Nidhogg 1.0

Félix Foriel

21 février 2017

Chapitre 1

Introduction

J'ai créé le langage Nidhogg d'abord pour faire comprendre la programmation à mon père. Je venais juste d'apprendre le langage Perl et j'avais découvert ses formidables possibilitées en matière de manipulation des chaînes de caractères, je décidais donc d'entreprendre la programmation d'un interpréteur Nidhogg/Python. Il fut programmé en quelques jours et il rassemble les mots-clés de base du langage python, sa syntaxe coexiste en majuscule et en minuscule et les deux seront détaillées.

Une extension prenant en charge les modules Python sera peut-être un jour disponible. En attendant je vais vous détailler ici les différents mots-clés et leur signification dans l'ordre du script Perl.

Chapitre 2

Les Mots-clés en majuscules

2.1 SINON SI elif

condition elif

2.2 SINON else

condition else

2.3 SI ECHOUE except

except du bloc try

2.4 SI if

condition if

2.5 ALORS :

le " :", then en basic

2.6 TANT QUE while

boucle while

2.7 POUR CHAQUE foreach

boucle spécifique à un tableau (pour chaque élément du tableau faire ...)

2.8 POUR for

boucle for (pour chaque ... dans ... faire ...)

2.9 DANS in

in, s'utilise avec for (for ... in ...)

2.10 CASSER break

sert à sortir d'une boucle

2.11 CONTINUER continue

fait reboucler la boucle

2.12 TOUT EST VRAI all

savoir si chaque élément d'une liste est vrai (renvoie True si vrai)

2.13 EST VRAI PARMI any

savoir si un élément d'une liste est vrai (renvoie True)

2.14 VRAI True

booléen Vrai

2.15 FAUX False

booléen Faux

2.16 ECRIRE print

affiche du texte à l'écran

2.17 ENTRER UNE CHAINE raw_input

demande une saise à l'utilisateur qui sera convertie en chaîne de caractères

2.18 ENTRER input

demande une saisie à l'utilisateur

2.19 NOMBRE ALEATOIRE randint

donne un nombre aléatoire, s'utilise avec le module random

2.20 NOMBRE int

type nombre

2.21 DECIMALE float

type décimale

2.22 CHAINE str

type chaîne de caractères

2.23 EST EGAL A ==

opérateur égal

2.24 EST INFERIEUR OU EGAL A <=

opérateur inférieur ou égal

2.25 EST SUPERIEUR OU EGAL A >=

opérateur supérieur ou égal

2.26 EST DIFFERENT DE !=

opérateur différent

2.27 EST INFERIEUR A <

opérateur inférieur

2.28 EST SUPERIEUR A >

opérateur supérieur

2.29 EGAL =

égal, s'utilise lors d'un calcul

2.30 PLUS +

plus, s'utilise lors d'un calcul (a = 2+3)

2.31 MOINS -

moins, idem

2.32 PUISSANCE **

puissance, idem

2.33 DIVISE AVEC ENTIER PAR

diviser en récupérant un entier, idem

2.34 MULTIPLIE PAR *

multiplier, idem

2.35 DIVISE PAR

diviser, idem

2.36 DIVISE AVEC RESTE PAR %

diviser en récupérant le reste, idem

2.37 LONGUEUR len

retourne la longueur d'une chaîne de caractères ou d'une liste

2.38 SUPPRIMER del

supprime un élément d'une liste

2.39 AJOUTER append

ajoute un élément à une liste

2.40 TRIER LA LISTE sorted

trie une liste en créant une nouvelle liste

2.41 TRIER sort

trie une liste par ordre alphabétique ou numérique

2.42 LISTE INVERSEE reversed

inverse une liste en en créat une autre

2.43 INVERSER reverse

inverse les éléments d'une liste

2.44 INDICE index

trouve l'indice d'un élément d'une liste

2.45 ENLEVER remove

enlève un élément d'une liste

2.46 GAMME range

donne une liste préremplie d'éléments

2.47 DEPUIS from

sert à un importer un élément d'un module, s'utilise avec import (from ... import ...)

2.48 IMPORTER import

sert à importer un élément d'un module, s'utilise avec from (idem)

2.49 DEFINITION def

sert à définir une fonction (def ...(...) ...)

