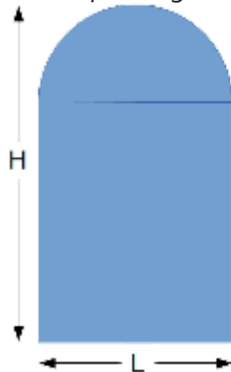


## Les structures de données

### 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 m<sup>2</sup>.



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

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

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

#### Solution Pascal

##### Pascal

```

program prog01;
{ Déclarations }
const
  PI = 3.14159265;
  QPM2 = 6; { 6 m²/Kg }
var
  n : integer;
  l, h : real;
  sp, qp : real;
begin
  { Entrée des données }
  Write('Largeur de la porte : ');
  Readln(l);

  Write('Hauteur de la porte : ');
  Readln(h);

  Write('Donner le nbre de portes : ');
  Readln(n);

  { Traitements }
  sp := (l * (h - l/2)) + PI * l * l / 8;
  qp := (n * 2 * sp) / QPM2;

  { Affichage }
  Writeln('Quantité de peinture requise : ', qp:5:1, 'Kg');
  Readln;
end.

```

## 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')

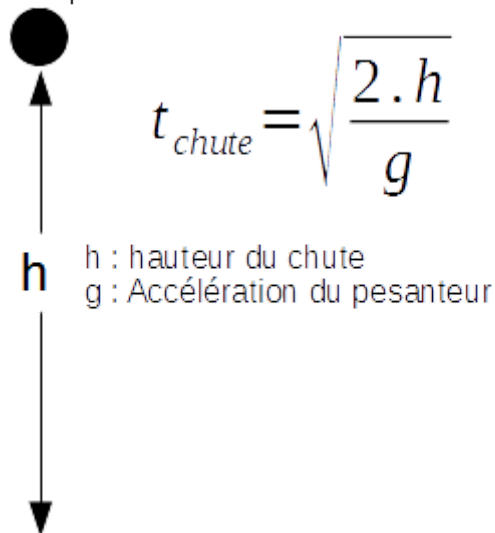
```

### 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.



### 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 :

#### Pascal

```
program prog02.pas;
{Déclarations}
var
    ...
{P.P}
begin
    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'] then sal := 'Mr'
    else if genre in ['F', 'f'] then sal := 'Mlle/Mme'
    else sal := '';

    ...
end.
```

#### 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 :

- Dédurre **les opérateurs** applicables sur les variables de ce type
- Dédurre **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 algorithm, le type entier représente un sous ensemble du corps des entiers relatifs  $\mathbb{Z}$ .

#### En Pascal

Le langage Pascal définit cinq types entiers. Chaque type a un domaine de définition spécifique qui dépend de sa taille.

Type	Domaine de définition	Taille (bits)
SHORTINT	-128..127	8 bits
INTEGER	-32768..32767	16 bits
LONGINT	-2147483648..2147483647	32 bits
BYTE	0..255	8 bits
WORD	0..65535	16 bits

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

### Activité 5

Evaluer les expressions arithmétiques, algorithmiques, suivantes :

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

### Le type réel

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

#### En Pascal

Le langage Pascal définit plusieurs types pour les réels le plus utilisé est : **real**.

#### En Python

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

### Fonctions pour les types numériques

Nom Fonction	Pascal	Python	Rôle	Test
abs(x)	abs(x)	abs(x)	Valeur Absolue de x	abs(58) = 58
racine(x)	sqrt(x)	x ** 0.5	La racine carré de x	racine(72) = 8.48528137423857
carré(x)	sqr(x)	x ** 2	Le carré de x	carré(67) = 4489
tronc(x)	trunc(x)	trunc(x)	Supprime la partie fractionnaire de x et renvoie un entier	tronc(63.900000000000006) = 63
arrondi(x)	round(x)	round(x)	Retourne l'entier le plus proche de x	arrondi(74.5) = 75

Nom Fonction	Pascal	Python	Rôle	Test
alea	random	from random import random random()	Retourne un réel aléatoire dans l'intervalle [0, 1[	alea = 0.8336996864350688
alea(x)	random(x)	from random import randint randint(0, x-1)	Spécifique Pascal Retourne un entier aléatoire dans l'intervalle [0, x[	alea(100) = 89
alea(a, b)	a + random(b - a + 1)	from random import randint randint(a, b)	Spécifique Python Retourne un entier aléatoire dans l'intervalle [a, b]	alea(10, 30) = 21
impair(x)	odd(x)	(x % 2 == 1)	Spécifique Pascal Retourne si un entier x est impair	odd(-86) = Vrai

## 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

## Pascal

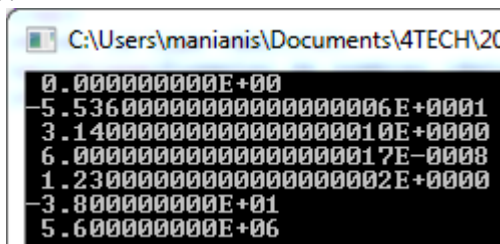
```
program prog03;  
begin  
  Writeln(0.0);      Writeln(-55.36);  
  Writeln(3.14);     Writeln(60E-9);  
  Writeln(1.23);     Writeln(-38.0);  
  Writeln(5.6E6);  
  Readln;  
end.
```

# Python

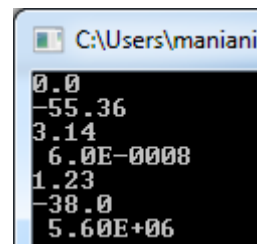
```
# program prog03
print(0.0)
print(-55.36)
print(3.14)
print(60E-9)
print(1.23)
print(-38.0)
print(5.6E6)
```

## Activité 7 - Pascal

Le programme précédent permet d'obtenir le résultat suivant :



Rectifier votre programme pour obtenir l'affichage suivant :

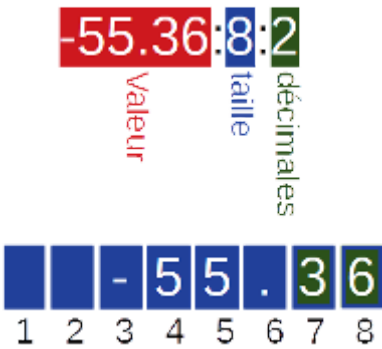


## Pascal

