INF1 : Algorithmique et Programmation

Cours 2 : Instructions conditionnelles

Domitile Lourdeaux

Université de technologie de Compiègne

Printemps 2024

Hello World

IDLE Shell 3.9.1

File Edit Shell Debug Options Window Help

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello World')
Hello World
>>>
= RESTART: C:/Users/dlourdea/AppData/Local/Programs/Python/Python39/HelloWorld.py
Hello World
>>>
```

Ln: 8 Col: 4

HelloWorld.py - C:/Users/dl...

File Edit Format Run Options Window Help

```
def hello() :
    print("Hello World")


hello()
```

Ln: 6 Col: 0

File Edit View Navigate Code Refactor Run Tools VCS Win TDPytho

TDPython HelloWorld.py HelloWorld

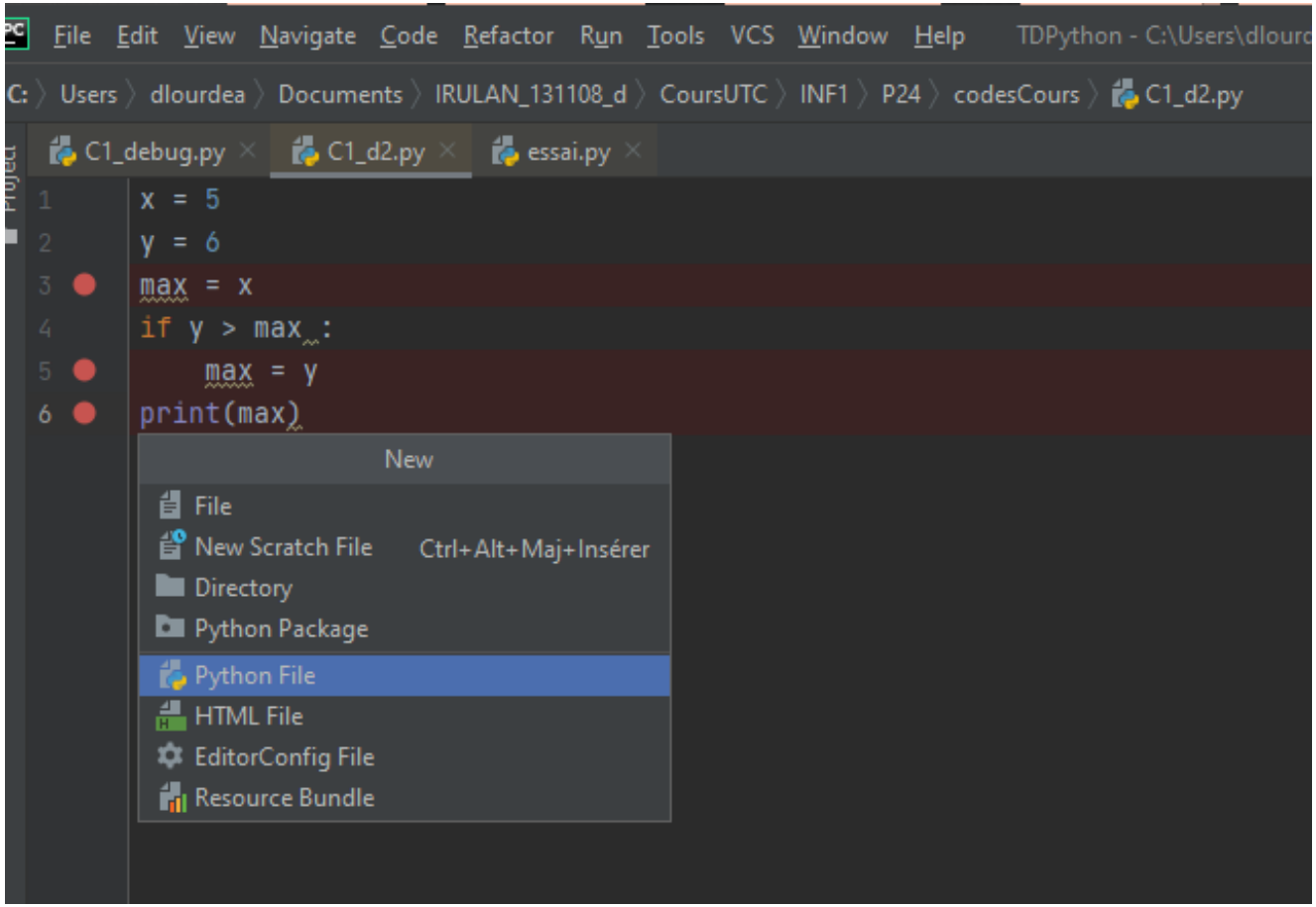
LabyrtinthelA02_v3DLo.py tdtp1.py HelloWorld.py tdtp2.py