2.50 RENVOYER return

définit la valeur renvoyée par une fonction

2.51 GLOBALE global

définit une variable globale au programme, réutilisable en dehors d'une fonction

2.52 OUVRIR open

ouvre un fichier (open("nom_du_fichier", "mode"))

2.53 INSCRIRE write

écrit une ligne à l'intérieur d'un fichier

2.54 LIRE UNE LIGNE readline

lit une ligne d'un fichier

2.55 LIRE read

lit tout un fichier

2.56 FERMER close

ferme un fichier

2.57 TYPE type

renvoie le type d'une variable

2.58 MELANGER LA LISTE zip

regroupe sous la forme d'un tuple les items d'une liste

2.59 ADDITIONNER LA LISTE sum

additionne les valeurs d'un élément itérable

2.60 LISTE list

type liste

2.61 COMPTER LES OCCURENCES I count

compte le nombre d'occurences d'une recherche dans une chaîne

2.62 COMPTER LES OCCURENCES II count

compte le nombre d'occurences d'une recherche dans une liste

2.63 ENUMERER enumerate

énumère les différents éléments d'une liste, est un itérateur

2.64 RECUPERER get

retourne la valeur d'une clé donnée, s'utilise sur un dictionnaire

2.65 COPIER copy

copie un dictionnaire, dans un ordre différent

2.66 ABSOLUE abs

retourne la valeur absolue d'un nombre

2.67 BINAIRE bin

convertit un nombre entier en chaîne de caractères binaires

2.68 EST APPELLABLE callable

détermine si on peu appeller un objet, retourne True si vérifié

2.69 CAPITALISER capitalize

formate une chaîne de caractère avec le format Xxxxx

2.70 CHOIX choice

retourne aléatoirement une des valeurs d'une liste

2.71 NOM DES STRUCTURES dir

retourne les noms de la structure d'un objet

2.72 SE TERMINE PAR endswith

teste si une chaîne de caractère se termine par la chaîne demandée, retourne True si vérifié

2.73 CALCULER eval

calcule une chaîne de caractères (eval("2+2") retourne 4)

2.74 TROUVER find

donne la place de la première occurence de la chaîne demandée

2.75 AIDER help

retourne des informations sur un élément demandé

2.76 HEXADECIMAL hex

convertit un nombre en hexadécimal

2.77 EST ALPHANUMERIQUE isalnum

vérifie que tous les caractères d'une chaîne sont alphanumériques, renvoie True si vérifié

2.78 EST CARACTERIELLE isalpha

vérifie que tous les caractrèes d'une chaîne sont des lettres, idem

2.79 EST CHIFFREE isdigit

vérifie que tous les caractères d'une chaîne sont des chiffres, idem

2.80 EST MINUSCULE islower

vérifie que tous les caractères d'une chaîne sont en minuscule, idem

2.81 EST ESPACEE isspace

vérifie que la chaîne contient au moins un espace

2.82 EST UN TITRE istitle

vérifie que la chaîne à un format titre, c'est à dire Xxxxx Xx Xxx Xxxx

2.83 TITRER title

formate une chaîne au format titre, idem

2.84 EST MAJUSCULE isupper

vérifie que tous les caractères d'une chaîne sont en majuscules

2.85 JOINDRE join

transforme une liste en chaîne de caractères ("mon_séparateur".join(ma_liste))

2.86 LOCALES locals

retourne un dictionnaire avec les valeurs des variables en cours

2.87 MINUSCULISER lower

met en minuscules une chaîne de caractères

2.88 FAIRE map

execute une fonction sur chaque item d'une liste

2.89 MAXIMUM max

retourne la valeur la plus élevée d'une liste

2.90 MINIMUM min

retourne la valeur la plus basse d'une liste

2.91 ALEATOIRE random

retourne une valeur aléatoire

2.92 REMPLACER replace

remplace un segment d'une chaîne de caractères par une autre

2.93 ARRONDIR round

arrondi un nombre par le bas

2.94 MELANGER shuffle

mélange aléatoirement une liste

2.95 COMMENCER PAR startswith

vérifie qu'une chaîne de caractères commence par un préfixe indiqué

2.96 SEPARER LES LIGNES splitlines

retourne une liste des lignes d'une chaîne

2.97 SEPARER split

transforme une chaîne de caractères en liste ("ma_chaîne".split("mon_séparateur"))

