Lesson 3

While + List

Con hãy nhặt cho xong chỗ gạo này rồi có đi đâu hãy đi

???

```
n = int(input('Nhập số:'))
if n > 0:
    print('Giá trị tuyệt đối là:', n)
else:
    print('Giá trị tuyệt đối là:', -n)
```

Nhập 'abc' thì sao?

???

```
Nhập số:abc
Traceback (most recent call last):
  File "input_validation.py", line 1, in <module>
    n = int(input('Nhập số:'))
ValueError: invalid literal for int() with base 10: 'abc'
```

- Lỗi ở int() vì không thể chuyển 'abc' ra số nguyên

=> Cần sửa lại để thông báo và cho phép người dùng nhập lại nếu nhập sai

=> Đoạn code nhập cần phải được lặp lại đến khi người dùng nhập đúng

Solution

```
4   n_str = input('Nhập số:')
5   while not n_str.isdigit():
6       n_str = input('Số bạn nhập không phải là số, nhập lại:')
7
8   n = int(n_str)
9
10  if n >= 0:
11      print('Giá trị tuyệt đối là:', n)
12  else:
13      print('Giá trị tuyệt đối là:', -n)
14
```

While

- While là gì?
- While dùng làm gì?
- While dùng thế nào?

While

```
while True:
    print('hi')
```

While

Lệnh print sẽ lặp mãi cho đến khi ctrl+c, đó là vòng lặp vô hạn

```
while True:
    print('hi')
```

Khác gì so với for

```
for v in range(5):
    print('Hi')
```

While

- Không giống như for, thông thường nhìn while ta ko biết ngay được số lần lặp

- Vì không có số lần lặp nên while sẽ còn lặp cho đến khi biểu thức điều kiện còn đúng

While

- Ăn xong thì mới đi ngủ!

- Dạy xong thì mới được nghỉ!

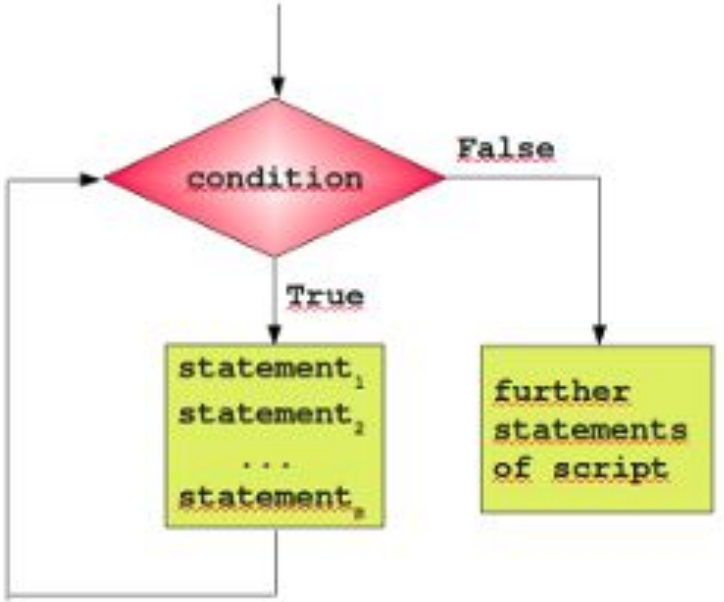
- Học xong thì mới được về!

While

- While là một cấu trúc lặp (giống for)

- Dùng để lặp đi lặp lại một công việc

  theo một điều kiện nào đó còn đúng



While

Đoạn code này sẽ chạy thế nào?

```
while False:
    print('hi')
```

"For" by "While"

Có cách nào dùng while để thay vòng for sau:

```
for v in range(10):
    print('Hi ', v)
```

Break

Sử dụng lệnh break để thoát while

```
dem = 0
while True:
    print('Hi', dem)

    dem += 1
    if dem >= 3:
        break
```

Break

Sử dụng biểu thức điều kiện để thoát while

```
dem = 0
while dem<3:
    print('Hi', dem)

    dem += 1
```

Break

Sử dụng 1 biến trạng thái để thoát while

```
loop = True
dem = 0
while loop:
    print('Hi', dem)
    dem += 1
    if dem >= 3:
        loop = False
```

Break

Lệnh break có thể dùng với for

```
for i in range(1000):
    print('Hi', i)
    if i >= 2:
        break
```

Ex 1

Viết chương trình yêu cầu người dùng nhập password, độ dài pass lớn hơn 8 ký tự. Nếu không cho phép nhập lại.

Ex 2

Lãi suất ngân hàng acb đang là 6.5% một năm. Hnay anh sẽ đi gửi 21 triệu. Viết chương trình để tính:

- Sau 9 năm thì a có bao nhiêu tiền trong tài khoản?

- Muốn mua một căn nhà giá 1.2 tỷ thì a cần gửi ngân hàng ít nhất là bao nhiêu năm?

List

- List là gì?
- List dùng làm gì?
- List dùng thế nào?

Why

? Cần lưu trữ danh sách các mặt hàng đang có trong cửa hàng.

- Bad: dùng tập các biến để lưu. Mat_hang_1, mat_hang_2..., mat_hang_n

```
mat_hang_1="Quần jean"
mat_hang_2="Áo sơ mi"
mat_hang_3="Áo phông"
```

=> Dài dòng, Khi cần tổng hợp, in ra các mặt hàng thì sao ???

