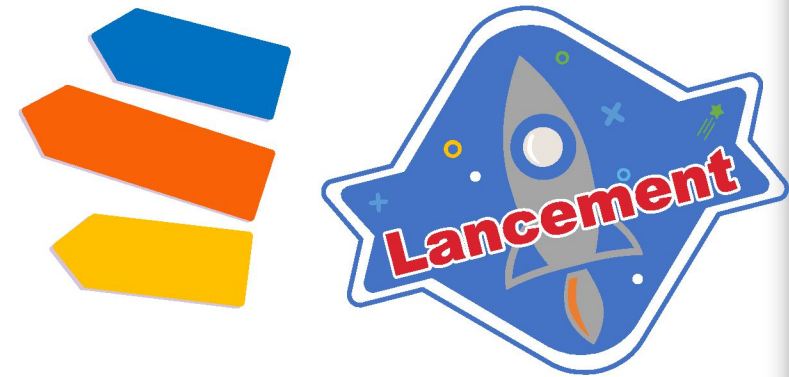




Les Types des Données :



Introduction



- Chaque variable est caractérisée par:
 1. Son nom/identificateur
 2. Son type
 3. Sa valeur/contenu

- En algorithmique, il existe **5 types des données**:
 1. L'Entier
 2. Le Réel
 3. Le Booléen
 4. Le caractère
 5. La chaîne des caractères

1. Le type Entier



- **Déclaration**

T.D.O

Objet	Type/Nature
Nom_variable	Entier

- **Les opérateurs**

+
-
*
/
mod
Div
< , > , = , >= , <= , !=

1. Le type Entier



Essayer de remplir le tableau ci-dessous:

Instruction algorithmique	Traduction en Python	Evaluation :
$A \leftarrow ((40 \text{ DIV } 3) \text{ MOD } 4) + 6$	$A = ((40 // 3) \% 4) + 6$	$A = 7$
$B \leftarrow 22 - 25 \text{ MOD } (2 * 3) + 1$	$B = 22 - 25 \% (2 * 3) + 1$	$B = 22$
$C \leftarrow 15 - (12 \text{ MOD } 2) * 3 - 16$	$C = 15 - (12 \% 2) * 3 - 16$	$C = -1$
$D \leftarrow (18 + 20 \text{ MOD } 3) * 2 - 18$	$D = (18 + 20 \% 3) * 2 - 18$	$D = 22$
$E \leftarrow 2 * 2 - 25 \text{ DIV } 7 * 10$	$E = 2 * 2 - 25 // 7 * 10$	$E = -26$

Exercice:

Ecrire un algorithme intitulé CHIFFRE permettant de saisir un nombre « N » formé de 3 chiffres, de calculer et d'afficher la somme S et le produit P des chiffres formant « N » .

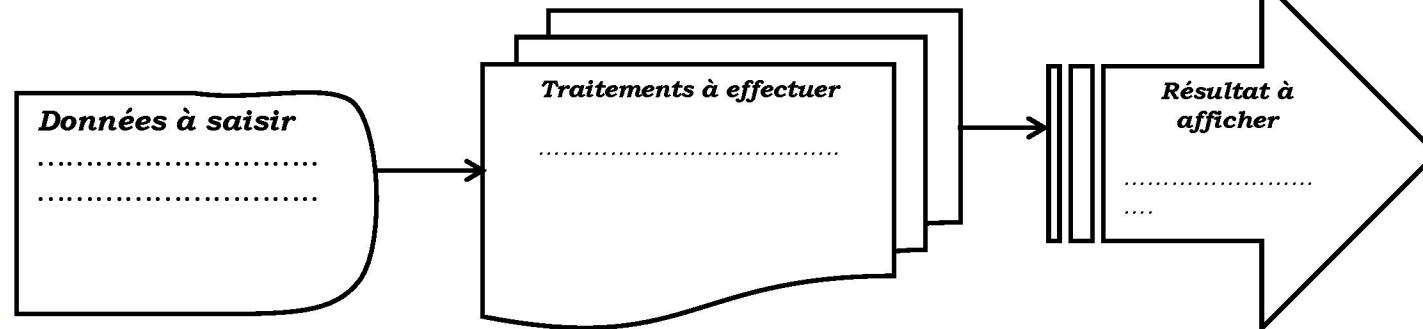
Exemple:

N= 265

Le programme affichera :

- La somme des chiffres est = 13 ($2+6+5 = 13$)
- Le produit des chiffres est = 60 ($2*6*5=60$)

• Etape N°1 : Analyse



• Etape N°2 : Algorithmme



ALGORITHMME

DEBUT

.....

.....

.....

.....

T.D.O

Objet	Type
.....
.....
.....

2. Le type Réel



T.D.O

- *Déclaration*

Objet	Type/Nature
Nom_variable	Réel

- *Les opérateurs*

+
-
*
/
< , > , = , >= , <= , !=

2. Le type Réel



- Les fonctions relatives au type Réel :

<u>Nom en algorithme</u>	<u>En Python</u>	<u>Rôle</u>	<u>Type du x</u>	<u>Type résultat</u>	<u>Exemples</u>
Arrondi(x)	round (x)	Arrondit une valeur réelle à l'entier le plus proche.	Réel	Entier	round (9.499) = round(2.35) =
abs(x) ou fabs(x)	abs(x)	Retourne la valeur absolue de x	Réel Entier	Type de x	abs(-5) = abs(-5.44) =
RacineCarré(x)	sqrt(x)	Retourne la racine carré de x (x>=0)	Entier Réel	Réel	sqrt (4) =
Aléa (x,y)	randint(x,y)	Retourne un entier entre x et y (avec x<=y)	Entier	Entier	x= randint(2,5) x = x= randint(2,10) x =

