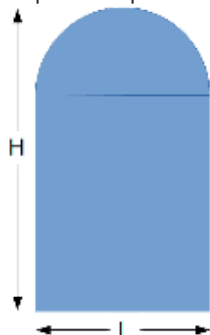


Les structures de données

Variables & Constantes

Activité 1

Ecrire un programme qui permet de calculer la quantité de peinture nécessaire pour peindre des portes de la forme suivante, sachant qu'un Kg de peinture permet de peindre une surface moyenne de 6 m².



$$S_{\text{demi cercle}} = \pi \cdot \frac{r^2}{2} = \pi \cdot \frac{\left(\frac{L}{2}\right)^2}{2} = \pi \cdot \frac{L^2}{4} \cdot \frac{1}{2} = \pi \cdot \frac{L^2}{8}$$

$$S_{\text{rectangle}} = \left(H - \frac{L}{2}\right) \cdot L$$

$$S_{\text{totale}} = S_{\text{demi cercle}} + S_{\text{rectangle}} = \pi \cdot \frac{L^2}{8} + \left(H - \frac{L}{2}\right) \cdot L$$

Solution Python

Python

```
PI = 3.14159265
QPM2 = 6 # 6 m²/Kg

# Entrée des données
l = float(input('Largeur de la porte : '))
h = float(input('Hauteur de la porte : '))
n = int(input('Donner le nbre de portes : '))

# Traitements
sp = (l * (h - l/2)) + PI * l * l / 8
qp = (n * 2 * sp) / QPM2

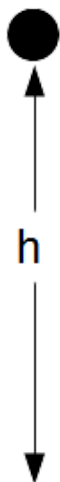
# Affichage
print('Quantité de peinture requise :', qp, 'Kg')
```

Activité 2

1. Relever à partir du programme précédent la structure générale d'un programme.
2. Quelle est la différence entre les objets **PI**, **QPM2** et **l**, **h** et **n** ?

Activité 3

Un projectile est laissé en chute libre sans vitesse initiale à partir d'un point M haut du sol. On souhaite écrire un programme qui permet de calculer le temps de chute.



$$t_{\text{chute}} = \sqrt{\frac{2 \cdot h}{g}}$$

h : hauteur du chute
g : Accélération du pesanteur

A Retenir

Une **constante** est un objet ayant une valeur fixe tout le long de l'exécution d'un algorithme ou d'un programme.

Une **constante** est caractérisée par son **nom** et sa **valeur**.

Une **variable** est un objet pouvant prendre différentes valeurs tout le long de l'exécution d'un algorithme ou d'un programme.

Une variable est caractérisée par son **nom**, son **type** et son **contenu**.

Types de données

Activité 4

Relever les types de données utilisés dans le programme suivant :

Python

```
nom = 'Sahar'
age = 18
genre = 'F'
p = 60.0
h = 1.60

imc = p / (h * h)
ok = (imc >= 18) and (imc <= 23)

if genre in ['G', 'g']:
    sal = 'Mr'
elif genre in ['F', 'f']:
    sal = 'Mlle/Mme'
else:
    sal = ''

...
```

A retenir

Le **type d'une variable** permet de déterminer le **domaine de valeurs possible** que peut prendre cette variable.

La connaissance du type permet également de :

- Déduire **les opérateurs** applicables sur les variables de ce type
- Déduire **l'espace mémoire** (en octets) qui sera réservé à ces variables
- Déterminer l'ensemble de **fonctions prédéfinies** compatible avec ce type de variables

Le type Entier

En algorithmique, le type entier représente un sous ensemble du corps des entiers relatifs \mathbb{Z} .

En Python

Python définit un seul type entier **int** et sa taille est illimitée.

Opérations sur les entiers

Le tableau suivant définit les opérations sur les entiers.

On suppose dans les exemples suivants que $a = 5$, $b = 3$.

