

Repetição

Exemplo motivador (1/3)

A sensação térmica é influenciada por vários fatores climáticos, como a umidade e densidade do ar, mas principalmente pela velocidade do vento. A seguinte fórmula empírica calcula seu valor:

$$ST = 33 + \left(10\sqrt{v} + 10,45 - v\right) \cdot \frac{T - 33}{22}$$

onde T é a temperatura em graus Celsius e v é a velocidade do vento em Km/h

Por exemplo, para temperatura de 5°C e ventos de 40km/h, a sensação térmica é em torno de -9.8°C.

Faça um programa que pergunte a temperatura e a velocidade do vento ocorridas no início da manhã de um dia e mostre a sensação térmica equivalente.

Exemplo motivador (2/3)

A sensação térmica é influenciada por vários fatores climáticos, como a umidade e densidade do ar, mas principalmente pela velocidade do vento. A seguinte fórmula empírica calcula seu valor:

$$ST = 33 + \left(10\sqrt{v} + 10,45 - v\right) \cdot \frac{T - 33}{22}$$

onde T é a temperatura em graus Celsius e v é a velocidade do vento em Km/h

Por exemplo, para temperatura de 5°C e ventos de 40km/h, a sensação térmica é em torno de -9.8°C. Faça um programa que pergunte a temperatura e a velocidade do vento ocorridas no início da manhã de um dia e mostre a sensação térmica equivalente.

Dados necessários ?

Respostas Exibidas ?

Como Dados → Respostas ?

Exemplo motivador (3/3)

A sensação térmica é influenciada por vários fatores climáticos, como a umidade e densidade do ar, mas principalmente pela velocidade do vento. A seguinte fórmula empírica calcula seu valor:

$$ST = 33 + \left(10\sqrt{v} + 10,45 - v\right) \cdot \frac{T - 33}{22}$$

onde T é a temperatura em graus Celsius e v é a velocidade do vento em Km/h

Por exemplo, para temperatura de 5°C e ventos de 40km/h, a sensação térmica é em torno de -9.8°C. Faça um programa que pergunte a temperatura e a velocidade do vento ocorridas no início da manhã de um dia e mostre a sensação térmica equivalente.

Dados necessários? Temperatura e velocidade do vento

Respostas Exibidas? Sensação térmica

Como Dados → Respostas? Fórmula no enunciado

Uma Solução para 1 horário

```
def sensacaoTermica(temp,vel):  
    st = 33+(10*vel**0.5+10.45-vel)* ((temp-33)/22)  
    return st  
  
temp = float(input('Temperatura? '))  
vel = float (input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente:',st)
```

Problema

```
def sensacaoTermica(temp,vel):  
    st = 33+(10*vel**0.5+10.45-vel)* ((temp-33)/22)  
    return st  
  
temp = float(input('Temperatura? '))  
vel = float (input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente:',st)
```

O que deve ser
feito para
mostrar a
sensação térmica
também no início
da tarde?

Uma Solução (?) para 2 horários

```
def sensacaoTermica(temp,vel):  
    st = 33+(10*vel**0.5+10.45-vel)* ((temp-33)/22)  
    return st
```

```
temp = float(input('Temperatura? '))  
vel = float (input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente:',st)
```

Copy/Paste

Uma Solução (?) para 2 horários

```
def sensacaoTermica(temp,vel):  
    st = 33+(10*vel**0.5+10.45-vel)* ((temp-33)/22)  
    return st
```

```
temp = float(input('Temperatura? '))  
vel = float (input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```

```
temp = float(input('Temperatura? '))  
vel = float (input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```



Uma Solução (?) para 2 horários

```
def sensacaoTermica(temp,vel):  
    st = 33+(10*vel**0.5+10.45-vel)* ((temp-33)/22)  
    return st
```

```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```

```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```



E para mostrar a sensação
térmica de vários horários do
dia?

