T.P. chapitre 5 : la boucle Tant que : `while`

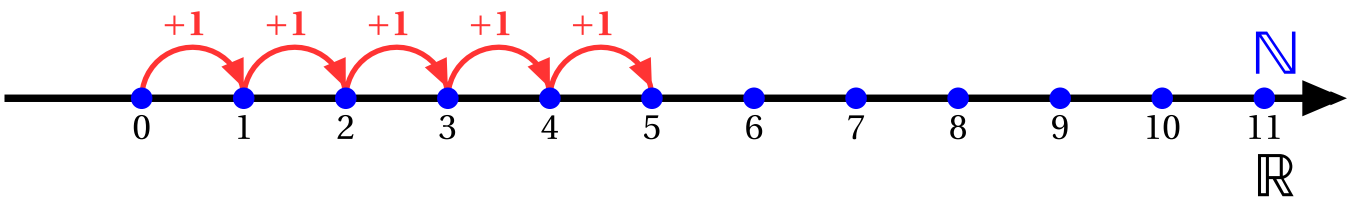
1 La boucle while

L'une des tâches que les ordinateurs font le mieux est la répétition de tâches identiques. Il existe plusieurs méthodes pour programmer ces tâches répétitives, notamment la boucle Tant que.

Exercice 1 : Promenade sur les entiers

La boucle "Tant que" commence à l'aide de l'instruction `while` :

```
nombre = 0
while nombre <= 5 :
    print(nombre)
    nombre = nombre + 1
```



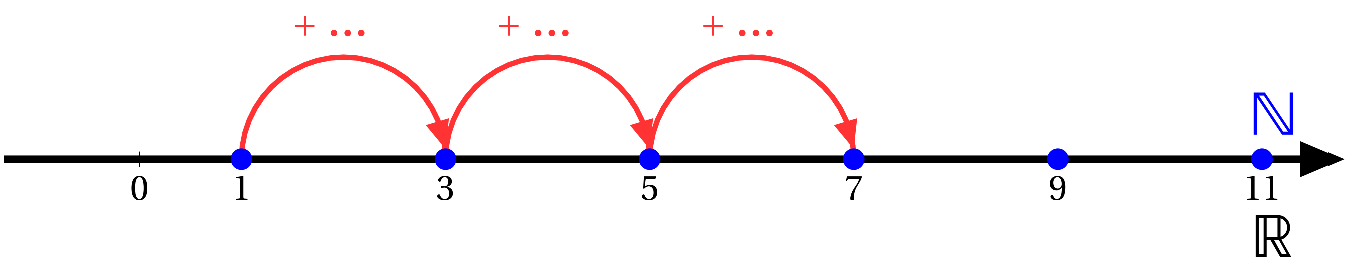
a. Prévoir le résultat du programme ci-dessus.

b. Recopier ce programme dans Python et vérifier votre réponse de la question précédente.

c. Modifier le programme pour qu'il affiche successivement les nombres : 0, 1, 2, 3, . . . , 9, 10.

Exercice 2 : Promenade impaire

a. Nous souhaiterions maintenant parcourir les nombres impairs de 1 à 41.
Compléter le schéma suivant :



b. Écrire un programme qui affichera successivement les nombres impairs de 1 à 41 :
1, 3, 5, 7, 9, . . . , 41.

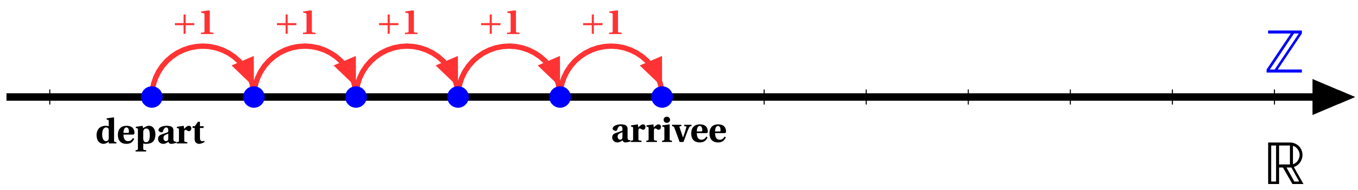
Exercice 3 : Fonction de promenade entre deux entiers

À chaque nouvelle suite de nombres, nous avons ré-écrit le même programme (à quelques variantes près).