2. Le type Réel



- Exercice 1:

Essayer de remplir le tableau ci-dessous:

Instruction	Evaluation	Type du résultat
a=2.25 a=a+2.75	a= 5.0 float
a=5.0 a -=2	a = 3.0 float
a = float("12.36") b = a + 5	a = 12.36 b= 17.36 float float
a= 15.23 x= str(a)	x=..... "15.23" str
a = round(4.235) b= sqrt(a)	a=..... 4 b=..... 2.0 int float

2. Le type Réel



- Exercice 2:

Compléter le tableau ci-dessous

Description :	Instruction Algorithmique :
Affecter la valeur 2.25 à la variable x
Afficher la valeur absolue de x
Affecter la racine carre de 24 à la variable y
Affecter un entier au hasard entre 10 et 20 à z
Arrondir la valeur 3.14 1et l'affecter à la variable a

3. Le type Booléen



- Définition

Une variable de type **Booléen** peut prendre soit la valeur **True** (vrai) soit **False** (faux).

T.D.O

- Déclaration

Objet	Type/Nature
Nom_variable	Booléen

- Les opérateurs

Et (and)
Ou (or)
Non (Not)

3. Le type Booléen



Lancer Thonny et taper le code ci-dessous:

```
1 a=int(input("donner un 1er entier:" ))
2 b=int(input("donner un 2ème entier:" ))
3 plus_grand = a > b
4 print (plus_grand)
```

1ère exécution ;

```
En cours d'exécution: booléen.py
donner un 1er entier:6
donner un 2ème entier:3
True
>>> |
```

2ème exécution ;

```
En cours d'exécution: booléen.py
donner un 1er entier:2
donner un 2ème entier:10
False
>>> |
```

Ttaper le code Python ci-dessous:

```
1 x=True
2 y="True"
3 print(type(x))
4 print(type(y))
5 print(x==y)
```

Contatation ;

La variable booléenne « x » (**True**) est
.....¹¹ différente de la variable « y » (" **True**").

3. Le type Booléen



Evaluez les expressions arithmétiques suivantes :

Instruction	Evaluation
$A \leftarrow (2 < 5 \text{ OU } 15 > 24) \text{ ET } (0 < 1)$	A = True
$B \leftarrow 2 < 5 \text{ OU } (15 > 24 \text{ ET } 0 < 1)$	B = True
$C \leftarrow (\text{Arrondi}(0.05) = 0) \text{ OU } ((2 ** 2) < 2) \text{ ET } (7 \text{ DIV } 3 > 2)$	C = True

3. Le type Booléen



Soit l'affectation en « Python » suivante : `a, b, c = 6, 7, 42`
Évaluer les instructions ci-dessous en donnant à chaque fois
le résultat de Bx obtenu :

Instructions	Evaluation
<code>B1= (a == 6)</code>	B1= True
<code>B2= (a==42)</code>	B2= ... False
<code>B3= (a == 6 and b == 7)</code>	B3= True
<code>B4=(a == 7 and b == 7)</code>	B4= False
<code>B5= (not (a != 7) and (b <= 7))</code>	B5= False
<code>B6= (a == 7) or (c == 42)</code>	B6= True
<code>B7=(not (a == 7)) and (not (c != 6))</code>	B7= False



4. Le type Caractère

- Définition

- Une variable de type **caractère** contient nécessairement un caractère placé entre **deux apostrophes**.
- Un caractère peut être **une lettre** minuscule ou **majuscule**, **un chiffre** ou **un symbole**.
- Tous ces caractères sont ordonnés selon leurs codes **ASCII**.

T.D.O

- Déclaration

Objet	Type/Nature
Nom_variable	Caractère

- Les opérateurs

+
< > = ≥ ≤ !=

4. Le type caractère



- Les fonctions standards

Nom en algorithme	Nom en Python	Rôle
$a \leftarrow \text{ord}(c)$	<code>a = ord(c)</code>	Renvoie le code ASCII du caractère c
$b \leftarrow \text{chr}(a)$	<code>b = chr(a)</code>	Renvoie le caractère dont le code ASCII est a



4. Le type Chaine des caractères

- Définition

Une variable de type **Chaine des caractères** est une suite de caractères placée entre guillemets.

- Déclaration

T.D.O	
Objet	Type/Nature
Nom_variable	Chaine des Caractère

- Les opérateurs

+
< , > , = , >= , <= , !=

4. Le type caractère



- Les fonctions standards relatives au type chaîne:

Nom en algorithme	Nom en Python	Rôle
$l \leftarrow \text{long}(ch)$	<code>l=len(ch)</code>	Renvoie le nombre des caractères de ch
$ch \leftarrow \text{convch}(n)$	<code>ch=str(n)</code>	Convertir un nombre en chaîne
$p \leftarrow \text{pos}(ch1,ch)$	<code>p=ch.find(ch1)</code>	Renvoie la position de ch1 dans ch
$a \leftarrow \text{estnum}(ch)$	<code>a=ch.isdicimal()</code>	Vérifier si ch est numérique
$n \leftarrow \text{valeur}(ch)$	<code>n=int(ch)</code> <code>n=float(ch)</code>	Convertir une chaîne en entier Convertir une chaîne en réel
$ch \leftarrow \text{sous_chaîne}(ch,d,f)$	<code>ch=ch[d:f]</code>	Renvoie une partie de ch
$ch \leftarrow \text{majus}(ch)$	<code>ch=ch.upper()</code>	Convertir ch en majuscule
$ch \leftarrow \text{effacer}(ch,d,f)$	<code>ch=ch[:d]+ch[f:]</code>	Effacer une partie de ch de d(début) jusqu'au (f-1)