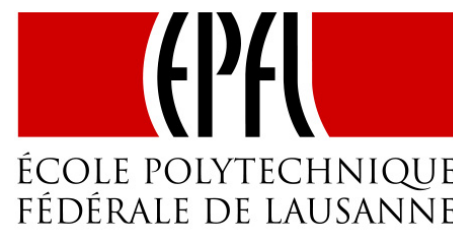


# TO BE IN SEASON OR NOT TO BE IN SEASON

KATE DOPIRO, FABIEN ZELLWEGER & CHRISTOPHER BENZ



## DATA SCRAPING:

### - Introduction:

Humankind has acquired knowledge about food for ages. As a result we know that each food has its season. However with the rise of civilization we are no longer dependent on it as any food can be transported to any point in the world. The question is then: how much food do we consume out of the season? Our project focuses on two aspects: the season the foods are naturally produced, and the area where they are naturally produced.

### - Números:

Python utiliza números inteiros e flutuantes. Pode ser utilizada a função type pra checar o valor de um objeto:

```
type(3)      retorna: <type 'int'>
type(3.14)   retorna: <type 'float'>
.....
```

### - Entrada de Dados:

```
A = input()  Aguarda a entrada de caracteres
              armazenados em A
B = int(input()) Aguarda a entrada de inteiros
              armazenados em B
A,B = map(int,input().split()) Aguarda a entrada de inteiros
                              separados por
                              espaço, armaze-
                              nados em A e B
                              respectivamente
input("Pressione ENTER") Aguarda pressionar ENTER para
                          prosseguir - como
                          não declarou nenhuma
                          variável, não irá gravar
                          nada.
```

## DATA PROCESSING

### - if

- if teste:  
.....# faça algo se teste der verdadeiro  
elif teste2  
.....# faça algo se teste2 der verdadeiro  
else:  
.....# faça algo se ambos derem falso

### - while:

- while teste:  
.....# enquanto verdadeiro continue fazendo algo

### - for:

- for x in sequência  
.....# enquanto o x estiver na sequência informada  
.....# faça algo para cada item na sequência  
.....# a sequência pode ser uma lista,  
.....# elementos de uma string, etc.
- for x in range(10)  
.....# repita algo 10 vezes (de 0 a 9)
- for x in range(5,10)  
.....# repita algo 5 vezes (de 5 a 9)

### - Testes Lógicos

```
10 == 10      retorna: True
10 == 11      retorna: False
10 != 11      retorna: True
"jack" == "jack" retorna: True
"jack" == "jake" retorna: False
10 > 10       retorna: False
10 >= 10      retorna: True
"abc" >= "abc" retorna: True
"abc" < "abc" retorna: False
```

## RESULTS

### - Listas no Python

Listas são compostas por elementos de qualquer tipo (podem ser alteradas)

### Manipulação de Listas no Python

```
Criação
uma_lista = [5,3,'p',9,'e']      cria: [5,3,'p',9,'e']

Acessando
uma_lista[0]                      retorna: 5

Fatiando
uma_lista[1:3]                    retorna: [3,'p']

Comprimento
len(uma_lista)                    retorna: 5

count( item)
Retorna quantas vezes o item foi encontrado na lista.
cont(uma_lista('p'))              retorna: 1
Pode ser usado juntamente com a função while para 'andar' pelo comprimento da lista:
while x < len(uma_lista):          retorna: [3,'p']

Ordenar - sort()
uma_lista.sort()                  retorna: [3,5,9,'e','p']
Ordenar sem alterar a lista
print(sorted(uma_lista))           retorna: [3,5,9,'e','p']

Adicionar - append(item)
uma_lista.append(37)               retorna: [5,3,'p',9,'e',37]

Inserir - insert(position,item)
insert(uma_lista.append(3),200)    retorna: [5,3,200,'p',9,'e']

Retornar e remover - pop(position)
uma_lista.pop()                   retorna: 'e' e a lista fica [5,3,'p',9] - remove o último elemento
uma_lista.pop(1)                  retorna: 3 e a lista fica [5,'p',9,'e'] - remove o elemento 1

Remover - remove(item)
uma_lista.remove('p')              retorna: [5,3,9,'e']

Inserir
uma_lista.insert(2,'z')            retorna: [5,'z',3,'p',9,'e'] - insere na posição numerada

Inverter - reverse()
reverse(uma_lista)                 retorna: ['e',9,'p',3,5]

Concatenar
uma_lista+[0]                     retorna: [5,3,'p',9,'e',0]
uma_lista+uma_lista                retorna: [5,3,'p',9,'e',5,3,'p',9,'e']

Encontrar
9 in uma_lista                     retorna: True
for x in uma_lista                 retorna toda a lista, um elemento por linha
.....print(x)
```