Afin d'éviter de ré-écrire le même programme à chaque nouvelle suite de nombres, nous pouvons penser à utiliser … une fonction !

a. Écrire une fonction **promenade(depart, arrivee)** :

- prenant en argument deux nombres entiers : `depart` et `arrivee`,
- et affichant tous les entiers de `depart` à `arrivee`.




```
def promenade(depart, arrivee):
    nombre = depart
    while .............:
        .................................................................
        .................................................................
```

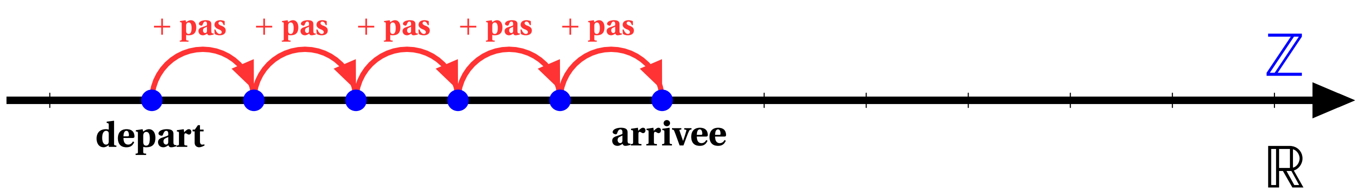
b. Tester votre fonction avec différentes valeurs.

Exercice 4 : Avec des pas de longueur variable

Nous souhaiterions maintenant pouvoir faire des pas de longueur variable (par exemple de longueur 2 pour parcourir des suites de nombres pairs ou impairs).

a. Écrire une fonction **promenade_2(depart, arrivee, pas)** :

- prenant en argument trois nombres entiers : `depart`, `arrivee` et `pas`,
- et affichant les entiers de `depart` à `arrivee`, en faisant des pas de longueur `pas`.



b. Tester votre fonction avec différentes valeurs.

c. Utiliser votre fonction pour afficher :

- les nombres pairs de 8 à 96,
- les nombres impairs de 133 à 151,
- les multiples de 7 de 7 à 78,
- les multiples de 3 de 3 à 300,
- les multiples de 5 de 5 à 100.

Exercice 5 : quiz

Réaliser le QCM : http://algoprog.fr/06-qcm/qcm.php?contenu=6&titre=La boucle while 1

2 Problèmes de seuil

Exercice 6 : menuiserie

Une menuiserie produit et installe des escaliers en bois. Nous estimerons que le bénéfice mensuel en euros, réalisé lors de la fabrication de x escaliers, est donné par la fonction :

$$f \text{ définie sur } [0\,;\,+\infty[\text{ par : } f(x) = 4000x + 2000$$

❑ Fonction f

a. Écrire en Python une fonction `f(x)` :
- prennant en argument un nombre x,
- et renvoyant l'image de x par la fonction f.

```
def f(x):
    ...................................................................
```

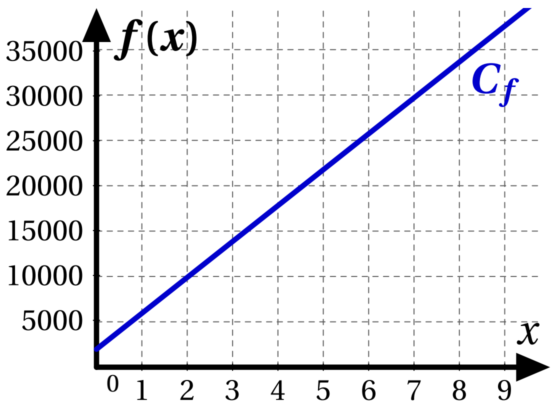
❑ Fonction de seuil

On donne ci-dessous le graphique de la fonction f sur $[0\,;\,10]$.

b. Résoudre graphiquement l'inéquation :

$$f(x) \geqslant 30\,000$$

....



