Joao Quinta

TP 1

In [44]:
```
import numpy as np
```

EX 1 code

In [45]:
```
#define table
p_t = np.array([100,101,102,103,104,105,106,107,108,109,110,111,112])
print("p_t : ",p_t)

print()

# 1.1
m_r = [p_t[i+1]/p_t[i] - 1 for i in range(12)]
print("Monthly return : ", np.round(m_r, 5))

print()

# 1.2
a_r = np.prod(np.array(m_r)+1) - 1
print("Annual return :                          ", a_r,   " --> TO avoid compounding errors, I'
print("Sum :                                    ", np.sum(m_r))

print()

# 1.3
avg_m_r = (1 + 0.12)**(1/12) - 1
mean_m_r = np.mean(m_r)
print("Average monthly return :                 ", avg_m_r)
print("Average of all monthly return :          ", mean_m_r)
print("Difference between them :               ", avg_m_r - mean_m_r)
```

```
p_t :  [100 101 102 103 104 105 106 107 108 109 110 111 112]

Monthly return :  [0.01    0.0099  0.0098  0.00971 0.00962 0.00952 0.00943 0.00935 0.009
26
 0.00917 0.00909 0.00901]

Annual return :                    0.11999999999999966  --> TO avoid compounding errors,
I'll round this value to 0,12
Sum :                              0.11386608961336608

Average monthly return :           0.009488792934583046
Average of all monthly return :    0.009488840801113 84
Difference between them :          -4.786653079458247 4e-08
```

EX 1 comment

1.1 - we simply use the formula: $R_m(t) = \frac{P(t)-P(t-1)}{P(t-1)}$

1.2 - we use the formula: $R_a = \prod_{t=1}^{12}(1+R_m(t)) - 1$ ===> $R_a = 0.12$

When we sum the 12 monthly returns we get 11.38%, which is wrong

1.3 - we compute the monthly avg return with the following formula: $R_m = (1+R_a)^{\frac{1}{12}} - 1 = 0.009489$

The mean of monthly returns is: $\frac{\sum_{t=0}^{12} R_m(t)}{12} = 0.009489$

As you can see the difference between both resuts is: -4.7866530794582474e-08

EX 2 code

In [46]:
```
nb_shares = [10,10]
cost = [85,30]

# 2.1
initial_value = np.sum(np.dot(nb_shares, cost))
print("Initial portofolio value :   ",  initial_value)

print()

# 2.2
alphas = [nb_shares[i]*cost[i]/initial_value for i in range(2)]
print("Value of alphas :              ", alphas)

print()

# 2.3
new_cost = [90, 28]
returns = [new_cost[i]/cost[i] - 1 for i in range(2)]
print("Returns :                      ", returns)

print()

# 2.4
p_return = np.sum([nb_shares[i]*new_cost[i] for i in range(2)]) / initial_value - 1
print("Portofolio return :            ", p_return)
end_value = np.sum([nb_shares[i]*new_cost[i] for i in range(2)])
print("Final portfolio value V(t) : ", end_value)
```

```
Initial portofolio value :    1150

Value of alphas :             [0.7391304347826086, 0.2608695652173913]

Returns :                     [0.05882352941176472, -0.06666666666666665]

Portofolio return :           0.026086956521739202
Final portfolio value V(t) :  1180
```

EX 2 comment

2.1 - we compute the initial value by computing the total value of our shares:
$V(t-1) = 10 \cdot 85 + 10 \cdot 30 = 1150$

2.2 - to compute the α we must compute the proportion of each asset to the initial portfolio value, in other words, how much of the inital portfolio value is due to a given stock, we use the following formula:

$$\alpha_{asset} = \frac{\#shares_{asset} \cdot cost_{asset}}{V(t-1)}$$

2.3 - the return of an asset is computed with :

$$R_{asset} = \frac{P_{asset}(t) - P_{asset}(t-1)}{P_{asset}(t-1)}$$

2.4 - the return of the portfolio is computed with :

$$R_p(t) = \frac{\sum_{a \in assets} \#shares_a \cdot P_a(t)}{V(t-1)}$$

EX 3 code

In [47]:
```
# 3.1
C = 10000
R = 4.5 / 100
m = 2

final_val = [C * (1 + R/m)**(i*m) for i in [1,5,10]]

for i, year in enumerate([1,5,10]):
    print("The final value in", year,"year(s) will be : ", final_val[i])

print()

# 3.2
final_goal = 10000
initial_val = [final_goal / (1+R/m)**(i*m) for i in [1,3,10]]

for i, year in enumerate([1,3,10]):
    print("The initial value in order to have",final_goal,"in",year,"year(s) has to be :

print()

# 3.3
C = 10000

cont_comp = C * np.exp(R*1)
print("Value in one year if it was continously compounded : ", cont_comp)
```

```
The final value in 1 year(s) will be :  10455.062499999998
The final value in 5 year(s) will be :  12492.034264621258
The final value in 10 year(s) will be :  15605.092006847151

The initial value in order to have 10000 in 1 year(s) has to be :  9564.744352317359
The initial value in order to have 10000 in 3 year(s) has to be :  8750.242719752925
The initial value in order to have 10000 in 10 year(s) has to be :  6408.164716755423

Value in one year if it was continously compounded :  10460.278599087169
```

EX 3 comment

3.1 - to compute the value of a porfolio in the future we use:

$$FV_n^m = V \cdot (1 + \frac{R}{m})^{m \cdot n}$$

where V is the initial value invested, R = 4.5% the return, m = 2 the frequency of payment, and n the number of years.

3.2 - if order to know the initial value we need to invest to obtain 10000 after x years we use:

$$V = \frac{FV_n^m}{(1 + \frac{R}{m})^{m \cdot n}}$$

3.3 - if the return is continuously compounded we compute it with:

$$FV_n^{\infty} = V \cdot e^{R \cdot n}$$

```
In [ ]:
```