```
def hello():
    print(f'Hello World')

hello()
```

Run: tdtp1 HelloWorld

```
C:\Users\dlourdea\Documents\IRULAN_131108_d\CoursUTC\INF1\TDPythor
Hello World

Process finished with exit code 0
```

New Python file - PyCharm



Programme : carré d'un entier

```
x = int(input("Saisir un nombre : "))
c = x * x
print("Le carre est de ce nb est : ", c)
```

Procédures et fonctions (1)

- **Une fonction ou une procédure**
  - est définie à l'aide du mot-clé def
  - a un nom
  - est suivie de parenthèses et du caractère ' :'
  - peut prendre des paramètres d'entrée ou pas (à l'intérieur des parenthèses)
  - Les instructions doivent être indentées
  - Une fonction retourne une valeur, une procédure n'en retourne pas
- **Exemple**

```
def carre (x)
    c = x * x
    return c

x = int(input("Saisir un nombre : "))
c = carre(x)
print("Le carre est de ce nb est : ", c)
```

Procédures et fonctions (2)

Fonction sans typage de variable

```
def carre (x)
    c = x * x
    return c

x = int(input("Saisir un nombre : "))
c = carre(x)
print("Le carre est de ce nb est : ", c)
```

Procédures et fonctions (3)

Fonction avec typage de variable et description

```
def carre (x : int):
    """
        fonction qui calcule le carre d'un entier
        :param int x : nombre
        :return : le carre du nombre x
    """
    # variables
    c : int

    c = x * x
    return c

x = int(input("Saisir un nombre : "))
c = carre(x)
print("Le carre est de ce nb est : ", c)
```

Identificateurs (1)

Idendificateur

Nom donné à un élément du programme (par le programmeur).
Désigne le nom donné à des entités telles que des variables, des fonctions, des classes, etc.

Attention

Les identificateurs doivent être uniques

Identificateurs (2)

Règles de formation des identificateurs en Python

- Suite alphanumérique
- Ne peut pas commencer par un chiffre, ne comporte pas d'espaces
- Peut contenir un tiret de soulignement : _
- Sensible à la « casse », ne doit pas être un mot réservé

Exemples :

Non

- ~~salaire net~~
- ~~2x~~

Oui

- salaireNet
- salaire_net
- _x
- moyenne

Conventions de nommage (1)

Bonne pratique

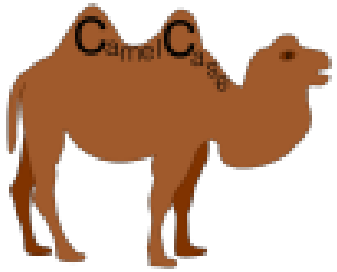
Bien d'utiliser une politique cohérente de nommage des identificateurs

Exemples

- `nom_de_ma_variable` pour les variables
- `NOM_DE_MA_CONSTANTE` pour les constantes
- `maFonction` pour les fonctions

Différentes conventions :

- LowerCamelCase : nomDeMaFonction
- UpperCamelCase : NomDeMaFonction
- Snake Case : nom_de_ma_fonction
- Kebab Case : nom-de-ma-fonction
- Flat Case : nomdemafonction



Conventions de nommage (2)

Bonne pratique

Bien d'utiliser une politique cohérente de nommage des identificateurs

Proposition pour INF1

- `nom, nom_de_ma_variable` pour les variables (Snake Case)
- `NOM_DE_MA_CONSTANTE` pour les constantes (LowerCamelCase)
- `maFonction` pour les fonctions (LowerCamelCase)

Mots réservés

Ne peuvent pas être redéfinis par le programmeur :

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Fonctions prédéfinies

Fonctions	Résultats
`print(x)`	Affiche le texte en paramètre (ici x)
`input()`	Retourne la saisie de l'utilisateur (sous forme de texte)
`len(x)`	Retourne la longueur (taille) du paramètre (ici x)
`range(n)`	Retourne la liste des entiers de 0 à n-1
`int(x)`	Convertit x en entiers
`str(x)`	Convertit x en chaine de caractères

Liste des fonctions :

`https://docs.python.org/2/library/functions.html`

Opérateurs (1)

Opérateurs arithmétiques

Symbol	Signification
+	Addition
−	Soustraction
*	Multiplication
/	Division
**	Puissance
//	Division entière
%	Modulo

Exemples

- `a = a + 1`
- `delta = b * b - 4 * a * c`
- `k = i % j`

Opérateurs (2)

Opérateurs de comparaison

Symbol	Signification
==	Egal
! =	Différent (non égal)
<	Strictement inférieur
<=	Inférieur ou égal
>	Strictement supérieur
>=	Supérieur ou égal

Exemple

