Plan du chapitre

Une instruction conditionnelle porte bien son nom : elle ne sera réalisée que si une certaine condition est respectée, condition que l'on vérifiera grâce à un test dont la valeur est booléenne.

I Structure d'une instruction conditionnelle

La structure d'une instruction conditionnelle est de l'une des trois formes suivantes :

```
if <condition> :
        <instr 1>
        ...
        <instr n>
<ensuite>
```

```
if <condition> :
        <instr 1>
        ...
else :
        <instr 1>
        ...
<ensuite>
```

```
if <condition1> :
        <instr 1>
        ...
elif <condition2> :
        <instr 1>
        ...
elif <condition n> :
        <instr 1>
        ...
else :
        <instr 1>
        ...
<ensuite>
```

Analysons ces structures en détails :

- la première ligne d'une instruction conditionnelle est toujours la même : le mot-clé **if**, suivi d'un **test conditionnel dont la valeur est booléenne**, puis d'un **double point**. Si le test renvoie le booléen **True**, alors les instructions situées aux lignes suivantes sont réalisées. En revanche, si le résultat du test est **False**, les instructions ne sont pas réalisées.

- Les lignes suivantes, contenant les instructions à réaliser en cas de test positif, doivent nécessairement être **indentées**, c'est à dire mise en retrait du même nombre d'espaces. En général, lorsque l'on travaille avec un IDE, comme ce sera toujours notre cas, cette indentation est automatique, mais si vous tapez votre programme avec un éditeur de texte simple, il faut impérativement y penser.

- Une fois la première série d'instructions écrite, il est **possible**, dans le cas où la première condition n'a pas été réalisée, de poser une seconde condition, ayant pour conséquence une seconde série d'instructions : on utilise pour cela le mot-clé **elif**, la structure de l'instruction étant alors la même que pour **if**. Ce mot-clé est la contraction des mots **if** (si) et **else** (sinon) que nous pourrions traduire par « sinon si ». Il est possible d'écrire ainsi un nombre

quelconque de conditions et d'instructions associées, en recommençant chaque nouvelle condition par ce mot-clé **elif**. **Attention** : à chaque nouvelle condition, commençant par le mot-clé **elif**, l'indentation doit être remise au même niveau que le **if** de début d'instruction.

- Enfin, lorsque toutes les conditions ont été posées, il est **possible** d'écrire des instructions à réaliser dans le cas où aucune des conditions précédentes n'a donné de résultat positif. Ces dernières instructions sont précédées du mot-clé **else**, suivi d'un double-point, et, comme dans les cas précédents, les instructions sont indentées.

Attention danger !

Il est important de noter que, dès qu'une condition est réalisée, son bloc d'instructions associé est interprété, et **l'instruction conditionnelle s'arrête là** : Python passe alors directement à la suite <ensuite>. **En conséquence, au plus une des conditions est évaluée à True, et au plus un bloc d'instructions est exécuté.**

Entrainement 1

Voici trois exemples d'instructions conditionnelles. Que se passe t-il dans les trois exemples quand j'ai colle de Physique ET de Maths le lendemain? Et quand je n'ai aucune colle le lendemain? On fera l'hypothèse que les conditions suffisantes énoncées sont nécessaires...

EXEMPLE 1

si <j'ai colle de Physique demain> :
    <je travaille ce soir mon cours de Physique>
sinon :
    <je travaille ce soir mon cours de Maths>

EXEMPLE 2

si <j'ai colle de Physique demain> :
    <je travaille ce soir mon cours de Physique>
sinon si <j'ai colle de Maths demain> :
    <je travaille ce soir mon cours de Maths>

EXEMPLE 3

si <j'ai colle de Physique demain> :
    <je travaille ce soir mon cours de Physique>
si <j'ai colle de Maths demain> :
    <je travaille ce soir mon cours de Maths>

II Exercices corrigés

Entrainement 2 : test de parité

Ecrire un script qui demande à l'utilisateur un entier n et qui affiche si cet entier est pair ou impair. Le résultat devra apparaître sous la forme « l'entier n est pair » ou « l'entier n est impair ».

```
texte=input('Un entier svp?')
entier=int(texte)
if entier%2==0:
    print("L'entier ",entier," est pair.")
else:
    print("L'entier ",entier," est impair.")
```

Entrainement 3 : une année est-elle bissextile?

Une année tropique (c'est-à-dire une révolution complète de la Terre autour du soleil) dure un peu moins de 365,25 jours. Pour compenser ce décalage de presque un quart de jour qui se produit chaque année entre notre calendrier de 365 jours et l'année tropique, il a été décidé d'ajouter un jour supplémentaire certaines années, dites *bissextiles*.

La règle définissant les années bissextiles est assez simple : une année est bissextile lorsque

- elle est divisible par 4 et non par 100, ou
- elle est divisible par 400.

Par exemple, les années 2000 et 2016 sont bissextiles, mais l'année 1900 ne l'est pas.

Ecrire un script qui demande à l'utilisateur une année et qui affiche si cette année est bissextile ou non.

```
annee=int(input("Une␣année␣svp?"))
if (annee%4==0 and annee%100!=0) or (annee%400==0):
    print("L'année␣",annee,"␣est␣bissextile.")
else:
    print("L'année␣",annee,"␣n'est␣pas␣bissextile.")
```

Entrainement 4 : tous au casino!

Nous allons, dans cet exercice, simuler une roulette de casino. Les règles sont simples :

- ▷ le programme doit d'abord demander à l'utilisateur combien il mise, puis sur quel numéro, compris entre 0 et 49, il met son argent.
- ▷ Ensuite, grâce à la fonction **randint** de la bibliothèque **random** (pas à connaître), le programme simule le lancer de la roulette en choisissant aléatoirement un nombre entre 0 et 49.
- ▷ Si le nombre choisi est le bon, le joueur regagne 5 fois sa mise. Si seule la couleur est la même (nombres pairs : rouge, nombres impairs : noir), le joueur regagne 1,5 fois sa mise. Dans les autres cas, le joueur a perdu.
- ▷ Le programme doit afficher à la fin le résultat de la roulette et l'argent éventuellement gagné.

```
mise=float(input("Combien␣misez-vous?"))
numero=int(input("Sur␣quel␣numéro␣misez-vous?"))

from random import randint
tirage=randint(0,49)
```

```
6
7  print("Le numéro ",tirage," est sorti")
8  if numero==tirage:
9      print("Vous avez le bon numéro!")
10     print("Vous remportez ",5*mise,".")
11 elif(numero-tirage)%2==0:
12     print("Vous avez la bonne couleur!")
13     print("Vous remportez ",1.5*mise,".")
14 else:
15     print("Vous avez perdu!")
```

Entrainement 5 : limitation de vitesse

Ecrire un programme le plus concis possible qui, après avoir demandé à quelle vitesse roule un conducteur, et sur quel type de voie (parmi « ville », « nationale », et « autoroute »), lui indique si sa vitesse est correcte ou s'il mérite une sanction.

Les données du code de la route utiles à cet exercice sont rassemblées dans le tableau suivant :

Voie	Vitesse maximum ($km.h^{-1}$)
Autoroute	130
Nationale	80
Ville	50

```
1 voie=input("Quelle voie?")
2 vitesse=float(input("Quelle vitesse en km/h?"))
3
4 if (voie=="Autoroute" and vitesse>130) or (voie=="Nationale" and vitesse>90) or (voie=="Ville" and
      vitesse>50):
5     print("Vous méritez une amende...")
6 else:
7     print("Vitesse OK")
```