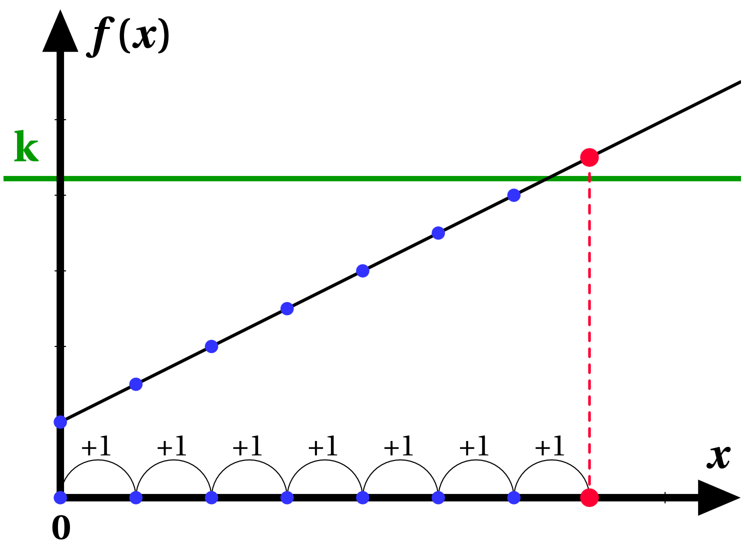
c. Quelle conclusion en tirer pour la menuiserie ?

....

....

d. Quel est le rôle de cette fonction ?

Fonction seuil(k) :
    $x = 0$
    Tant que $f(x) < k$:
        $x \leftarrow x + 1$
    Retourner x



....

....

e. Écrire en Python la fonction de seuil de la question précédente.

```
def f(x):
    ..............................................................

def seuil(k):
    x = 0
    while .............:
        x = ..............................................................
    return ...............................................................
```

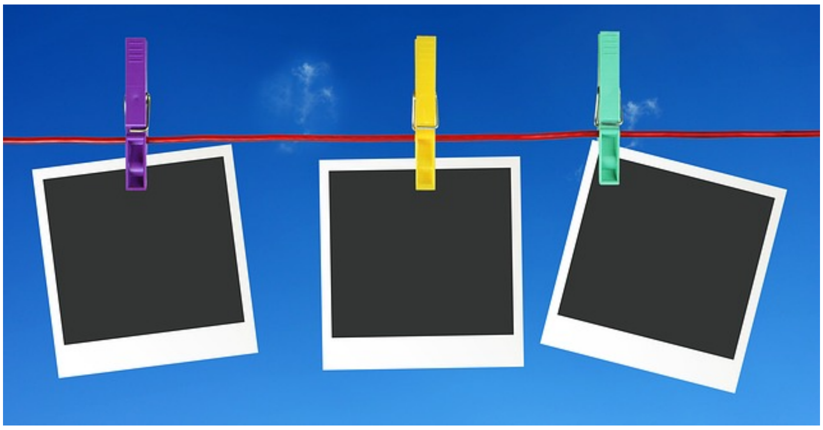
f. Combien d'escaliers la menuiserie doit-elle produire et installer dans le mois pour voir son bénéfice mensuel dépasser 100 000 euros ?

....

Exercice 7 : quiz

Réaliser le QCM : https://www.algoprog.fr/06-qcm/qcm.php?contenu=7&titre=La%20boucle%20while%202

Exercice 8 : un problème d'impression de photos



Un site d'impression de photos en ligne propose deux types de tarifs :

- tarif 1 : 0,5 € par photo,
- tarif 2 : un abonnement de 20 €, puis 0,35 € par photo.

Problème : déterminer à partir de combien de photos il est intéressant de prendre l'abonnement.

Résolution informatique

1. **a.** Écrire une fonction `tarif_1(x)` :

- prenant en argument le nombre x de photos à imprimer,
- renvoyant le prix à payer par la première formule.

b. Utiliser votre fonction pour déterminer le prix d'impression de 65 photos par la formule 1.

2. **a.** Écrire une fonction `tarif_2(x)` :

- prenant en argument le nombre x de photos à imprimer,
- renvoyant le prix à payer par la seconde formule.

b. Utiliser votre fonction pour déterminer le prix d'impression 65 photos par la formule 2.

