Information, Calcul et Communication

Partie Programmation

Cours 2: Structures de contrôle

29.09.2023

Patrick Wang

Au sommaire cette semaine

1. Quelques éléments de l'algèbre de Boole
2. Structures conditionnelles
3. Structures itératives

Au sommaire cette semaine

1. Quelques éléments de l'algèbre de Boole
2. Structures conditionnelles
3. Structures itératives

1. Quelques éléments de l'algèbre de Boole

Retour sur un exemple vu la semaine dernière

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

1. Quelques éléments de l'algèbre de Boole

Retour sur un exemple vu la semaine dernière

```
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

1. Quelques éléments de l'algèbre de Boole

Nouveau type: `bool`

```
my_var: str = "un bout de texte"
test: bool = my_var.endswith("xte")
print(test) # Affiche True
```

- Une variable booléenne vaut soit `True` soit `False`

2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

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

1. Quelques éléments de l'algèbre de Boole

Opérateurs booléens, tables de vérité

- En Python, il existe trois opérateurs booléens :
  - `and`
  - `or`
  - `not`

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

A	not A
0	1
1	0

1. Quelques éléments de l'algèbre de Boole

Arrêt sur le vocabulaire

- Déjà 5 mots du vocabulaire de Python !

2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

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

1. Quelques éléments de l'algèbre de Boole

Quelques exemples...

```
exemple1: bool = True or False
exemple2: bool = True and False
exemple3: bool = not False

plus_dur: bool = False or not False and True
```

1. Quelques éléments de l'algèbre de Boole

Autres opérations résultant en une valeur booléenne

```
# Est-ce que 0.1 + 0.2 est égal à 0.3 ?
print(0.1 + 0.2 == 0.3)

# Est-ce que 4.0 est différent de 4 ?
print(4.0 != 4)

# Est-ce que "bonjour" est strictement inférieur à "Bonjour" ?!!
print("bonjour" < "Bonjour")

# Est-ce que "bonjour" est supérieur ou égal à "bonsoir" ?!!
print("bonjour" >= "bonsoir")
```

Au sommaire cette semaine

1. Quelques éléments de l'algèbre de Boole
2. Structures conditionnelles
3. Structures itératives

2. Structures conditionnelles

Retour sur l'exercice 5 de la série 1

```python
day: int = 22
month: str = "09"
year: int = 2023
print(f"Aujourd'hui, nous sommes le {day}.{month}.{year}")
print(f"Le prochain cours d'ICC aura lieu le {day+7}.{month}.{year}")
```

- Ajouter 7 à `day` fonctionne bien entre le premier et le deuxième cours...
- ... mais plus entre le deuxième et le troisième cours à cause du changement de mois
- En fonction des valeurs de `day` et de `month`, le code à exécuter sera différent !

2. Structures conditionnelles

Retour sur l'exercice 5 de la série 1

- Comment construire un programme qui soit capable de calculer la date du prochain cours ICC en tenant bien compte des changements de mois ?

  Si day + 7 > nb_days_in_month, alors:
  month <— month + 1
  day <— (day + 7) - nb_days_in_month
  Sinon: day <— day + 7

- Reste à savoir comment déterminer nb_days_in_month

2. Structures conditionnelles

Retour sur l'exercice 5 de la série 1

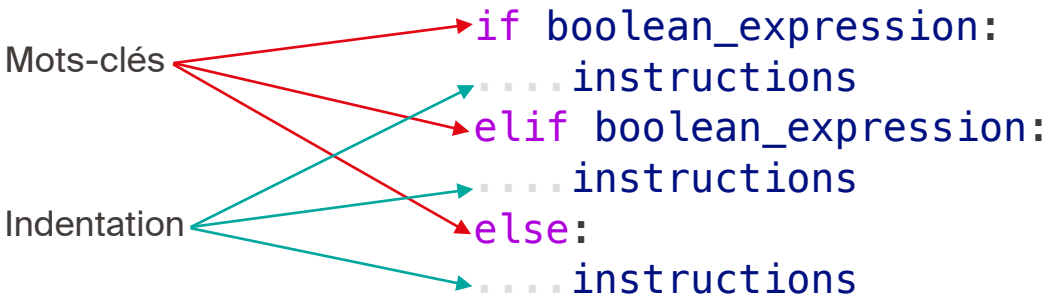
```
if month in (1, 3, 5, 7, 8, 10, 12):
    nb_days_in_month = 31
elif month in (4, 6, 9, 11):
    nb_days_in_month = 30
else:
    nb_days_in_month = 28

if day + 7 > nb_days_in_month:
    month += 1
    day = (day + 7) - nb_days_in_month
else:
    day += 7

print(f"Le prochain cours ICC aura lieu le {day}.{month}.{year}.")
```

2. Structures conditionnelles

Arrêt sur la syntaxe

