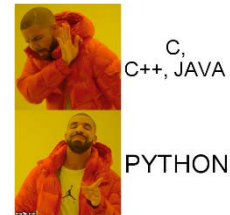


## Love Python

### Contexte

Un étudiant TS pense avoir trouvé LA solution pour programmer le jeu Lizord du labo 1 sans qu'on puisse exploiter de vulnérabilité.

Ayant suivi avec assiduité le cours SLO, il en a conclu que le langage C et le Java provoquait trop d'erreur. Il se souvient notamment de ces labos d'inf1 où il oubliait toujours de mettre ses breaks dans ses switches.



Malin, Il choisit alors de coder en python, langage qu'il a découvert lors du cours de Cryptographie (CRY). Il estime que celui-ci protège facilement et efficacement des erreurs de programmation.

#### Infrastructure :

Pour des raisons techniques, le script python n'a pas été mis sur un serveur. Néanmoins, votre ami vous a déclaré qu'il n'utilisait que du **python2** sur son incroyable serveur car il n'a pas écouté son enseignant de CRY quand il lui avait dit d'installer python 3.

**Prouvez-lui que son implémentation est mauvaise en gagnant 10 fois de suite.**

### Code

Le code du programme est disponible à la page suivante.

```
# -*- coding: utf-8 -*-
# Auteur : Ruckstuhl Michael et Sauge Ryan
#Ce programme est fait pour fonctionner avec la version python 2.7

import random

tuple = ["Lizard", "Spock", "Paper", "Scissors", "Rock"]
score = 0
stopBreak = 0

def checkLiz(srvString):
    if srvString == "Spock" or srvString == "Paper":
        return "Win"
    elif srvString == "Lizard":
        return "Draw"
    else:
        return "Lose"

def checkSpock(srvString):
    if srvString == "Scissors" or srvString == "Rock":
        return "Win"
    elif srvString == "Spock":
        return "Draw"
    else:
        return "Lose"

def checkPaper(srvString):
    if srvString == "Spock" or srvString == "Rock":
        return "Win"
    elif srvString == "Paper":
        return "Draw"
    else:
        return "Lose"

def checkSciss(srvString):
    if srvString == "Lizard" or srvString == "Paper":
        return "Win"
    elif srvString == "Scissors":
        return "Draw"
    else:
        return "Lose"

def checkRock(srvString):
    if srvString == "Lizard" or srvString == "Scissors":
        return "Win"
    elif srvString == "Rock":
        return "Draw"
    else:
        return "Lose"

#Fonction appelée si l'utilisateur gagne
def Win():
    global score
    score = score + 1
```

```
# Source switch : https://jaxenter.com/implement-switch-case-statement-python-138315.html
#Gère le choix de l'utilisateur
def switch(arg):
    if (arg > 4):
        return "Lost"
    srvValue = random.randint(0, 4)
    srvString = tuple[srvValue]
    switcher = {
        0: checkLiz(srvString),
        1: checkSpock(srvString),
        2: checkPaper(srvString),
        3: checkSciss(srvString),
        4: checkRock(srvString),
    }
    # Get the function from switcher dictionary
    fun = switcher.get(arg, lambda: "NON")
    if fun == "Win":
        Win()
    return fun

def main():
    global score
    MAX = 10
    count = 0
    print("Choix possible : Lizard(0), Spock(1), Paper(2), Scissors(3),
Rock(4)")
    print("Chaque choix est représenté par un chiffre")
    while (True):
        try:
            arg = input("Entrez votre valeur : ")
        except:
            print("Ce n'est pas permis")
            continue
        if isinstance(arg, (int, float)):
            res = switch(int(arg))
            print(res)
        if score >= MAX:
            print("Bravo, tu as gagné " + str(MAX) + " fois de suite")
            count = count + 1
        if count == MAX:
            score = 0
            count = 0

if __name__ == "__main__":
    # execute only if run as a script
    main()
```