3. Écrire une fonction `seuil()` :

- faisant appel aux fonctions des questions **1.** et **2.**,
- renvoyant le nombre de photos à partir duquel il est intéressant de prendre l'abonnement.

```
def tarif_1(x):
    ......................................................................

def tarif_2(x):
    ......................................................................

def seuil():
    x = 1
    while ...........................:
        x = ..............................................................
    return ...............................................................
```

4. À partir de combien de photos, l'abonnement est-il intéressant ?

Résolution mathématique

Résoudre maintenant le problème posé à l'aide d'une méthode **mathématique**.

3 Modélisation d'une propagation épidémique

Exercice 9 : modélisation d'une épidémie
Considérons une population humaine ou animale.

- au jour 0, trois individus sont contaminés par le virus d'une épidémie,
- chaque jour, le nombre d'individus contaminés triple.

Problème : si la population compte `n` individus, combien faut-il de jours pour que toute la population soit contaminée ?

a. Écrire une fonction Python :
- prenant en argument l'effectif de la population,
- et renvoyant le premier jour où toute la population sera contaminée.

```
def epidemie(n):
    jour = 0
    malade = 3
    while ..............:
        jour = .............................................................
        malade = ...........................................................
    return .................................................................
```

b. Utiliser la fonction pour déterminer au bout de combien de jours une population d'un million d'individus sera entièrement contaminée.

....

c. Notre modèle (→ triple tous les jours), est très simplifié. Mais il a déjà une caractéristique très nette : ce modèle vous semble-t-il correspondre à une épidémie se propageant rapidement ou lentement ?

....

d. Proposer en langage naturel un algorithme correspondant à une épidémie où le nombre d'individus contaminés augmente de 5% tous les jours.

....

....

....

....

....

....

....

Ronald Ross : paludisme et modélisation

Ronald Ross né en Inde en 1857 et meurt à Londres en 1932. Il est britannique, médecin bactériologiste et entomologiste de l'Armée des Indes britanniques. Il a reçu le prix Nobel de physiologie ou médecine pour ses travaux sur le paludisme.

Ses travaux établissent que la transmission du paludisme des oiseaux se fait par un moustique. Il estime par ailleurs, par la modélisation, le seuil de moustiques à éradiquer pour éradiquer le paludisme.

Exercice 10 : propagation d'une rumeur

Imaginons un modèle très simplifié de propagation d'une rumeur au sein d'une population :

- au jour 1, un individu était au courant de cette rumeur,
- chaque jour, 3 nouveaux individus apprennent cette rumeur.

Problème : si la population compte `n` individus, combien faut-il de jours pour que toute la population soit au courant de la rumeur ?

a. Écrire une fonction `rumeur(n)` :

- prenant en argument l'effectif total n de la population,
- renvoyant le premier jour où toute la population connaîtra cette rumeur.

```
def rumeur(n):
    jour = ...
    individu = ...
    while ..............:
        ....................................................................
        ....................................................................
    ........................................................................
```

b. Au bout de combien de jours une population de 16 000 individus sera intégralement atteinte par la rumeur ?

....

Synthèse

Henri Poincaré :

« La mathématique est l'art de donner le même nom à des choses différentes. »

Henri Poincaré né en 1854 à Nancy et meurt en 1912 à Paris. Il est mathématicien, physicien, philosophe et ingénieur français. Il est l'un des derniers mathématiciens à comprendre l'ensemble des savoir mathématiques de son époque. Il est aussi philosophe, et est souvent considéré comme un des derniers grands savants universels.

Les deux exercices que nous venons de traiter, modélisation d'une propagation épidémique et propagation d'une rumeur, donnent une illustration de l'une de ses affirmations restée célèbre : « La mathématique est l'art de donner le même nom à des choses différentes ».

En considérant une épidémie puis une rumeur, nous avons imaginé des modèles et des programmes informatiques presque identiques. Nous avons ainsi abstrait l'épidémie et la rumeur de leurs contextes respectifs. Nous avons finalement, en quelques sortes, donné le même nom à des choses différentes.