```
program prog04;  
begin  
  Writeln(0.0:3:1);      Writeln(-55.36:6:2);  
  Writeln(3.14:4:2);     Writeln(60E-9:10);  
  Writeln(1.23:4:2);     Writeln(-38.0:5:1);  
  Writeln(5.6E6:9);  
  Readln;  
end.
```

A retenir - Pascal

La mise en forme des réels est effectuée de la façon suivante :



3.2E-5 est équivalente à 3,2×10<sup>-5</sup>

Le type booléen



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

En Pascal

Le type booléen correspond à `boolean` en Pascal.

En Python

Le type booléen correspond à `bool` en Python.

Les opérateurs

Opérateur	Signification	Pascal	Python
NON	Fonction inverse	not a	not a
ET	Fonction ET	a and b	a and b
OU	Fonction OU	a or b	a or b
OUEX	Fonction OU Exclusif	a xor b	a ^ b

Compléter les tableaux de vérités suivants :

### Fonction NON

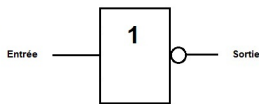


Figure 1, Schéma électrique

A	NON A
Faux	-
Vrai	-

### Fonction OU

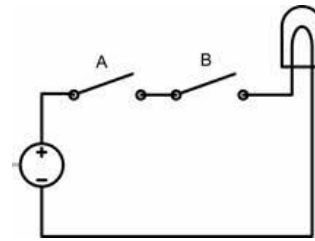


Figure 2, Schéma électrique

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

### Fonction ET

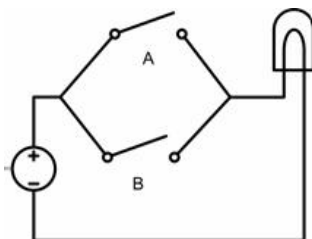


Figure 3, Schéma électrique

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

### Fonction XOR

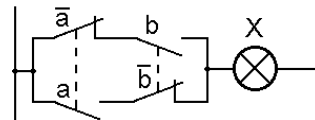


Figure 4, Schéma électrique

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  $\rightarrow$  127, les caractères de code 128  $\rightarrow$  255 ne sont pas standard.

Regular ASCII Chart (character codes 0 - 127)									
000	(nul)	016	(dle)	032	sp	048	0	064	0
001	(soh)	017	(dcl)	033	!	049	1	065	A
002	(stx)	018	(dc2)	034	"	050	2	066	B
003	(etx)	019	(dc3)	035	#	051	3	067	C
004	(eot)	020	(dc4)	036	\$	052	4	068	D
005	(enq)	021	(nak)	037	%	053	5	069	E
006	(ack)	022	(syn)	038	&	054	6	070	F
007	(bel)	023	(etb)	039	'	055	7	071	G
008	(bs)	024	(can)	040	(	056	8	072	H
009	(tab)	025	(em)	041	)	057	9	073	I
010	(lf)	026	(eof)	042	*	058	:	074	J
011	(vt)	027	(esc)	043	+	059	;	075	K
012	(np)	028	(fs)	044	,	060	<	076	L
013	(cr)	029	(gs)	045	-	061	=	077	M
014	(so)	030	(rs)	046	.	062	>	078	N
015	(si)	031	(us)	047	/	063	?	079	0
								095	-
								111	o
								112	p
								113	q
								114	r
								115	s
								116	t
								117	u
								118	v
								119	w
								120	x
								121	y
								122	z
								123	{
								124	
								125	}
								126	~
								127	0

## En Pascal

Pascal définit le type `char` pour stocker un caractère.

## 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	Pascal	Python	Exemples
Code ASCII d'un caractère	ORD(car)	ORD(car)	ORD("0") → 48 ORD("A") → 65 ORD("a") → 97
Caractère correspondant à un code ASCII	CHR(code)	CHR(code)	CHR(48) → "0" CHR(65) → "A" CHR(97) → "a"
Successeur d'un caractère	SUCC(car)	Spécifique Pascal chr(ord(car) + 1)	SUCC("A") → "B"
Prédécesseur d'un caractère	PRED(car)	Spécifique Pascal chr(ord(car) - 1)	PRED("A") → "@"
Mettre un caractère en majuscules	UPCASE(car)	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	Rôle
Prénom	Chaîne [10]	Prénom d'un individu sur 10 caractères maximum
Nom	Chaîne	Chaîne pouvant contenir jusqu'à 255 caractères.

### En Pascal

Le type `string` correspond en Pascal à une chaîne de caractères.

### 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

#### Pascal