2.98 MAJUSCULISER upper

met en majuscules une chaîne de caractères

2.99 ET and

vérifie qu'une condition ET une autre soient vérifiées

2.100 OU or

vérifie qu'une condition OU une autre soient vérifiées

2.101 PAS not

vérifie qu'une condition n'est pas vérifiée

2.102 ESSAYER try

essaye un bout de code

2.103 FINALEMENT finally

execute les instructions quelque soient les erreurs du bloc try

2.104 CLASSE class

regroupe les attributs et les fonctions qui définissent un objet

2.105 ELLE-MEME self

indique qu'il s'agit d'un attribut d'un objet

2.106 INITIALISER __init__

initialise un objet

2.107 NOUVEAU __new__

construit un objet

2.108 OBJET object

type objet

2.109 DICTIONNAIRE __dict__

donne les valeurs de l'attribut d'instance

2.110 DONNER yield

retourne une valeur sans la fin de la fonction

2.111 AVEC with

execute du code avant et après un autre code

2.112 COMME as

permet de récupérer une valeur dun with (with ... as ... : ...)

2.113 N'EST PAS is not

opérateur différent

2.114 EST is

opérateur égal

Chapitre 3

Les Mots-clés en minuscules

3.1 sinon si elif

condition elif

3.2 sinon else

condition else

3.3 si échoue except

except du bloc try

3.4 si if

condition if

3.5 alors :

le " :", then en basic

3.6 tant que while

boucle while

3.7 pour chaque foreach

boucle spécifique à un tableau (pour chaque élément du tableau faire ...)

3.8 pour for

boucle for (pour chaque ... dans ... faire ...)

3.9 dans in

in, s'utilise avec for (for ... in ...)

3.10 casser break

sert à sortir d'une boucle

3.11 continuer continue

fait reboucler la boucle

3.12 tout est vrai all

savoir si chaque élément d'une liste est vrai (renvoie true si vrai)

3.13 est vrai parmi any

savoir si un élément d'une liste est vrai (renvoie true)

3.14 vrai true

booléen vrai

3.15 faux false

booléen faux

3.16 écrire print

affiche du texte à l'écran

3.17 entrer une chaîne raw_input

demande une saise à l'utilisateur qui sera convertie en chaîne de caractères

3.18 entrer input

demande une saisie à l'utilisateur

3.19 nombre aléatoire randint

donne un nombre aléatoire, s'utilise avec le module random

3.20 nombre int

type nombre

3.21 décimale float

type décimale

3.22 chaîne str

type chaîne de caractères

3.23 est égal à ==

opérateur égal

3.24 est inférieur ou égal à <=

opérateur inférieur ou égal

3.25 est supérieur ou égal à >=

opérateur supérieur ou égal

3.26 est différent de !=

opérateur différent

3.27 est inférieur à <

opérateur inférieur

3.28 est supérieur à >

opérateur supérieur

3.29 égal =

égal, s'utilise lors d'un calcul

3.30 plus +

plus, s'utilise lors d'un calcul (a = 2+3)

3.31 moins -

moins, idem

3.32 puissance **

puissance, idem

3.33 divisé avec entier par

diviser en récupérant un entier, idem

3.34 multiplié par *

multiplier, idem

3.35 divisé par

diviser, idem

3.36 divisé avec reste par %

diviser en récupérant le reste, idem

3.37 longueur len

retourne la longueur d'une chaîne de caractères ou d'une liste

3.38 supprimer del

supprime un élément d'une liste

3.39 ajouter append

ajoute un élément à une liste

3.40 trier la liste sorted

trie une liste en créant une nouvelle liste

3.41 trier sort

trie une liste par ordre alphabétique ou numérique

3.42 liste inversee reversed

inverse une liste en en créat une autre

3.43 inverser reverse

inverse les éléments d'une liste

3.44 indice index

trouve l'indice d'un élément d'une liste

3.45 enlever remove

enlève un élément d'une liste

3.46 gamme range

donne une liste préremplie d'éléments

3.47 depuis from

sert à un importer un élément d'un module, s'utilise avec import (from ... import ...)

3.48 importer import

sert à importer un élément d'un module, s'utilise avec from (idem)

3.49 définition def

sert à définir une fonction (def ...(...) ...)