Uma Solução para vários horários

```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```

NECESSIDADE DE **REPETIR** UMA SEQUÊNCIA DE INSTRUÇÕES!

Repetição para vários horários

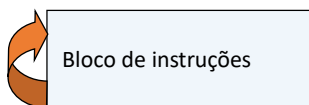
```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp, vel)  
print('Sensação térmica equivalente: ', st)
```



NECESSIDADE DE **REPETIR** UMA SEQUÊNCIA DE INSTRUÇÕES!

Estruturas de Repetição

Permitem que um bloco de instruções seja executado *mais de uma vez* (também chamada de *loop* ou *laço*).



Problema: Como/quando parar o ciclo formado?

Controlando os tipos de repetição

Quanto a quantidade de ciclos, a repetição pode ser:

Determinada:

A quantidade de execuções é conhecida a priori.

- ✓ Nadar 50 piscinas
- ✓ Lotação de um elevador por número de pessoas

Quando parar o ciclo formado?

Ao atingir a meta determinada

Como controlar?

Contando número de execuções,

X

Indeterminada:

A quantidade de execuções é desconhecida

- ✓ Nadar até cansar
- ✓ Lotação de um elevador por peso máximo

Quando parar o ciclo formado?

- a) Por determinação do usuário
- b) Por determinado contexto

Como controlar?

- a) Entrada de um valor finalizador
- b) Avaliação de valores das variáveis

Estrutura de repetição: *while*

Sintaxe:

```
while condição:
```

```
    i1  
    i2  
    . . .  
    in
```

Executa a sequência de instruções **enquanto** certa condição for verdadeira.

Quando a condição for falsa, "pula" o bloco de instruções interno e executa o resto do programa

Estrutura de repetição: *while*

Sintaxe:

```
while condição:
```

```
    i1
```

```
    i2
```

```
    ...
```

```
    in
```

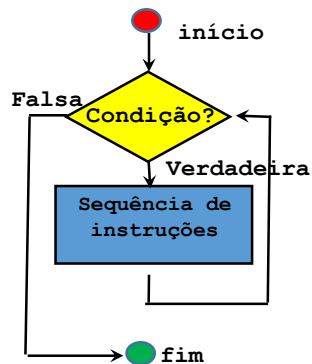
Instruções internas (que serão repetidas)
devem estar identadas

Executa a sequência de instruções **enquanto** certa condição for verdadeira.

Quando a condição for falsa, "pula" o bloco de instruções interno e executa o resto do programa

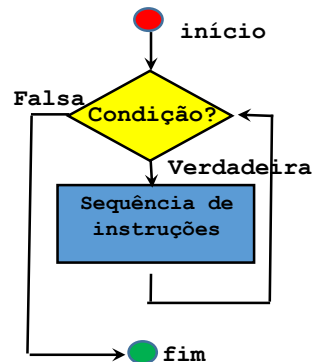
While em Funcionamento (1/4)

1. Testa a condição.



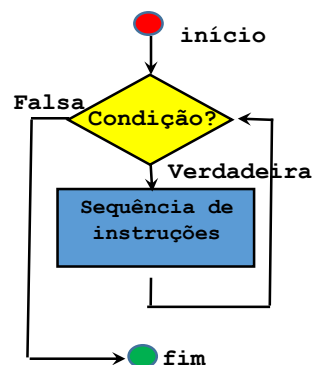
While em Funcionamento (2/4)

1. Testa a condição.
2. Caso a condição seja **falsa**, sai do ciclo.



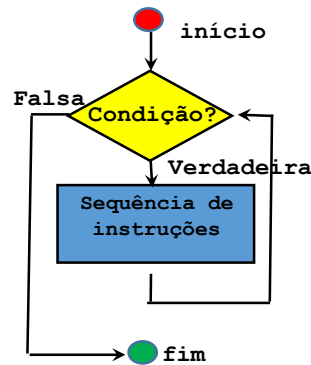
While em Funcionamento (3/4)

1. Testa a condição.
2. Caso a condição seja **falsa**, sai do ciclo.
3. Caso a condição seja **verdadeira**, executa todas as instruções da sequência interna.



While em Funcionamento (4/4)

1. Testa a condição.
2. Caso a condição seja **falsa**, sai do ciclo.
3. Caso a condição seja **verdadeira**, executa todas as instruções da sequência interna.
4. Depois de executar a última instrução da sequência interna ao **while**, retorna para o passo 1.



Usando *while* para vários horários

while **condição:**

```
temp = float(input('Temperatura? '))
vel = float(input('Velocidade do Vento? '))
st = sensacaoTermica(temp,vel)
print('Sensação térmica equivalente: ',st)
```

While infinito?

p/ interromper o ciclo,
a condição deve se
tornar falsa

```
while ??????:
```

```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```

While e condição de parada

p/ interromper o ciclo,
a condição deve se
tornar falsa

Depende do tipo de
repetição

```
while ??????:
```

```
temp = float(input('Temperatura? '))  
vel = float(input('Velocidade do Vento? '))  
st = sensacaoTermica(temp,vel)  
print('Sensação térmica equivalente: ',st)
```

Repetição Determinada

Para repetir um número pré-determinado de vezes precisa-se:

CONTAR quantas vezes o bloco já foi executado



variável para armazenar a contagem de execuções realizadas
(contador de execuções)

- Qual o valor inicial?
- Qual o valor final?
- Qual o incremento cada vez que a repetição é executada?

