Introduction à l'algorithmique et à la programmation en python TP2

Objectif(s)

Introduction aux instructions conditionnelles

La syntaxe est

|  |  |
| --- | --- |
| if *condition1*:  *Instruction1*  else:  *Instruction2* | if *condition1*:  *Instruction1*  elif *condition2*:  *Instruction2*  elif *condition3:*  *Instruction3*  Else:  *Instruction4* |

•

Introduction aux exceptions

•

Exercice 1 – Premières instructions conditionnelles

1.

a) Que fait le programme suivant ?

z= input("z?"))

if (z>0):

print "entier strictement positif"

b) Modifier ce programme pour qu'il affiche "entier négatif" dans le cas contraire.

c) Modifier ce programme pour qu'il affiche "entier négatif" si z est un entier strictement négatif et qu'il affiche "Cet entier est nul." si l'entier vaut 0. Tester avec les valeurs 12, -23 et 0.

2.

Que se passe-t-il si l'utilisateur ne saisit pas une valeur entière ? Tester avec les valeurs 12.5 et toto.

3.

Que se passe-t-il si vous effectuez les instructions suivantes juste après la saisie ?

Changer la première ligne en enlevant int() puis taper les lignes suivantes:

Tester avec toutes les valeurs précédentes.

if type(z)==type(1.0):

print "c'est un reel = %f" % float(z)

if type(z)==type(1):

print "c'est un entier = %d" % int(z)

if type(z)==type("a"):

print "chaine = %s" %z

Pour le moment, on fait attention à la saisie car on ne peut pas connaître le type d'une valeur saisie au clavier via raw\_input. On verra une manière de contrôler le type des saisies à l'exercice 5.

Exercice 2 – Instructions conditionnelles simples

On veut faire un programme qui convertit une durée exprimée sous la forme d'un entier au format

HHMMSS en une chaîne de caractères au format 'HHh MMmin Sss', en fonction de la durée renseignée. Ecrivez l'algorithme en français au stylo et papier ou sous un éditeur de texte, montrer le au professeur puis écrivez le programme en python.

Indication: pensez à la concaténation (exemple: "j'apprends"+"le python"="j'apprend le python")

Exemples d'exécution :

>>>Entrez la durée au format HMMSS

>>>123450

>>>12h 34min 50s

>>>Entrez la durée au format HMMSS

>>>3450

34min 50s

>>>Entrez la durée au format HMMSS

>>>50

>>>50s

>>>Entrez la durée au format HMMSS

>>>120050

>>>12h 50s

Exercice 3 – Code ASCII (chercher sur internet la table des codes ASCII)

1. Que fait ce programme ?

x=raw\_input("tapez sur une touche puis sur Entrée.")

print ord(x)

if (ord(x)>=48) and (ord(x)<=57):

print "que fait ce programme ?"

2. Modifier ce programme pour qu'il affiche :

- "Cette touche correspond a un chiffre" si c'est un chiffre qui a été saisie

- "Cette touche ne correspond pas à un chiffre" dans le cas contraire

3. Modifier ce programme pour qu'il affiche :

- "Cette touche correspond a un chiffre" si c'est un chiffre qui a été saisie

- "Cette touche correspond a une lettre majuscule" si c'est une lettre majuscule qui a été saisie

- "Cette touche correspond a une lettre minuscule" si c'est une lettre minuscule qui a été saisie

- "Cette touche n'est ni une lettre ni un chiffre" dans les autres cas.

Tester avec différentes touches.

Exercice 4 – Calcul de discriminant

1. Ecrivez un programme discriminant.py qui demande à l'utilisateurs d'entrez successivement 3 valeurs numériques a, b et c et qui affiche le discriminant d défini par d=b²-4ac de l'équation ax²+bx+c=0

2. Modifiez le programme pour qu'il affiche "Il existe au moins une solution réelle" si le discriminant est positif ou nul.

3. Modifiez le programme pour qu'il affiche "Il n'existe pas de solution réelle" si le discriminant est strictement négatif.

4. Modifier le programme pour qu'il affiche "Il existe une solution réelle double" si le discriminant est nul, "il existe 2 solutions réelles distinctes" si le discriminant est strictement positif, ainsi que leurs valeurs respectives.

Exercice 5 – Les Exceptions initiation

Les problèmes dans un programme ce sont les bugs. On voudrait faire tourner même s'il y a un bug. On peut pour cela utiliser la syntaxe:

try:

*instruction1*

except:

*instruction2*

avec cette syntaxe le programme fait l'instruction1 jusqu'à ce qu'il rencontre une erreur. Au lieu de bugger, il fera l'instruction2

Pour faire un exemple simple, utilisons le problème de la division par zéro. Ecrire un programme qui demande une valeur x (valeur réel de type float) calcul la valeur de f(x)=(2x–5)/(x-3) en utilisant un try:except:

Exercice 6 Les Exceptions:

Le but de cet exercice est d'écrire un programme qui demande à l’utilisateur deux **entiers (int(x))**, les affiche et ensuite le programme demande à l'utilisateur de donner le résultat de la multiplication de ces deux entiers.

1. Dans un premier temps, il n’est pas demandé de vérifier que le résultat de la multiplication est correct. Ecrivez ce programme et testez-le.

2. Quelles erreurs pourraient se produire durant l’exécution de ce programme ?

3. Modifiez ce programme pour qu’il affiche un message d’erreur au lieu de planter

4. Maintenant on souhaite vérifier que l’utilisateur a répondu le bon résultat et lui afficher un message de félicitations dans ce cas sinon lui dire qu’il s’est trompé et qu’il peut relancer le programme pour recommencer. Complétez le programme en ce sens

Exercice 6 – Lewis Caroll (avancé)

Ecrivez un programme qui répond au problème de logique donné par Lewis Caroll :

R1 : Il n'y a pas de chat non dressé aimant le poisson;

R2 : Il n'y a pas de chat sans queue jouant avec un gorille;

R3 : Les chats avec moustaches aiment toujours le poisson;

R4 : Il n'y a pas de chat dressé aux yeux verts;

R5 : Il n'y a pas de chats avec une queue, à moins d'avoir des moustaches.

Question: Les chats aux yeux verts jouent-ils avec les gorilles?