

1.1 Variables, structure conditionnelle et autres boucles

1.1.1 Structure conditionnelle, boucles et autres réjouissances

- En fonction d'une valeur entière saisie par l'utilisateur (on utilisera à cette fin la fonction `input` de Python et on prendra bien garde à convertir le résultat en entier), afficher l'un des messages suivants :
 - “Ce nombre est pair”
 - “Ce nombre est impair, mais est multiple de 3”
 - “Ce nombre n'est ni pair ni multiple de 3”
- Dénombrer le nombre de voyelles dans une phrase que l'utilisateur saisira.
- Projet Euler n°1^a: Calculer la somme des entiers divisibles par 3 ou par 5 dans l'intervalle $[0, N[$ où N sera fourni par l'utilisateur.
- Projet Euler n°16: Donner la somme des chiffres de la représentation décimale de 2^N où N est un entier naturel fourni par l'utilisateur. Faire de même avec $N!$ (Projet Euler n°20).
- La suite de Fibonacci est définie de la façon suivante

$$\begin{aligned}\mathcal{F}_1 &= 1 \\ \mathcal{F}_2 &= 1 \\ &\dots \\ \mathcal{F}_n &= \mathcal{F}_{n-2} + \mathcal{F}_{n-1}\end{aligned}$$

- Calculer le résultat de cette suite pour un ordre n donné par l'utilisateur. On essaiera en particulier de tirer profit de l'affectation parallèle des variables dans Python.
 - Projet Euler n°2: Calculer la somme des termes de Fibonacci pairs jusqu'à une valeur N donnée par l'utilisateur.
 - Projet Euler n°25: Déterminer l'indice n du premier terme \mathcal{F}_n dont la valeur contient plus de 1000 chiffres.
- La formule de Wallis permet de calculer la valeur de π via l'expression

$$\pi = 2 \prod_{n=1}^{\infty} \frac{4n^2}{4n^2 - 1}$$

Déterminer le nombre d'itérations n nécessaires pour que la valeur de π soit correcte jusqu'à la 7^{ème} décimale.

^aLe *Project Euler* est situé à l'adresse suivante: <http://projecteuler.net>. Ce site propose un grand nombre de problèmes (587 en date du 22/01/2017) aux apprentis programmeurs comme aux programmeurs chevronnés.

1.

```
N = int(input("Saisissez un entier ? "))
if N % 2 == 0:
    print("Ce nombre est pair")
elif N % 3 == 0:
    print("Ce nombre est impair, mais est multiple de 3")
else:
    print("Ce nombre n'est ni pair ni multiple de 3")
```
2.

```
voyelles = "aeiouy"
phrase = input("Quelle est la phrase ? ")
count = 0
for c in phrase:
    if c in voyelles:
        count += 1
print("Nombre de voyelles =", count)
```
3.

```
N = int(input("Quelle est la valeur de N ? "))
somme = 0
for i in range(0, N):
    if i % 3 == 0 or i % 5 == 0:
        somme += i
print("Somme =", somme)
```
4.

```
N = int(input("Quelle est la valeur de N ? "))
somme = 0
for i in str(2**N):
    somme += int(i)
print("Somme =", somme)
```
5. (a)

```
N = int(input("Quelle est la valeur de N ? "))
f = g = 1
for i in range(0, N-1):
    f, g = g, f+g
print("F({}) = {}".format(N, f))
```

 (b)

```
N = int(input("Quelle est la valeur de N ? "))
f = g = 1
somme = 0
while f < N-1:
    if f % 2 == 0:
        somme += f
    f, g = g, f+g
print("Somme =", somme)
```

 (c)

```
f = g = 1
n = 0
while f < 10**1000:
    f, g, n = g, f+g, n+1
print("n = ", n)
```
6.

```
i = 1; p = 1
while abs(2*p - math.pi) > 1e-7:
    p = 4*i**2/(4*i**2-1)
    i += 1
print("pi = {} pour {} itérations".format(2*p, i))
```

1.2 Containers et fonctions

1.2.1 Utilisation des *containers* de Python

1. Reprendre le Projet Euler n°1 dont l'objet est le calcul de la somme des entiers divisibles par 3 ou par 5 dans l'intervalle $[0, N[$ **en utilisant une liste établie "en compréhension"**. On s'aidera de la fonction intégrée `sum` pour calculer la somme.

2. Projet Euler n°6 : Calculer le résultat de la formule suivante

$$\left(\sum_{k=1}^n k\right)^2 - \sum_{k=1}^n k^2$$

où n sera saisi par l'utilisateur.