```
if (i % 2 == 0 ):
    print("i est pair")
```

Opérateurs composés

On peut composer l'addition avec l'affectation

Exemple :

- x += 1 est équivalent à : $x = x + 1$

Deux opérations sont composées en une seule (incrémentation de x) :

- ajout de 1 à la valeur de x
- affectation du résultat à x

Les autres opérateurs peuvent être composés avec l'affectation de la même façon :

- $-=$, $*=$, $/=$, $\%=$, $//=$, $**=$

Documentation des programmes (1)

Commentaires

```
# Ceci est un commentaire sur une ligne

"""
Ceci est un commentaire
sur
plusieurs lignes
"""
```

Les commentaires ne sont visibles que des programmeurs

- Ils sont ignorés à l'exécution

Importance de commenter

- Se relire soi-même
- Aider les autres à comprendre son code
- Attention à l'examen...

Documentation des programmes (2)

De manière générale :

- Avoir des commentaires **cohérents** (des commentaires qui contredisent le code sont pires que pas de commentaire)
- Mettre **à jour** les commentaires quand le code change
- Faire des **phrases** complètes
- *Ecrire les commentaires en **anglais** à moins d'être sûr que le code ne sera lu que par des francophones*

Instructions de base

Entrées / Sorties

- **Affectation** de la valeur d'une expression à une variable
  - =
- **Sélection**
  - if ... else ...
- **Répétition** (prochain cours)
  - while, for

Entrées / Lecture (1)

Entrée clavier : input

```
>>> x = input("Entrez un nombre : ")
Entrez un nombre : 25
>>> x
'25'
```

Par défaut

- Le type de la valeur retournée est une chaine de caractère

Entrées / Lecture (2)

Pour lire une variable d'un autre type

- il faut « transtyper » (ou « caster » anglicisme)

Pour un entier :

```
x = input("Entrez un nombre : ")
x = int(x)
Ou directement :
x = int(input("Entrez un nombre : "))
```

Pour un réel :

```
alpha = float(input("Entrez un nombre : "))
```

Sorties / Affichage (1)

Affichage simple à l'écran

```
>>> print("Valeur du nombre : ",x)
Valeur du nombre : 25
```

Affichage avec des chaines formatées

- La chaine de caractères à afficher est précédée de la lettre f. Les variables et les expressions se notent entre accolades
- Exemples :

