savoir si un chiffre est plus grand qu'un autre, votre Pseudo-Code ressembleras alors à cela :

```
monPremierChiffre <- 1 -- Je déclare ma variable monPremierChiffre et lui attribut la valeur 1
monSecondChiffre <- 2 -- Je déclare ma variable monSecondChiffre et lui attribut la valeur 2

-- Je met en place une condition, afin de comparer mes deux variables

-- Si ma condition est vrai ..
SI monPremierChiffre > monSecondChiffre ALORS

    -- .. alors j'envoie un texte qui confirme ma condition
    ECRIRE ('Mon premier chiffre est plus grand que mon second chiffre')

-- Sinon, ma condition est donc fausse ..
SINON

    -- .. alors j'envoie un texte, qui infirme ma condition
    ECRIRE ('Mon premier chiffre est plus petit que mon second chiffre')
FINSI
```

Vous avez donc construit une suite d'instruction qui forme un algorithme et avais posés la base d'une structure conditionnelle, certes en Pseudo-Code mais c'est déjà un bon début.

Si vous souhaitez en savoir plus sur le Pseudo-Code je vous conseil ceci : [Article Wikipédia très complet](#)

Si vous souhaitez en savoir plus sur l'algorithme je vous conseil ceci : [Cours enseignée à l'université de Paris 7](#)

Qu'est ce qu'une structure conditionnelle ?

Une structure conditionnelle est bloc de code qui vous permet de tester votre condition, savoir si elle est vrai ou fausse.

En Lua une structure conditionnelle contient toujours des instructions conditionnelle.

Elles sont au nombre de 3 :

- if ... then
- elseif .. then
- else

En Pseudo-Code cela ressemblerais à ceci :

- SI ... ALORS
- SINONSI ... ALORS
- SINON

Voici donc à quoi ressemble un bloc de code qui contiendrais une structure conditionnelle :

(Je reprend l'exemple du robot qui doit comparer deux chiffres)

```
local monPremierChiffre = 1
local monSecondChiffre = 2

if monPremierChiffre > monSecondChiffre then
    print ('Mon premier chiffre est plus grand que mon second chiffre')
else
    print ('Mon premier chiffre est plus petit que mon second chiffre')
end
```

Voyons un peu le cas du "if", ça "fonction" est très simple.

Cette instruction conditionnelle permet simplement de laisser passer la condition si elle est vraie.

```
if 1 < 2 then
  print('La condition est vraie')
end
```

Le "if" ne laisseras donc jamais passer une condition qui est fausse. Cela parait logique c'est la base même d'une condition.

elseif

Tout comme le "if" le "elseif" va lui aussi attendre que sa condition soit vraie.

Il est utilisés lorsque que plusieurs cas son possible au sein d'une même condition.

Vous pourriez être tenter de me dire : "Oui mais si j'ai plusieurs cas possible pourquoi ne pas faire plusieurs bloc d'instruction if ?".

Tout simplement car le "elseif" permet de mettre fin à la condition lorsque est vraie.

Si vous mettez plusieurs "if "à la suite comme ceci :

```
if 1 < 2 then
  print('La condition 1 est vraie')
end
if 1 > 2 then
  print('La condition 2 est fausse')
end
if 2 < 3 then
  print('La condition 3 est vraie')
end
if 2 > 3 then
  print('La condition 4 est fausse')
end
```

Votre programme va tout simplement tester la totalités de votre structure conditionnelle, qu'importe si la condition précédente est **vrai**. Il ne s'arrêtera qu'a la fin de sa lecture.

De plus votre programme vous enverra un message lorsque votre condition est **vrai**, grâce à la fonction `print` , mais il vous enverras aussi un message toutes les fois ou la condition est **fausse** .

Ainsi au lieu de recevoir 1 message lorsque votre condition est **vrai**, vous recevrez 1 message lorsque votre condition est **vrai** mais 2 messages de plus lorsque votre condition est **fausse** .

Or avec "elseif" la condition s'arrête dés que celle-ci est **vrai**.

```
if 1 < 2 then
  print('La condition 1 est vraie')
elseif 1 > 2 then
  print('La condition 2 est fausse')
elseif 2 < 3 then
  print('La condition 3 est vraie')
elseif 2 > 3 then
  print('La condition 4 est fausse')
end
```

Une fois qu'une des conditions est **vrai** alors le programme s'arrête et / ou exécute le code mis en place dans la condition.

else

Cette instruction est très utile au cas où vous voudriez exécuter une autre série d'instruction si votre condition n'est pas remplie, qu'elle est donc **fausse**.

```
if 1 < 2 then
    print('La condition 1 est vraie')
else
    print('La condition 1 est fausse')
end
```

Qu'est ce qu'un Switch

Le Switch est une structure conditionnelle différente du cas "if // else if // else", il permet d'écrire un ensemble de condition sous une autre forme.

Notre Switch est introduit par le mot-clé que nous allons définir, qui reçoit en argument la variable à tester, puis, entre crochets, les clés reçoivent les différentes valeurs attendues pour cette variable.

La manière de créer un Switch en Lua est assez spécifique au Lua bien qu'elle ressemble beaucoup à la manière de faire des autres langages de développement.

Le Switch n'est pas natif au Lua, il nous faut alors bidouiller et pour ça rien de mieux que les tableaux.

Dans notre exemple nous allons attribuer la valeur "Fraise" à notre variable du nom de "fruits".

```
local fruits= "Fraise"

local Switch = {
    ["Banane"] = function()
        io.write('Je mange une banane')
    end,

    ["Fraise"] = function()
        io.write('Je mange une fraise')
    end,

    ["Pomme"] = function()
        io.write('Je mange une pomme')
    end
}

Switch[fruits]()

-- Resultat : Je mange une fraise
```

J'espère que ce guide vous aura aidé à en apprendre un peu plus sur Lua, et j'espère vous avoir aidé.

noc

 **iThorgrim** Le Switch ou Case est une structure conditionnelle très utilisée lorsque que votre condition est très

Magnifique tutoriel à qui il manque la fin, la suite au prochain numéro?

iThorgrim

 **noc**

Où il faut que je le termine ☐ ☐

iThorgrim

Petit bump le guide est terminés !
