Algorithmique et structure de données

Expressions et variables :

* Chaque langage est muni d’un certain type de base
* Il est également muni d’un certain nombre d’opérateurs
* En python : réel, entier, booléen et chaine de caractères

Entier grâce au type **int** (integer) (python)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Addition | Soustraction | Multiplication | Division entière | Modulo | Puissance |
| >>>3+10 | >>>56-7 | >>>3x4 | >>>45/16 | >>>67% | >>>4\*\*3 |
| 13 | 49 | 12 | 7 | 3 | 64 |

Priorité des opérateurs

La puissance est prioritaire sur la multiplication et prioritaire sur l’addition (on peut quand même mettre des parenthèses)

Autres opérations sur les réels

**Booléen**

Le type est bool

>>>true or false

True

>>>false and false

False

>>>not true

False

Tests :

* Egalité :

>>>10==3\*4-2

True

* Différence

>>>1+3 !=2+2

False

* Inférieur strict

>>> 3<10-7

False

* Supérieur strict

>>>10>4+3

True

* Supérieur ou égal

>>> 10>=5+3

True

* Inférieur ou égal

>>> 7<=4+3

False

**Chaine de caractères**

Le type est **str**

* **Affichage d’une chaine**

>>>  « hello »

‘hello’

* **Concaténation**

>>> « bon »+« jour »

‘bonjour’

* **Récupérer le 4ème caractère**

>>> « bonjour »[3]

‘j’

* **Récupérer la chaine jusqu’au 3ème caractère**

>>> « bonjour » [ :3]

‘bon’

* **Récupérer la chaine a partie du 4ème caractère**

>>> « bonjour »[3 :]

‘jour’

* **Extraction d’une chaine**

>>> « bonjour »[2 :5]

‘njo’

* **Chaine vide**

>>> »bonjour »[5 :3]

‘’

* **Longueur d’une chaine**

>>> len (« bonjour »)

7

Comparaison de chaine

* **Egalité**

« bonjour »== « bon »+ »jour »

TRUE

* **Différence**

« soir » !=  « bon »

**Variables**

Définition : Emplacement dans la mémoire de l’ordinateur, muni d’un nom et contenant une valeur variable (identificateur)

Affectation : opération consistant à fixer ou changer la valeur d’une variable -> identificateur =expression

Exemple :

>>> a=5+6

>>> a=11

(En python, le type d’une variable peut changer)

**Langage fortement et faiblement typé**

* Python est un langage dans la famille des faiblement typé
* Un langage fortement typé permet d’éviter les beugs

**Changement de type**

>>>3+  « 45 »

Provoque une erreur

>>> 3+int(« 45 ») => transformation de chaine en entier (donne le résultat : 48)

Str(4)+ « 5 » => transformation entier en chaine (résultat ‘45’)

**Saisie au clavier**

>>> a=input(« a= »)

a=45

>>>a

‘45’

>>> a=int(input(“a=”))

A=45

>>>a

45 => changement de type : entier avec int

**Affichage**

>>> a=6+7

>>>print (« a= »+str(a))

A=13

**Création d’une fonction**

Définition d’une fonction

>>> def f(x) :

Y=x+3

Return y ;

>>>f(4)

7

**Plusieurs paramètres**

>>>def g(a ;b)

Return a+b-4

>>>g(7,45)

48

**Plusieurs valeurs de retour**

>>>def g(a ;b)

Sum = a+b

Sub = a-b

Return sum, sub

>>>x; y=g(5,6)

>>>x

11

>>>y

-1

**Un vrai programme**

>>> def g()

a=int(input(« tapez la valeur de a »))

b=int(input(tapez la valeur de b »))

c=a+b

print(« la somme vaut : » +str (c))

>>>g()

Tapez la valeur de a : 10

Tapez la valeur de b : 20

La somme vaut : 30

**Bonne pratique : on écrit le contrat de la fonction**

Def : surface de ce rayon

Fonction paramètre

:entrée rayon : réel

:pré-cond : rayon>=0

:sortie surface : réel Contrat

:post-cond : surface vaut la surface d’un cercle de rayon valant rayon

Pi=301415926

S=pi\*rayon\*\*2 implémentation

Return s

**Entrées sorties**

Def go()

“’’”

:entrée r: reel SAISI au clavier

:pré-cond : r>=0

:sortie s : réel affiché à l’écran

:post-cond : s est la surface d’un cercle de rayon r

R=float((input(« tapez le rayon du cercle : »)))

S=surface (r )

Print (« la surface vaut : » +str(s))