```
>>> print(f"Valeur du nombre : {x}")
Valeur du nombre : 25

>>> print(f"Le carré de {x} est {x**2}")
Le carré de 25 est 625
```

Affectation (1)

Syntaxe

- <identificateur de variable> = <expression>

Deux rôles :

- Evaluation (calcul) de l'expression
- Puis, affectation (rangement) à la variable (identificateur)

Exemples

- $x = 10$
- $nb\_eleves = nbEleves + 1$
- $delta = b * b - 4 * a * c$

Affectation (2)

Exemple de séquence

Exemple 1	Variables			
Instructions	a	b	q	r
$a = 19$	19	?	?	?
$b = 6$	19	6	?	?
$q = 0$	19	6	0	?
$r = a$	19	6	0	19
$r = r - b$	19	6	0	13
$q = q + 1$	19	6	1	13
$r = r - b$	19	6	1	7
$q = q + 1$	19	6	2	7
$r = r - b$	19	6	2	1
$q = q + 1$	19	6	3	1

Exemple 2	Variables		
Instructions	a	b	r
$a = 12$	12	?	?
$b = 18$	12	18	?
$r = a\%b$	12	18	12
$a = b$	18	18	12
$b = r$	18	12	12
$r = a\%b$	18	12	6
$a = b$	12	12	6
$b = r$	12	6	6
$r = a\%b$	12	6	0
$a = b$	6	6	0
$b = r$	6	0	0

D'après : `http://iriaf.univ-poitiers.fr/enibook/algorithmic/learning/site/html/affectation-0-index.html`

Sélection : structures de contrôle conditionnelles (1)

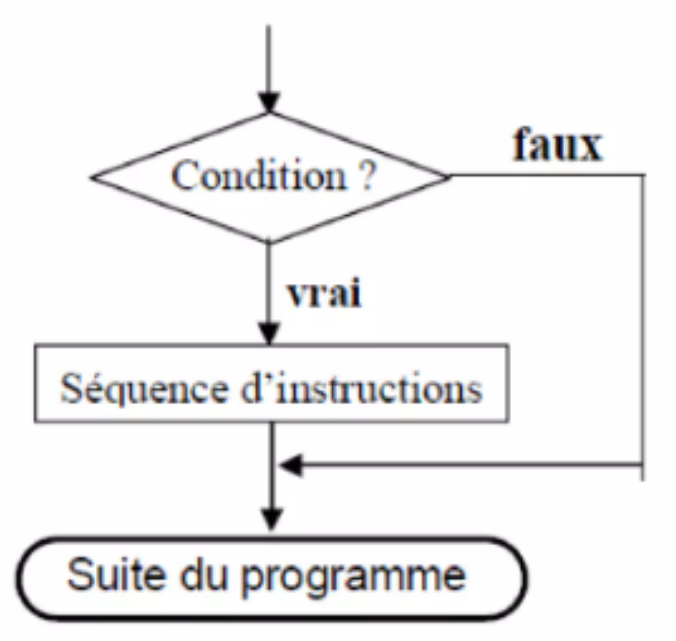
Exemple - « alternative SI... ALORS... SINON... »

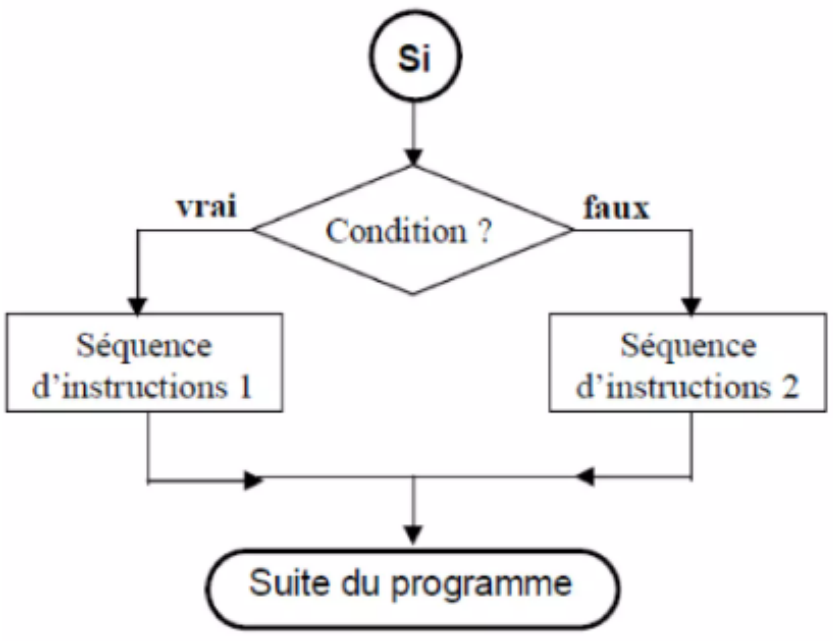
si j'ai bien tout compris INF1 alors
    je joue à Minecraft
je retravaille INF1

si j'ai bien tout compris INF1 alors
    je joue à Minecraft
sinon
    je retravaille INF1

Sélection : structures de contrôle conditionnelles (2)

Organigramme - « alternative SI... ALORS... SINON... »





D'après : `https://fr.slideshare.net/Abdouinfo/cours-algorithme-et-structure-de-donnes-1er-anne`

Sélection : structures de contrôle conditionnelles (3)

Algorithme - « alternative SI... ALORS... SINON... »

Algorithm 1 SI ... ALORS

```
// Sélection simple
si < condition(s) > alors
    < instruction(s) >
```

Algorithm 2 SI ... ALORS... SINON

...