```
var
  Prenom: string[20];
  Nom: string;
begin
  Nom := 'MANI';
  Prenom := 'Mohamed Anis';
end.
```

#### Python

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

### Accès aux caractères - Pascal

En Pascal, les caractères d'une chaîne sont accessibles via leur indice qui commence de 1. La figure suivante explique comment accéder aux caractères d'une chaîne et les modifier.

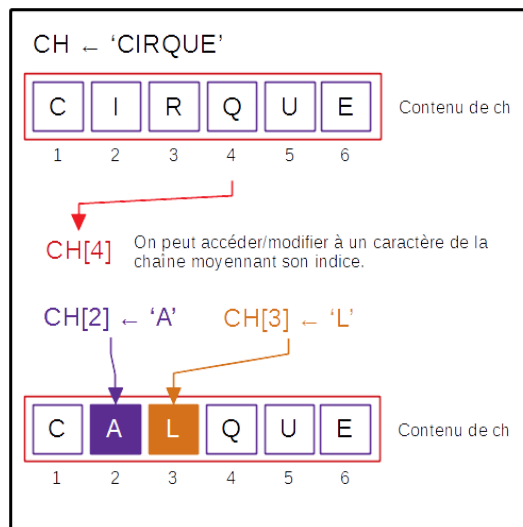


Figure 5, Indices dans une chaîne

## Accès aux caractères - Python

En Python, les caractères d'une chaîne sont accessibles via leur indice qui commence de 0.

Python supporte deux types d'indexation :

- Positive : qui commence de 0 jusqu'à `len(ch)-1`
- Négative : qui comment de -1 jusqu'à `-len(ch)`

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

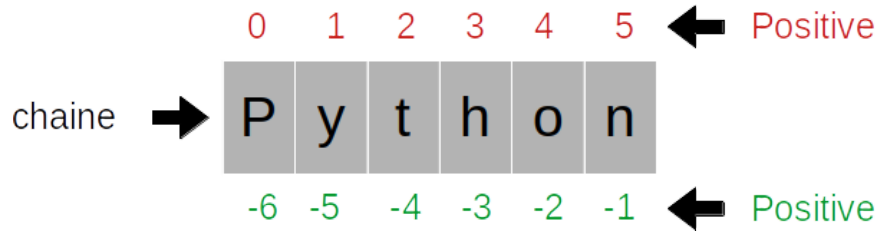


Figure 6, Indices dans une chaîne

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

- 2 en utilisant l'indexation positive
- -4 en utilisant l'indexation négative

## Fonctions sur les chaînes - Pascal

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

Fonction	Algorithme	Pascal	Test
Retourne la longueur d'une chaîne de caractères	<code>long(ch)</code>	<code>length(ch)</code>	<code>long("Bac 2022") → 8</code>
Retourne la <b>première position</b> de <code>ch1</code> dans <code>ch2</code>	<code>pos(ch1, ch2)</code>	<code>pos(ch1, ch2)</code>	<code>pos("02", "Bac 2022") → 6</code>
Retourne une sous chaîne de <code>nbc</code> caractères à partir de la position <code>pos</code> de <code>ch</code>	<code>sous_chaine(ch, pos, nbc)</code>	<code>copy(ch, pos, nbc)</code>	<code>sous_chaine("Baccar", 4, 3) → "car"</code>
Supprime <code>nbc</code> caractères de <code>ch</code> à partir de la position <code>pos</code> .	<code>efface(ch, pos, nbc)</code>	<code>delete(ch, pos, nbc)</code>	<code>ch ← "Baccar"</code> <code>efface(ch, 4, 3)</code> <code>ch</code> contient "Bac"
Insérer <code>ch1</code> dans <code>ch2</code> à la position <code>pos</code> .	<code>Inserer(ch1, ch2, pos)</code>	<code>Insert(ch1, ch2, pos)</code>	<code>ch1 ← "Ka"</code> <code>ch2 ← "Librairie rim"</code> <code>Inserer(ch1, ch2, 11)</code> <code>ch</code> contient "Librairie Karim"
Convertit une chaîne <code>ch</code> en une valeur numérique <code>x</code> , si la chaîne n'est pas une valeur numérique ou contient des erreurs renvoyer la position de l'erreur <code>e</code> .	<code>Valeur(ch, x, e)</code>	<code>Val(ch, x, e)</code>	<code>val("18.5", x, e)</code> <code>x</code> contient 18.5 <code>e</code> contient 0
Convertit une valeur numérique en chaîne.	<code>ConvCh(x, ch)</code>	<code>Str(x, ch)</code>	<code>ConvCh(15, ch)</code> <code>ch</code> contient "15"

## 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	long(ch)	length(ch)	long("Bac 2022") → 8
Retourne la <b>première position</b> de <b>ch1</b> dans <b>ch2</b>	pos(ch1, ch2)	ch2.find(ch1))	pos("02", "Bac 2022") → 5
Retourne une sous chaîne de caractères de <b>ch</b> de la position <b>pos1</b> jusqu'à <b>pos2</b> non incluse. <b>ch</b>	sous_chaine(ch, pos1, pos2)	ch[pos1:pos2]	sous_chaine("Baccar", 3, 6) → "car"
Supprime les caractères de <b>ch</b> à partir de la position <b>pos1</b> à la position <b>pos2</b> .	efface(ch, pos1, pos2)	ch ← ch[:pos1] + ch[pos2:]	ch ← "Baccar" ch ← efface(ch, 3, 6) ch contient "Bac"
Insérer <b>ch1</b> dans <b>ch2</b> à la position <b>pos</b> .	Inserer(ch1, ch2, pos)	ch2[:pos] + ch1 + ch2[pos:]	ch1 ← "Ka" ch2 ← "Librairie rim" Inserer(ch1, ch2, 10) ch contient "Librairie Karim"
Convertit une chaîne <b>ch</b> en une valeur numérique.	Valeur(ch)	x = int(ch) # pour les entiers x = float(ch) # pour les réels	x ← val("18.5") x contient 18.5
Convertit une valeur numérique en chaîne.	ch ← ConvCh(x)	ch = str(x) # pour x réel ou entier	ch ← ConvCh(15) ch contient "15"

## 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 Pascal

Un tableau peut être déclaré de l'une des façons suivantes :

### Méthode 1

#### Pascal