3. On s'aidera des méthodes `join` et `split` de la classe `str`. Demander à l'utilisateur de saisir une chaîne de caractères puis
- (a) renvoyer une chaîne telle qu'un lecteur l'aurait lue de droite à gauche
 - (b) renvoyer une chaîne où le premier mot devient le dernier, le second mot devient l'avant dernier, ...
 - (c) renvoyer la chaîne de caractères saisie sans les voyelles
4. Demander à l'utilisateur de saisir son nom, prénom et âge sur une seule ligne et en utilisant le caractère `,` comme séparateur de champ.
- (a) Présenter le résultat sous la forme "Votre nom et prénom sont XXX XXX et vous êtes agé(e) de XXX" en n'oubliant pas les majuscules aux noms propres.
 - (b) Créer ensuite un dictionnaire indexé sur le nom de la personne et conservant son âge.
5. Demander à plusieurs utilisateurs de saisir nom, prénom et âge en utilisant toujours le caractère `,` comme séparateur de champ et stocker chacune des chaînes dans une liste. On terminera la saisie en entrant le caractère `$`. Construire un dictionnaire toujours indexé sur le nom mais qui conservera le prénom et l'âge de la personne. Présenter finalement chacune des entrées du dictionnaire sous une forme intelligible.

```
1. In [1]: sum([x for x in range(N) if x % 3 == 0 or x % 5 == 0])

2. In [1]: N = int(input("Quelle est la valeur de N ? "))

In [2]: r = range(1, N+1)

In [3]: sum(r)**2 - sum([x**2 for x in r])

3. In [1]: s = input("Saisissez votre phrase\n")
Qu'est qu'un chat qui voit dans le futur ?

In [2]: s[::-1]
Out[2]: "? rutuf e! snad tiouv iuq tahc nu'uq tse'uQ"

In [3]: " ".join(s.split()[::-1])
Out[3]: "? futur le dans voit qui chat qu'un Qu'est"

In [4]: "".join(c for c in s if c.lower() not in "aeiouy")
Out[4]: "Q'st q'n cht q vt dns l ftr ?"

4. (a) In [1]: s = input("Veuillez saisir votre nom, prénom et âge en séparant chaque champ par une virgule ")
Simpson, Homer, 55

In [2]: l = s.split(",")

In [3]: print("Votre nom et prénom sont {}{} et vous êtes agé(e) de{}".format(l[0], l[1], l[2]))

(b) In [4]: d = {l[0]:l[2]}
```

correction

```

5. In [1]: infos={}
...: while True:
...:     print("Tapez '$' pour quitter...")
...:     chaine = input()
...:     if chaine == "$":
...:         break
...:     l = chaine.split(",")
...:     infos[l[0]] = (l[1],l[2])
...:
...: for key,values in infos.items():
...:     print("Nom {}, prénom {}, âge {}".format(key, values[0], values[1]))

```

correction

1.2.2 Modularité

- Dans un fichier intitulé `euler_project.py`, reprenez les différentes solutions aux projets Euler résolus jusqu'à présent et isoler ces solutions dans des fonctions dont le nom correspondra au numéro du projet préfixé du terme `euler`. Documenter chacune de ces fonctions puis résoudre les 587 problèmes proposés^b
- Exécuter le fichier `euler_project.py` dans la console ipython à l'aide de la commande `%run euler_project.py`. Tester les différentes fonctions puis construire un dictionnaire qui associera le numéro du projet à la fonction idoine. Demander finalement à l'utilisateur quel numéro de projet résoudre, s'assurer que le projet a une solution et proposer cette solution.

Fichier `euler_project.py`

tion

^bje plaisante ! ne le faites pas

```

1  """
2  Module for Euler projects
3
4  This file holds several solution of Euler project
5  """
6
7  def euler001(n=None):
8      """Solution for Euler project n°1
9
10     Find the sum of all the multiples of 3 or 5 below n.
11
12     """
13     if n == None:
14         n = int(input("Give the n value : "))
15     return sum([x for x in range(n) if x % 3 == 0 or x % 5 == 0])
16
17  def euler002(n=None):
18      """Solution for Euler project n°2
19
20     By considering the terms in the Fibonacci sequence whose values do not
21     exceed a given value, find the sum of the even-valued terms.
22
23     """
24     if n == None:
25         n = int(input("Give the n value : "))
26     f, g, somme = 1, 1, 0
27     while f < n:
28         if f % 2 == 0:
29             somme += f
30         f, g = g, f+g
31     return somme
32
33  def euler006(n=None):
34      """Solution of Euler project n°6
35
36     Find the difference between the sum of the squares of the first n natural
37     numbers and the square of the sum.
38
39     """
40     if n == None:
41         n = int(input("Give the n value : "))
42     r = range(1, n+1)
43     return sum(r)**2 - sum([x**2 for x in r])
44
45  def euler016(n=None):
46      """Solution for Euler project n°16
47
48     What is the sum of the digits of the number 2**n?
49
50     """
51     if n == None:
52         n = int(input("Give the n value : "))
53     somme = 0
54     for i in str(2**n):
55         somme += int(i)
56     return somme
57
58  def euler025(n=None):
59      """Solution for Euler project n°25
60
61     What is the index of the first term in the Fibonacci sequence to contain n digits?
62
63     """
64     if n == None:
65         n = int(input("Give the n value : "))
66     f, g, i = 1, 1, 0
67     while f < 10**n:
68         f, g, i = g, f+g, i+1
69     return i

```

```
In [1]: %run euler_project.py

In [2]: projects = {1 : euler001, 2 : euler002, 6 : euler006, 16 : euler016, 25 : euler025}

In [3]: i = 0
...: while True:
...:     i = int(input("Project number ? "))
...:     if i not in projects.keys():
...:         print("Project n°{} not solved yet".format(i))
...:         continue
...:     print(projects[i]())
...:     break
...:
```