Opérateurs

Algorithme	Signification	Exemple Python
+	Addition	<code>c = a + b # c = 8</code>
-	Soustraction	<code>d = a - b # d = 2</code>
*	Multiplication	<code>e = a * b # e = 15</code>
/	Division réelle	<code>f = a / b # f = 1.666666</code>
div	Quotient de la division entière	<code>g = a // b # g = 1</code>
mod	Reste de la division entière	<code>h = a % b # h = 2</code>
	Exponentiation	<code>i = a ** b # i = 125</code>

Activité 5

Evaluer les expressions arithmétiques, algorithmiques, suivantes :

- 17 DIV 5
- 17 MOD 5
- $((58 \text{ DIV } 7) \text{ MOD } 2) + 5$
- $(49 \text{ MOD } 17) \text{ DIV } (4 * 3)$

Le type réel

Le type réel représente les valeurs à virgule flottantes, exemple : 1.5, 170.5, etc.

En Python

Le langage Python définit le type `float` pour représenter les nombres décimaux.

Fonctions pour les types numériques

Nom Fonction	Python	Rôle	Test
<code>abs(x)</code>	<code>abs(x)</code>	Valeur Absolue de x	<code>abs(-73) = 73</code>
<code>racine(x)</code>	<code>sqrt(x)</code>	La racine carré de x	<code>racine(19) = 4.358898943540674</code>
<code>ent(x)</code>	<code>int(x)</code>	Supprime la partie fractionnaire de x et renvoie un entier	<code>ent(-33.39999999999999) = -33</code>
<code>arrondi(x)</code>	<code>round(x)</code>	Retourne l'entier le plus proche de x	<code>arrondi(-4.3999999999999915) = -4</code>
<code>alea(a, b)</code>	<code>from random import randint randint(a, b)</code>	Spécifique Python Retourne un entier aléatoire dans l'intervalle [a, b]	<code>alea(10, 30) = 22</code>

Activité 6

Ecrire un programme qui permet d'afficher les valeurs suivantes :

0.0	-55.36	3.14	$60 \cdot 10^{-9}$	1.23	-38.0	$5.6 \cdot 10^6$
-----	--------	------	--------------------	------	-------	------------------

Solution

Python

```
# program prog03
print(0.0)
print(-55.36)
print(3.14)
print(60E-9)
print(1.23)
print(-38.0)
print(5.6E6)
```

Le type booléen



Une variable de type booléen prend uniquement deux valeurs différentes **vrai** ou **faux**.

En Python

Le type booléen correspond à `bool` en Python.

Les opérateurs

Opérateur	Signification	Python
NON	Fonction inverse	<code>not a</code>
ET	Fonction ET	<code>a and b</code>
OU	Fonction OU	<code>a or b</code>

Compléter les tableaux de vérités suivants :

Fonction NON

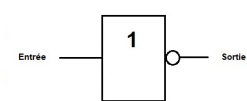


Figure 1, Schéma électrique

A	NON A
Faux	-
Vrai	-

Fonction OU

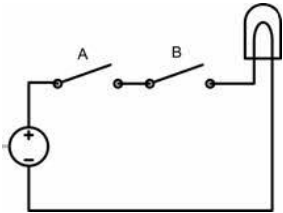


Figure 2, Schéma électrique

A	B	A OU B
F	V	-
F	V	-
V	F	-
V	V	-

Fonction ET

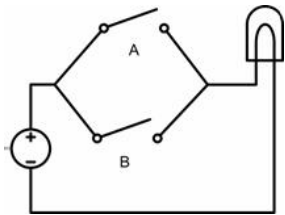


Figure 3, Schéma électrique

A	B	A ET B
F	V	-
F	V	-
V	F	-
V	V	-

Fonction XOR

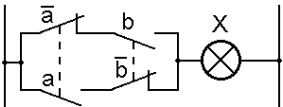


Figure 4, Schéma électrique

A	B	A XOR B
F	V	
F	V	
V	F	
V	V	

Activité 9

Evaluer les expressions logiques suivantes :

- $(x \leq 4)$ ET $(x \geq 1)$ pour $x = 3$
- $(x \leq 2)$ ET $(x \geq 0)$ pour $x = -4$
- NON $(x \leq 55)$ OU $(x \geq 0)$ pour $x = 21$

Le type caractère

Il s'agit du domaine constitué des **caractères alphabétiques** et **numériques**. Une variable de ce type ne peut contenir qu'un seul et unique caractère.