```
if month in (1, 3, 5, 7, 8, 10, 12):
    nb_days_in_month = 31
elif month in (4, 6, 9, 11):
    nb_days_in_month = 30
else:
    nb_days_in_month = 28
```

Sous-entendu
« Il est vrai que... »



2. Structures conditionnelles

Arrêt sur le vocabulaire

- Déjà 9 mots du vocabulaire !

2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

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

Au sommaire cette semaine

1. Quelques éléments de l'algèbre de Boole
2. Structures conditionnelles
3. Structures itératives

3. Structures itératives

Retour sur l'exercice 5 de la série 1

- Comment compléter notre programme pour qu'il affiche toutes les dates du cours ICC, en ne connaissant que la première date ?

```
Répéter 14 fois:
    Déterminer nb_days_in_month
    Si day + 7 > nb_days_in_month, alors:
        month <— month + 1
        day <— (day + 7) - nb_days_in_month
    Sinon: day <— day + 7
```

- Il ne reste plus qu'à afficher le message (et à faire attention à l'afficher au bon moment)

3. Structures itératives

Boucle `for`

```
for i in range(14):
    print(f"Le cours no. {course_number + i} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
```

3. Structures itératives

Boucle `for`

```
for i in range(14):
    print(f"Le cours no. {course_number + i} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
```

3. Structures itératives

Boucle `for`

```
for i in range(14):
    print(f"Le cours no. {course_number + i} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
```

3. Structures itératives

Boucle `for`

```
for i in range(0, 14, 1):
    print(f"Le cours no. {course_number + i} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
```

- La fonction `range()` va être très utile pour les boucles `for`
- `range(stop)`
- `range(start, stop)`
- `range(start, stop, step)`

3. Structures itératives

Boucle `while`

```
course_number: int = 1
while course_number <= 14:
    print(f"Le cours no. {course_number} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
    course_number += 1
```

3. Structures itératives

Boucle `while`

```
course_number: int = 1
while course_number <= 14:
    print(f"Le cours no. {course_number} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
    course_number += 1
```

3. Structures itératives

Boucle `while`

```
course_number: int = 1
while course_number <= 14:
    print(f"Le cours no. {course_number} aura lieu le {day}.{month}.{year}")
    if month in (1, 3, 5, 7, 8, 10, 12):
        nb_days_in_month = 31
    elif month in (4, 6, 9, 11):
        nb_days_in_month = 30
    else:
        nb_days_in_month = 28
    if day + 7 > nb_days_in_month:
        day = (day + 7) - nb_days_in_month
        month += 1
    else:
        day += 7
    course_number += 1
```

3. Structures itératives

Boule `for` ou `while` ?

- Il existe deux types de boucles. Quelles sont les différences ? Comment savoir quand utiliser plutôt l'une que l'autre ?
- Boucle for:
  - Utile pour parcourir un «objet itérable» (`str`, `range()`, ...)
  - On sait d'avance combien d'itérations seront réalisées
- Boucle while:
  - Utile si une condition doit être vérifiée pour répéter les instructions
  - Peut faire le même travail qu'une boucle for
  - Peut également construire des boucles infinies

3. Structures itératives

Arrêt sur la syntaxe

```
for loop_variable in iterable:
    instructions
```

```
while boolean_expression:
    instructions
```

3. Structures itératives

Arrêt sur le vocabulaire

2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

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

11 sur 35 déjà ! Et en plus, on ne va pas tous les rencontrer cette année !

Bilan

Ce que l'on vient de voir en cours

- Les expressions booléennes
- Les structures de contrôle :
  - Structures conditionnelles (`if` – `elif` – `else`)
  - Structures itératives (`for` – `while`)
- Quelques fonctions utiles:
  - `input()`
  - `int()`
  - `range()`

Bilan

Ce que vous allez faire en séance d'exercices

- Créer des programmes interactifs grâce à `input()`
- Réaliser quelques conversions de types
- Lire des programmes utilisant des structures de contrôle
- Écrire des programmes utilisant des structures de contrôle

- ***Attention***: Il y a beaucoup d'exercices. Il ne s'agit pas des les finir durant la séance d'exercices, mais de travailler dessus tout au long de la semaine.