```
// Sélection Si...Sinon
si < condition(s) > alors
    < instruction(s) >
sinon
    < instruction(s) >
```

Sélection : structures de contrôle conditionnelles (4)

Python - « alternative SI... ALORS... SINON... »

```
if  <condition(s)> :
    <bloc d'instructions>
```

```
if  <condition(s)> :
    <bloc d'instructions>
else:
    <bloc d'instructions>
```

Sélection : structures de contrôle conditionnelles (5)

Indentation - « alternative SI... ALORS... SINON... »

- Algorithme

Algorithm 3 Racine double

```
// Si delta est nul alors double racine
si delta == 0 alors
    r ← −b/(2 ∗ a)
    Afficher ("La solution est double : ", r)
```

- Code Python

```
if delta == 0 :
    r = - b / (2 * a)
    print("Une solution double : ", r)
```

Sélection : structures de contrôle conditionnelles (6)

Expressions booléennes - « alternative SI... ALORS... SINON... »

Tests conditionnels

Symbol	Signification
==	Egal
! =	Différent (non égal)
<	Strictement inférieur
<=	Inférieur ou égal
>	Strictement supérieur
>=	Supérieur ou égal

Opérateurs

- *and*, *or*, *not*, *in*

Valeurs booléennes

- *True*, *False*

Exemples

- $(x > 0)$ *and* $(x < 100)$ `# si x strictement compris entre 0 et 100`
- *not* $((x <= 0)$ *or* $(x >= 100))$ `# idem`
- y *in* $\{1, 3, 5, 7, 9\}$ `# True si y impair entre 0 et 10, False sinon`
- *bjaimelespommes* == *True*
- *bjaimelespommes*

Sélection : structures de contrôle conditionnelles (7)

Execution - « alternative SI... ALORS... SINON... »

```python
# Grand ou petit

# Attention :
#     Cet exemple comporte des biais...
#     Il ne prend en compte qu'un genre binaire (homme ou femme)
#     Les valeurs sont arbritraires et comportent des biais
#     Il conviendrait de refaire cet exemple pour plus d'ouverture

# Saisie du sexe, de l'âge et de la taille
sexe = input("Quel est votre sexe (M/F) : ")
age = int(input("Quel est votre âge : "))
taille = int(input("Quelle est votre taille : "))

# Calcul des booléens : femme, homme, majeur
femme = sexe == 'F'
homme = not femme
majeur = age >= 18

# Test conditionnel pour sélectionner si la personne est
#      petite ou grande
if femme :
    # Une femme est petite si elle mesure moins de 1 m 50
    #               grande si elle mesure plus de 1 m 70
    #               ni petite ni grande dans les autres cas
    petite = taille < 150
    grande = taille > 170
else :
    # Une homme est petit s'il mesure moins de 1 m 60
    #               grand s'il mesure plus de 1 m 80
    #               ni petit ni grand dans les autres cas
    petite = taille < 160
    grande = taille > 180

# Affichages des valeurs majeur, femme, homme, age, petite, grande
print(majeur, femme, homme)
print(age, petite, grande)
```

```
C:\Users\dlourdea\Documents\IRULAN_131108
Quel est votre sexe (M/F) : F
Quel est votre âge : 18
Quelle est votre taille : 165
True True False
18 False False

