

Objectifs

- Savoir exploiter les principes de logique dans les structures itératives
- Savoir analyser une expression logique

1 Itérations

Autant que possible, n'exécutez pas immédiatement le code Python de cet exercice, il s'agit de réfléchir "sur papier" avant de lancer. Le notebook avec le code est sur le Moodle.

1. A quel moment une boucle s'interrompt-elle ?
2. Dans l'exemple suivant, à partir de quand la condition d'entrée devient fausse ?

```
i = 2
while i <= 8:
    print("coucou")
    print(i)
    i += 1
print("Fin")
```

3. La finalité de l'exemple ci-dessous est-elle similaire à l'exemple précédent ?

```
for i in range(2, 9):
    print("coucou")
    print(i)
    i += 1
print("Fin")
```

4. Analysez les différentes propositions de l'exemple ci-dessous et indiquez quelle est la condition de sortie de boucle.

```
i = 5
encore = True
while encore == True:
    print("coucou")
    i += 1
    if i > 10:
        encore = False
print("Fin")
```

5. Étudiez la proposition logique utilisée dans cet algorithme. Que va faire le code ?

```
if True:
    print("c'est vrai")
```

6. Et dans ce cas ?

```
if False:
    print("c'est faux")
```

7. Que va faire ce code ?

```
while True:
    print("c'est vrai")
```

8. L'algorithme ci-dessous permet-il d'écrire "bonjour" ?

```
A = True
B = False
if A and B:
    print("bonjour")
```

9. Dans l'exemple ci-dessous, pour quelle valeur de i pourrons-nous sortir de la boucle ?

```
A = True
B = False
i = 0
while A or B:
    print("bonjour")
    if i > 10 and i < 20:
        A = False
    i += 1
```

10. Étudiez le comportement de l'algorithme suivant. Peut-on rentrer dans la boucle ? Quand sortirons nous de la boucle ?

```
A = True
B = True
i = 0
while A or B:
    print("bonjour")
    A = False
```

11. Quelle différence avec celui-ci (NB: ^ est le ou exclusif ou "XOR")?

```
A = True
B = True
i = 0
while A ^ B:
    print("bonjour")
    A = False
```

2 Apprendre les opérateurs, compléter des tables de vérité

2.1 Négation logique \neg

A	$\neg A$
VRAI	
FAUX	

2.2 ET logique (\wedge), OU logique (\vee)

A	B	$A \wedge B$	A	B	$A \vee B$
VRAI	VRAI		VRAI	VRAI	
VRAI	FAUX		VRAI	FAUX	
FAUX	VRAI		FAUX	VRAI	
FAUX	FAUX		FAUX	FAUX	

2.3 XOR (OU-exclusif, \oplus), Implication (\Rightarrow)

A	B	$A \oplus B$	A	B	$A \Rightarrow B$
VRAI	VRAI		VRAI	VRAI	
VRAI	FAUX		VRAI	FAUX	
FAUX	VRAI		FAUX	VRAI	
FAUX	FAUX		FAUX	FAUX	

2.4 Compléter des tables de vérité complexes

A	B	$\neg A \vee \neg B$
VRAI	VRAI	
VRAI	FAUX	
FAUX	VRAI	
FAUX	FAUX	

A	B	$A \wedge \neg B$	$B \vee \neg A$	$(A \wedge \neg B) \vee (B \vee \neg A)$
VRAI	VRAI			
VRAI	FAUX			
FAUX	VRAI			
FAUX	FAUX			

A	B	C	$(A \wedge B)$	$\neg C$	$(A \wedge B) \vee \neg C$
VRAI	$\neg A$	$\neg B$			
VRAI	FAUX	$\neg A$			
FAUX	VRAI	$\neg B$			
FAUX	FAUX	$\neg B$			