Les caractères sont ordonnés selon leurs code ASCII. En ASCII, il est possible de représenter 256 caractères.

La table suivante contient les 128 premiers caractères standard, code 0 → 127, les caractères de code 128 → 255 ne sont pas standard.

Regular ASCII Chart (character codes 0 - 127)											
000 (nul)	016 ► (dle)	032 sp	048 0	064 @	080 P	096 `	112 p				
001 ☉ (soh)	017 ◄ (dcl)	033 !	049 1	065 A	081 Q	097 a	113 q				
002 Ⓢ (stx)	018 ↑ (dc2)	034 "	050 2	066 B	082 R	098 b	114 r				
003 ▼ (etx)	019 !! (dc3)	035 #	051 3	067 C	083 S	099 c	115 s				
004 † (eot)	020 ¶ (dc4)	036 \$	052 4	068 D	084 T	100 d	116 t				
005 ♣ (enq)	021 \$ (nak)	037 %	053 5	069 E	085 U	101 e	117 u				
006 ‡ (ack)	022 − (syn)	038 &	054 6	070 F	086 V	102 f	118 v				
007 ▪ (bel)	023 ‡ (etb)	039 '	055 7	071 G	087 W	103 g	119 w				
008 ■ (bs)	024 † (can)	040 (056 8	072 H	088 X	104 h	120 x				
009 (tab)	025 ↓ (em)	041)	057 9	073 I	089 Y	105 i	121 y				
010 (lf)	026 (eof)	042 *	058 :	074 J	090 Z	106 j	122 z				
011 ♂ (vt)	027 − (esc)	043 +	059 ;	075 K	091 [107 k	123 {				
012 ‡ (np)	028 L (fs)	044 ,	060 <	076 L	092 \	108 l	124				
013 (cr)	029 ↔ (gs)	045 −	061 =	077 M	093]	109 m	125 }				
014 ♂ (so)	030 ▲ (rs)	046 .	062 >	078 N	094 ^	110 n	126 ~				
015 ♂ (si)	031 ▼ (us)	047 /	063 ?	079 0	095 _	111 o	127 ª				

En Python

Python ne possède pas un type caractère un **str** de longueur 1 est considéré comme étant un caractère.

Fonctions prédéfinies sur les caractères

Le tableau suivant résume les principales fonctions prédéfinies sur les caractères :

Fonction	Python	Exemples
Code ASCII d'un caractère	ORD(car)	ORD("0") → 48 ORD("A") → 65 ORD("a") → 97
Caractère correspondant à un code ASCII	CHR(code)	CHR(48) → "0" CHR(65) → "A" CHR(97) → "a"
Mettre un caractère en majuscules	car.upper()	MAJUS("a") → "A" MAJUS("A") → "A" MAJUS("0") → "0"

Activité 10

Ecrire un programme qui permet de saisir une lettre Majuscule puis l'affiche en minuscules.

Solution

Algorithme

Algorithme prog05

Début

Ecrire("Entrer une lettre majuscule : ")

Lire(cmaj)

cmin ← Chr(Ord(cmaj) + 32)

Ecrire("Miniscule(' ", cmaj, "') = ' ", cmin, "')

Fin

TDO

Objet	Type
cmin, cmaj	caractère

Le type chaîne de caractères

Une chaîne de caractères est une suite de caractères. Elle peut être définie comme suit :

Objet	Type/Nature
Prénom	Chaîne
Nom	Chaîne

En Python

Le type `str` correspond en Python à une chaîne de caractères. En Python, on ne peut pas limiter le nombre de caractères dans une chaîne.

Activité 11

Déclarer deux variables `nom` et `prenom`. Puis, les initialiser avec votre nom et prénom.