Process finished with exit code 0
```

Sélection : structures de contrôle conditionnelles (8)

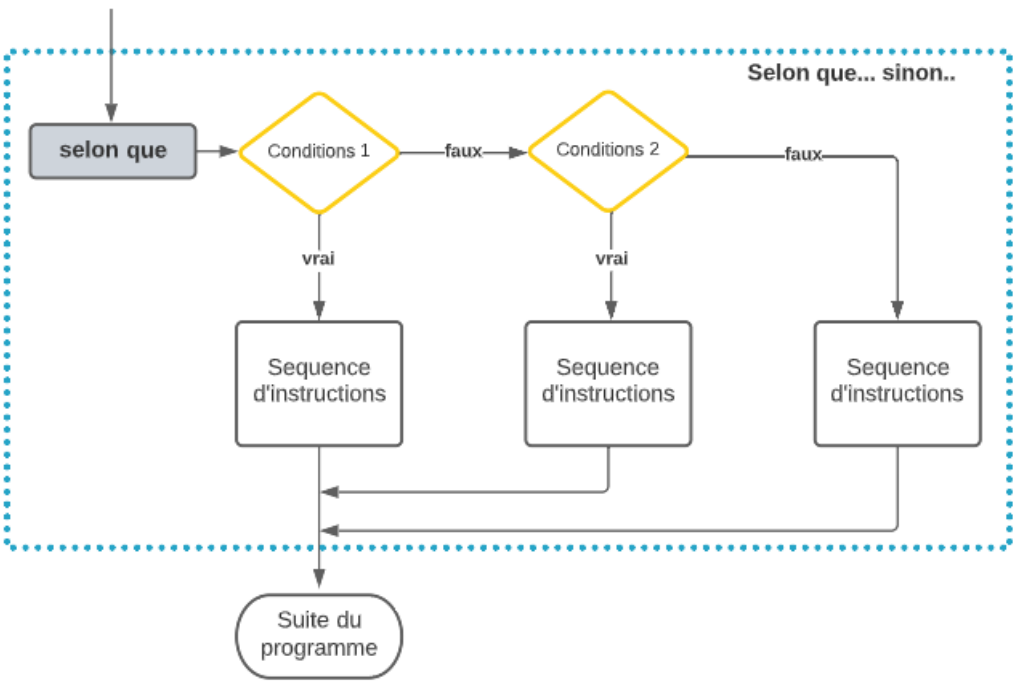
Exemple - « Choix multiples SELON QUE... »

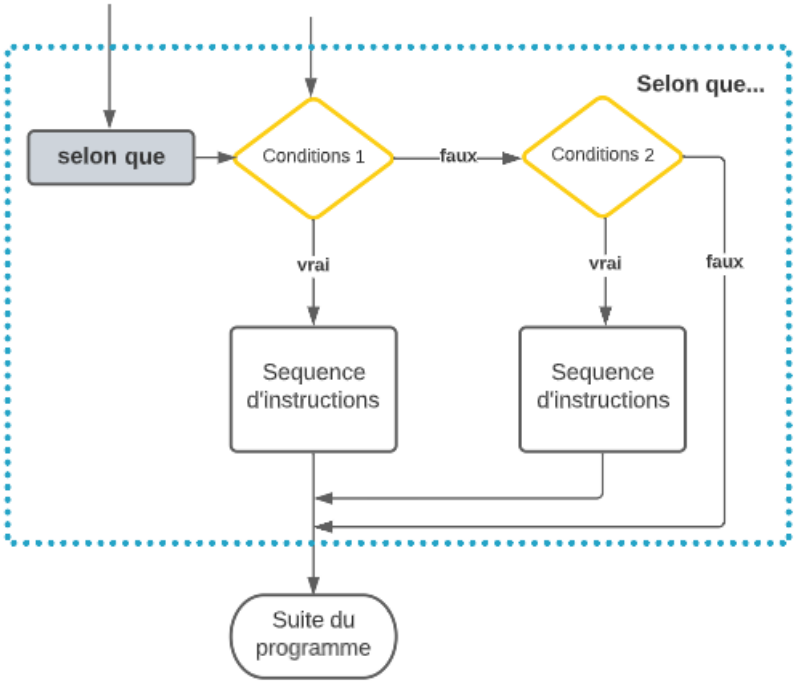
selon que

j'ai bien tout compris INF1 **alors** je joue beaucoup à Minecraft
j'ai presque tout compris INF1 **alors** je joue un peu à Minecraft et
je retravaille un peu INF1
sinon je retravaille beaucoup INF1

Sélection : structures de contrôle conditionnelles (9)

Organigramme - « Choix multiples SELON QUE... »





Sélection : structures de contrôle conditionnelles (10)

Algorithme - « Choix multiples SELON QUE... »

SELON QUE

- Raccourci pour des **Si imbriqués**
- N'existe pas sous cette forme en Python

Sélection : structures de contrôle conditionnelles (11)

Algorithme - « Choix multiples SELON QUE... »

Algorithm 4 SI... SINON SI... SINON

```
2 si < condition(s) > alors
  | < instruction(s) >
sinon
  | si < condition(s) > alors
  |   | < instruction(s) >
  | sinon
  |   | < instruction(s) >
```

Algorithm 5 SI... SINON SI...