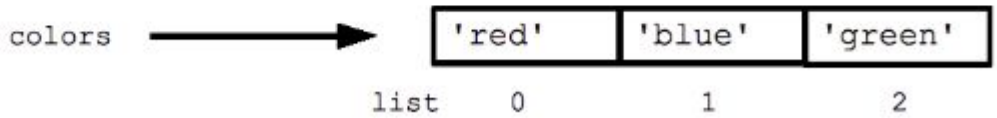
=> Thêm hàng mới, bỏ hàng cũ thì sao ???

=> Cần có một cấu trúc dữ liệu lưu trữ linh động, ngắn gọn, tương thích tốt với các vòng lặp.

```
mat_hang = ["Quần jean", "Áo sơ mi", "Áo phông"]
```

List in python

```
colors = ['red', 'blue', 'green']
print(colors[0])  # red
print(colors[1])  # blue
print(colors[2])  # green
```



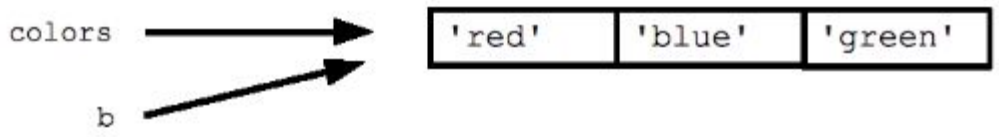
- Built-in type

- within square brackets []

- Similar to strings: len and indexer, zero base index

List in python

```
b = colors   ## Does not copy the list
```



when b[0]=1

colors[0] = ?

CRUD - Create, read, update, Delete

CRUD - Create, read, update, Delete

Init

- Empty list: a=[]
- a=[1,2,3]
- a=list(range(3))

CRUD - Create

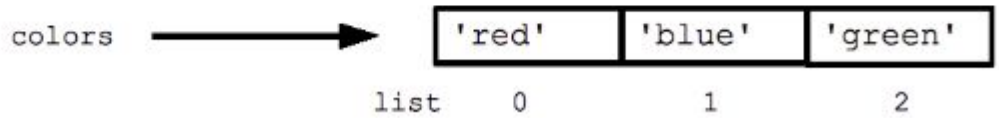
- Append: list.append(elem)

- Insert: list.insert(index, elem)

- Extend: list.extend(list2)

CRUD - Read

```
colors = ['red', 'blue', 'green']
print(colors[0])  # red
print(colors[1])  # blue
print(colors[2])  # green
```



<list variable>[index]

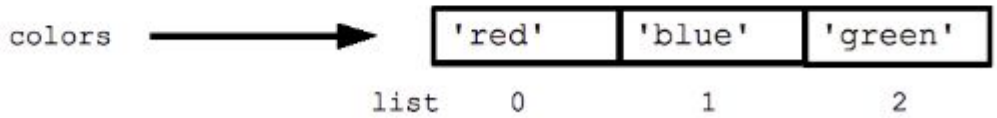
- Base zero index

- Lưu ý về index

- ex: e=colors[0]

CRUD - Update

```
colors = ['red', 'blue', 'green']
print(colors[0])  # red
print(colors[1])  # blue
print(colors[2])  # green
```

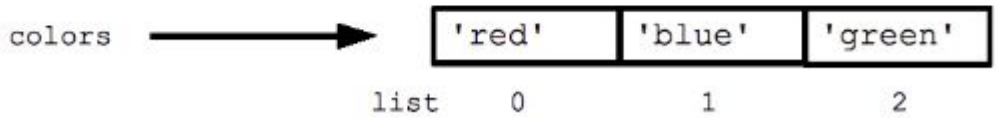


<list variable>[index] = <new value>

- ex: colors[1]='yellow'

CRUD - Delete

```
colors = ['red', 'blue', 'green']
print(colors[0])  # red
print(colors[1])  # blue
print(colors[2])  # green
```



- Delete by index: del colors[index]

- Delete by index: colors.pop(index)

- Delete FIRST item: colors.remove("red")

CRUD - Slices

```
list = ['a', 'b', 'c', 'd']
print(list[1:-1])  # ['b','c']
list[0:2] = 'z'  # replace ['a','b'] with z
print(list)  # ['z', 'c', 'd']
```

Functions

- list.reverse()
- list.clear()
- list.extend(otherList)
- list.index(value)
- list.sort()

.....

?

```
arr = [x * x for x in range(100)]

print(arr)
```

Ex

Nhập n số nguyên từ bàn phím.

- In ra dãy vừa nhập

- In ra trung bình cộng các số chẵn

Ex

```
1
2   n = int(input('nhap so phan tu:'))
3   ds = []
4
5   for v in range(n):
6       so = input("Nhap phan tu " + str(v) + " :")
7       so = int(so)
8       ds.append(so)
9
10  print('day so vua nhap la:', ds)
11
12  tong_so_chan = 0
13  so_so_chan = 0
14  for v in ds:
15      if v % 2 == 0:
16          tong_so_chan += v
17          so_so_chan += 1
18  print('Trung binh cong cac so chan la: ', tong_so_chan / so_so_chan)
19
```

Ex

Nhập số nguyên n 0<=n<=1000000, phân tích n ra thừa số nguyên tố

```
➜  list git:(master) ✗ python3 ex1.py
Nhập số cần phân tích:121212
Số bạn vừa nhập là: 121212
là tích các số nguyên tố sau:
[2, 2, 3, 3, 7, 13, 37]
```

Ứng dụng