Repetição Determinada: contador de repetições

1. Valor inicial:

✓ Atribuído ANTES do laço

2. Valor final (meta):

✓ Utilizado na condição do laço. Uma condição mal construída leva a um loop infinito

3. Incremento:

✓ Determina a variação do contador, do valor inicial até o valor final.

Uma Solução: 5 horários diários

Contador de repetições: **jaFiz**

- Qual o valor inicial? **0**
- Qual o valor final (meta)? **5**
- Qual o incremento cada vez que a repetição é executada? **+1**

Código Python: 5 horários diários

```
jaFiz = 0
```

```
while jaFiz < 5:
```

```
    temp = float(input('Temperatura? '))
```

```
    vel = float(input('Velocidade do Vento? '))
```

```
    st = sensacaoTermica(temp,vel)
```

```
    print('Sensação térmica equivalente: ',st)
```

```
    jaFiz = jaFiz +1
```

“Chinês”

É uma **simulação** da execução de código, linha a linha, acompanhando os valores das variáveis. Útil para depurar e compreender o quê o código faz.

Para rastrear um programa deve-se:

- ✓ **escrever** os nomes das variáveis
- ✓ **seguir** uma a uma as instruções
- ✓ **atualizar e observar** as variáveis



Chinês: 3 horários diários (1/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz + 1
```

[illegible]

Chinês: 3 horários diários (2/22)

[illegible]

jaFiz = 0

```
while jaFiz < 3:
```

```
temp=float(input('Temp? '))
```

```
vel=float(input('Vento?'))
```

```
st=sensacaoTermica(temp,vel)
```

```
print('Sensação térmica:',st)
```

jaFiz = jaFiz +1

Chinês: 3 horários diários (3/22)

[illegible]

jaFiz = 0

```
while jaFiz < 3:
```

```
temp=float(input('Temp? '))
```

```
vel=float(input('Vento?'))
```

```
st=sensacaoTermica(temp,vel)
```

```
print('Sensação térmica:',st)
```

jaFiz = jaFiz +1

Chinês: 3 horários diários (4/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz + 1
```

[illegible]

Chinês: 3 horários diários (5/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz + 1
```

[illegible]

Chinês: 3 horários diários (6/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8

Chinês: 3 horários diários (7/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8

Chinês: 3 horários diários (8/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8

Chinês: 3 horários diários (9/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8

Chinês: 3 horários diários (10/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8

Chinês: 3 horários diários (11/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	-11.8

Chinês: 3 horários diários (12/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5

Chinês: 3 horários diários (13/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5

Chinês: 3 horários diários (14/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5

Chinês: 3 horários diários (15/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5

Chinês: 3 horários diários (16/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5

Chinês: 3 horários diários (17/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5

Chinês: 3 horários diários (18/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5
2	40	20	44.19

Chinês: 3 horários diários (19/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5
2	40	20	44.19

Chinês: 3 horários diários (20/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5
2	40	20	44.19
3	40	20	44.19

Chinês: 3 horários diários (21/22)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5
2	40	20	44.19
3	40	20	44.19


```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz +1
```

jaFiz	temp	vel	st
0			
0	5		
0	5	30	
0	5	30	-11.8
1	5	30	-11.8
1	3	30	-11.8
1	3	1	6.5
2	3	1	6.5
2	40	1	6.5
2	40	20	6.5
2	40	20	44.19
3	40	20	44.19

E agora, o que acontece?

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

Chinês solução #2 (1/10)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

[illegible]

Chinês solução #2 (2/10)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

[illegible]

Chinês solução #2 (3/10)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
...			

Chinês solução #2 (4/10)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
...			

Chinês solução #2 (5/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	1	6.5
...			

Chinês solução #2 (6/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	1	6.5
0	3	1	6.5
...			

Chinês solução #2 (7/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	30	-11.8
0	5	30	-11.8
0	40	20	44.19
...			

Chinês solução #2 (8/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	1	6.5
0	3	1	6.5
0	40	20	44.19
0	40	20	44.19
...			

Chinês solução #2 (9/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	1	6.5
0	3	1	6.5
0	40	20	44.19
0	40	20	44.19
0	25	30	20.19
...			

Chinês solução #2 (10/10)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
0	5	30	-11.8
0	3	1	6.5
0	3	1	6.5
0	40	20	44.19
0	40	20	44.19
0	25	30	20.19
0	25	30	20.19

loop infinito!!!

3 horários diários: solução #3

Tentativa de corrigir a solução #2 para interromper repetição!

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

Chinês solução #3 (1/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
jaFiz = jaFiz - 1
```