```
// Déclaration directe
var
  t : array [1..15] of integer;
```

### Méthode 2

#### Pascal

```
// Déclaration d'un nouveau type
type
  tab = array [1..15] of integer;
// Déclaration du tableau
var
  t : tab;
```

La **deuxième méthode est la méthode conseillée** en Pascal, car on en aura besoin lorsqu'on abordera les sous-programmes.

t	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	15

## Indexation

Lors de sa déclaration on indique l'intervalle d'indexation du tableau. Ainsi, dans l'exemple précédent, le premier élément du tableau **t** possède l'indice **1** alors que le dernier possède l'indice **15**.

## Déclaration en Python

En Python, on écrit tout simplement :

### Python

```
t = [int()] * 15
```

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

	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
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**.

En Python, on peut aussi utiliser l'**indexation négative** marquée en **rouge**.

## Activité 12 - Pascal

Déclarer les tableaux suivants en Pascal :

te	14	63	97	76	85	53	13	5	95	23
	1	2	3	4	5	6	7	8	9	10

tr	0.1	8.6	3.5	6.7	5.5	6.5	4.8
	-3	-2	-1	0	1	2	3

tb	True	True	True	False	False	False	...	False	True
	"A"	"B"	"C"	"D"	"E"	"F"	...	"Y"	"Z"

### Pascal

type

```
tab_e = array [1..10] of integer;
tab_r = array [-3..3] of real;
tab_b = array ['A'..'Z'] of boolean;
```

var

```
te : tab_e;
tr : tab_r;
tb : tab_b;
```

Activité 12 - Python

Déclarer les tableaux suivants en Python :

te	80	65	38	16	19	36	79	27	77	76
	0	1	2	3	4	5	6	7	8	9

tr	2.7	9.7	0.9	3.4	0.7	9.1	1.3	5.2
	0	1	2	3	4	5	6	7

tb	False	True	True	False	True	True	...	True	True
	"0"	"1"	"2"	"3"	"4"	"5"	...	24	25

Python

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