```
2 si < condition(s) > alors
  | < instruction(s) >
sinon
  | si < condition(s) > alors
  |   | < instruction(s) >
```

Sélection : structures de contrôle conditionnelles (12)

Python - « Choix multiples SELON QUE... »

```
if  <condition(s)> :
    <bloc d'instructions>
elif <condition(s)> :
    <bloc d'instructions>
else:
    <bloc d'instructions>
```

```
if  <condition(s)> :
    <bloc d'instructions>
elif <condition(s)> :
    <bloc d'instructions>
```

Remarque

Possible de faire des *if*...*else* imbriqués

Sélection : structures de contrôle conditionnelles (13)

Exemple - « Choix multiples SELON QUE... »

Algorithm 6 Etat de l'eau

Afficher ("Temp. de l'eau")
Lire (*temperature*)
si *temperature* < 0 **alors**
  etat $\leftarrow$ "*glace*"
sinon
  si *temperature* < 100 **alors**
    etat $\leftarrow$ "*liquide*"
  sinon
    etat $\leftarrow$ "*vapeur*"
Afficher ("Etat : ", etat)

```
# Calcul de l'état de l'eau à partir de sa température
temperature = int(input("Quelle est la température de l ea
if temperature < 0 :
    # Etat de glace (t < 0)
    etat = 'glace'
elif temperature < 100 :
    # Etat liquide (0 < t < 100)
    etat = 'liquide'
else :
    # Etat de vapeur (t >= 0)
    etat = 'vapeur'

print("Etat = ", etat)
```

Sélection : structures de contrôle conditionnelles (14)

On peut également utiliser :

Opérateur ternaire

Il arrive souvent qu'il y ait deux résultats simples possibles après le test d'une condition. Il est dans ce cas un peu lourd d'utiliser la structure **if ...: else: ...**. Cela peut être écrit en une ligne selon

```
res = restrue if test_condition else resfalse
# exemple: minimum entre x et y
minimum = x if x<y else y
# ou même définition de la fonction Min avec lambda
Min = lambda x,y: x if x<y else y
Min(10,2)
```

D'après http://lptms.u-psud.fr/wiki-cours/index.php/Python:_Tests_conditionnels

Jeu 9 des erreurs

```
# Argent de poche
a = input("Quel est l age de l'enfant ? ")

elif a < 10
    argent_poche = 10
elif a < 7
    argent_poche = 5
elif a < 15
    argent_poche = 20
elif a < 18
    argent_poche = 50
else
    argent_poche = 100
    print(argent_poche)
```

A vous de jouer

Exercices possibles

- Vérifier si un mot de passe a suffisamment de caractères
  - 8 caractères au minimum
  - Mieux si plus de 12
  - Fonction Python pour calculer la longeur d'une chaine *len*(*str*)
- Faire une calculatrice
  - 2 opérandes
  - 1 opérateur (+, -, *, /)

Mot de passe

Algorithm 7 Longueur Mot de passe

variables

mot_de_passe : *chainedecaracteres*

longueur_mdp : *entier*

Afficher ("Saisir du mot de passe : ")

Lire (*mot_de_passe*)

longueur_mdp ← *longueur*(*mot_de_passe*)

```
si longueur_mdp < 8 alors
 |  Afficher ("Trop court")
sinon
 |  si longueur_mdp < 12 alors
 |   |  Afficher ("Moyen")
 |  sinon
 |   |_ Afficher ("Parfait")
 |_
```

Mot de passe

Python

```python
# Calcul de la longueur d'un mot de passe
mot_de_passe = input("Saisir un mot de passe : ")
longueur_mdp = len(mot_de_passe)

if longueur_mdp < 8 :
    print("Longueur trop courte")
elif longueur_mdp < 12 :
    print("Longueur moyenne")
else :
    print("Longueur parfaite")
```

Calculatrice

Algorithm 8 Calculatrice

```
variables
        x, y, res : reels
        op : caractere
Afficher ("Saisir 2 opérandes et 1 opérateur : ")
Lire (x, y, op)
si op == " + " alors
 |   res ← x + y
sinon
 |   si op == " − " alors
 |    |   res ← x − y
 |   sinon
 |    |   si op == " ∗ " alors
 |    |    |   res ← x ∗ y
 |    |   sinon
 |    |    |_  res ← x/y
 |_
Afficher ("Résultat : ", res)
```

Calculatrice

Python

```python
# Calculatrice
x = float(input("Saisir opérande 1 : "))
y = float(input("Saisir opérande 2 : "))
op = input("Sasir l'opérateur : ")

if op == '+' :
    res = x + y
elif op == '-' :
    res = x - y
elif op == '*' :
    res = x * y
else :
    res = x / y

print("Résultat : ", res)
```

Sondage Moodle

Mes cours/INF1/Cours 2 : Sondage « Je me situe » (2)

Questions...