Solution

Python

```
nom = "MANI"
prenom = "Mohamed Anis"
```

Accès aux caractères

Les caractères d'une chaîne sont accessibles via leur indice qui commence toujours à partir de 0.

La fonction `len(ch)` indique le nombre de caractères dans la chaîne, en comptant aussi les espaces vides.

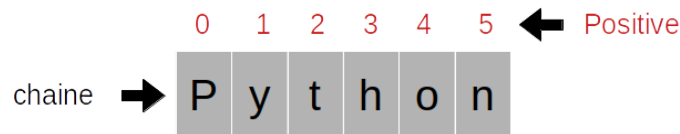


Figure 5, Indices dans une chaîne

Exemple :

Le caractère `"t"` dans la chaîne `"Python"` se trouve à la position : 2

Fonctions sur les chaînes - Python

Le tableau suivant résume l'ensemble des fonctions prédéfinies sur les chaînes en Pascal

Fonction	Algorithme	Python	Test
Retourne la longueur d'une chaîne de caractères	<code>long(ch)</code>	<code>len(ch)</code>	<code>long("Bac 2022") → 8</code>
Retourne la première position de <code>ch1</code> dans <code>ch2</code>	<code>pos(ch1, ch2)</code>	<code>ch2.find(ch1)</code>	<code>pos("02", "Bac 2022") → 5</code>
Retourne une sous chaîne de caractères de <code>ch</code> de la position <code>p1</code> jusqu'à <code>p2</code> non incluse.	<code>sous_chaine(ch, p1, p2)</code>	<code>ch[p1:p2]</code>	<code>sous_chaine("Baccar", 3, 6) → "car"</code>
Supprime les caractères de <code>ch</code> à partir de la position <code>p1</code> à la position <code>p2</code> non incluse.	<code>efface(ch, p1, p2)</code>	<code>ch[:p1] + ch[p2:]</code>	<code>ch ← "Baccar"</code> <code>ch ← efface(ch, 3, 6)</code> <code>ch contient "Bac"</code>
Convertit une chaîne <code>ch</code> en une valeur numérique.	<code>Valeur(ch)</code>	# pour les entiers <code>x = int(ch)</code> # pour les réels <code>x = float(ch)</code>	<code>x ← valeur("18.5")</code> <code>x contient 18.5</code>
Convertit une valeur numérique en chaîne.	<code>ch ← ConvCh(x)</code>	# x réel ou entier <code>ch = str(x)</code>	<code>ch ← ConvCh(15)</code> <code>ch contient "15"</code>

Le type tableau

Le type tableau permet de stocker un ensemble de valeurs de même type. Les éléments du tableau sont accessibles via le nom du tableau suivi par l'indice de l'élément.

Exemple :

t[3] : Permet d'accéder à la case d'indice 3 dans le tableau t.

Un tableau est déclaré de la façon suivante :

nom_type = tableau de **nbre_éléments** **type**

Exemple

Pour déclarer un tableau qui peut contenir jusqu'à 20 entiers, on écrit :

tab = tableau de **15** **entier**

Déclaration en Python

En Python, on écrit tout simplement :

Python

```
from numpy import array
t = array([int()] * 15)
```

Le code précédent crée un tableau pour stocker **15** entiers.

t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Indexation

En Python, l'indexation se fait toujours à partir de l'indice **0**, qui est l'indice du premier élément. L'indice du dernier élément étant **14**.

Activité 12

Déclarer les tableaux suivants en Python :

te	58	64	11	70	1	7	62	6	52	69
	0	1	2	3	4	5	6	7	8	9
tr	2.5	0.6	9.5	4.5	4.2	2.3	1.6	9.9		
	0	1	2	3	4	5	6	7		
tb	True	False	False	False	True	False	...	False	True	
	0	1	2	3	4	5	...	24	25	

Python

```
from numpy import array
te = array([int()] * 10)
tr = array([float()] * 7)
tb = array([bool()] * 26)
```