3.50 renvoyer return

définit la valeur renvoyée par une fonction

3.51 globale global

définit une variable globale au programme, réutilisable en dehors d'une fonction

3.52 ouvrir open

ouvre un fichier (open("nom_du_fichier", "mode"))

3.53 inscrire write

écrit une ligne à l'intérieur d'un fichier

3.54 lire une ligne readline

lit une ligne d'un fichier

3.55 lire read

lit tout un fichier

3.56 fermer close

ferme un fichier

3.57 type type

renvoie le type d'une variable

3.58 mélanger la liste zip

regroupe sous la forme d'un tuple les items d'une liste

3.59 additionner la liste sum

additionne les valeurs d'un élément itérable

3.60 liste list

type liste

3.61 compter les occurences I count

compte le nombre d'occurences d'une recherche dans une chaîne

3.62 compter les occurences II count

compte le nombre d'occurences d'une recherche dans une liste

3.63 énumérer enumerate

énumère les différents éléments d'une liste, est un itérateur

3.64 récupérer get

retourne la valeur d'une clé donnée, s'utilise sur un dictionnaire

3.65 copier copy

copie un dictionnaire, dans un ordre différent

3.66 absolue abs

retourne la valeur absolue d'un nombre

3.67 binaire bin

convertit un nombre entier en chaîne de caractères binaires

3.68 est appellable callable

détermine si on peu appeller un objet, retourne true si vérifié

3.69 capitaliser capitalize

formate une chaîne de caractère avec le format xxxxx

3.70 choix choice

retourne aléatoirement une des valeurs d'une liste

3.71 nom des structures dir

retourne les noms de la structure d'un objet

3.72 se termine par endswith

teste si une chaîne de caractère se termine par la chaîne demandée, retourne true si vérifié

3.73 calculer eval

calcule une chaîne de caractères (eval("2+2") retourne 4)

3.74 trouver find

donne la place de la première occurence de la chaîne demandée

3.75 aider help

retourne des informations sur un élément demandé

3.76 hexadécimal hex

convertit un nombre en hexadécimal

3.77 est alphanumérique isalnum

vérifie que tous les caractères d'une chaîne sont alphanumériques, renvoie true si vérifié

3.78 est caractérielle isalpha

vérifie que tous les caractrèes d'une chaîne sont des lettres, idem

3.79 est chiffrée isdigit

vérifie que tous les caractères d'une chaîne sont des chiffres, idem

3.80 est minuscule islower

vérifie que tous les caractères d'une chaîne sont en minuscule, idem

3.81 est espacée isspace

vérifie que la chaîne contient au moins un espace

3.82 est un titre istitle

vérifie que la chaîne à un format titre, c'est à dire xxxxx xx xxx xxxx

3.83 titrer title

formate une chaîne au format titre, idem

3.84 est majuscule isupper

vérifie que tous les caractères d'une chaîne sont en majuscules

3.85 joindre join

transforme une liste en chaîne de caractères ("mon_séparateur".join(ma_liste))

3.86 locales locals

retourne un dictionnaire avec les valeurs des variables en cours

3.87 minusculiser lower

met en minuscules une chaîne de caractères

3.88 faire map

execute une fonction sur chaque item d'une liste

3.89 maximum max

retourne la valeur la plus élevée d'une liste

3.90 minimum min

retourne la valeur la plus basse d'une liste

3.91 aléatoire random

retourne une valeur aléatoire

3.92 remplacer replace

remplace un segment d'une chaîne de caractères par une autre

3.93 arrondir round

arrondi un nombre par le bas

3.94 mélanger shuffle

mélange aléatoirement une liste

3.95 commence par startswith

vérifie qu'une chaîne de caractères commence par un préfixe indiqué

3.96 séparer les lignes splitlines

retourne une liste des lignes d'une chaîne

3.97 séparer split

transforme une chaîne de caractères en liste ("ma_chaîne".split("mon_séparateur"))

3.98 majusculiser upper

met en majuscules une chaîne de caractères

3.99 et and

vérifie qu'une condition et une autre soient vérifiées

3.100 ou or

vérifie qu'une condition ou une autre soient vérifiées

3.101 pas not

vérifie qu'une condition n'est pas vérifiée

3.102 essayer try

essaye un bout de code

3.103 finalement finally

execute les instructions quelque soient les erreurs du bloc try

3.104 classe class

regroupe les attributs et les fonctions qui définissent un objet

3.105 elle-même self

indique qu'il s'agit d'un attribut d'un objet

3.106 initialiser __init__

initialise un objet

3.107 nouveau __new__

construit un objet

3.108 objet object

type objet

3.109 dictionnaire __dict__

donne les valeurs de l'attribut d'instance

3.110 donner yield

retourne une valeur sans la fin de la fonction

3.111 avec with

execute du code avant et après un autre code

3.112 comme as

permet de récupérer une valeur dun with (with ... as ... : ...)

3.113 n'est pas is not

opérateur différent

3.114 est is

opérateur égal