[illegible]

Chinês solução #3 (2/14)

jaFiz = 0

while jaFiz < 3:

temp=float(input('Temp? '))

vel=float(input('Vento?'))

st=sensacaoTermica(temp,vel)

print('Sensação térmica:',st)

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0			
...			

Chinês solução #3 (3/14)

jaFiz = 0

while jaFiz < 3:

temp=float(input('Temp? '))

vel=float(input('Vento?'))

st=sensacaoTermica(temp,vel)

print('Sensação térmica:',st)

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0	5	30	-11.8
...			

Chinês solução #3 (4/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
...			

Chinês solução #3 (5/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
...			

Chinês solução #3 (6/14)

jaFiz = 0

while jaFiz < 3:

```
temp=float(input('Temp? '))
vel=float(input('Vento?'))
st=sensacaoTermica(temp,vel)
print('Sensação térmica:',st)
```

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
...			

Chinês solução #3 (7/14)

jaFiz = 0

while jaFiz < 3:

```
temp=float(input('Temp? '))
vel=float(input('Vento?'))
st=sensacaoTermica(temp,vel)
print('Sensação térmica:',st)
```

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
...			

Chinês solução #3 (8/14)

jaFiz = 0

while jaFiz < 3:

temp=float(input('Temp? '))

vel=float(input('Vento?'))

st=sensacaoTermica(temp,vel)

print('Sensação térmica:',st)

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
...			

Chinês solução #3 (9/14)

jaFiz = 0

while jaFiz < 3:

temp=float(input('Temp? '))

vel=float(input('Vento?'))

st=sensacaoTermica(temp,vel)

print('Sensação térmica:',st)

jaFiz = jaFiz - 1

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
...			

Chinês solução #3 (10/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
-3	40	20	44.19
...			

Chinês solução #3 (11/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
-3	40	20	44.19
...			

Chinês solução #3 (12/14)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

```
    jaFiz = jaFiz + 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
-3	40	20	44.19
-3	25	30	20.19
...			

Chinês solução #3 (13/14)

```
jaFiz = 0
while jaFiz < 3:
```

```
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
```

```
    jaFiz = jaFiz + 1
```

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
-3	40	20	44.19
-3	25	30	20.19
-4	25	30	20.19

Chinês solução #3 (14/14)

```
jaFiz = 0
while jaFiz < 3:
    temp=float(input('Temp? '))
    vel=float(input('Vento?'))
    st=sensacaoTermica(temp,vel)
    print('Sensação térmica:',st)
    jaFiz = jaFiz - 1
```

loop infinito de novo!!!

jaFiz	temp	vel	st
0			
0	5	30	-11.8
-1	5	30	-11.8
-1	3	1	6.5
-2	3	1	6.5
-2	40	20	44.19
-3	40	20	44.19
-3	25	30	20.19
-4	25	30	20.19
-4	25	30	20.19
.....			

Problema #1

O que muda se
houvesse 1
medição por hora
diariamente?

```
jaFiz = 0
while jaFiz < 5:

    temp = float(input('Temperatura? '))
    vel = float(input('Velocidade do Vento? '))
    st = sensacaoTermica(temp,vel)
    print('Sensação térmica equivalente: ',st)
    jaFiz = jaFiz +1
```

Repetição Determinada Constante

```
jaFiz = 0
while jaFiz < 24:

    temp = float(input('Temperatura? '))
    vel = float(input('Velocidade do Vento? '))
    st = sensacaoTermica(temp,vel)
    print('Sensação térmica equivalente: ',st)
    jaFiz = jaFiz +1
```

Problema #2

```
jaFiz = 0
while jaFiz < 24:
```

```
    temp = float(input('Temperatura? '))
    vel = float(input('Velocidade do Vento? '))
    st = sensacaoTermica(temp,vel)
    print('Sensação térmica equivalente: ',st)
    jaFiz = jaFiz +1
```

E se a quantidade de
medições dependesse do
dia e o usuário sabe qual
é?

Repetição Determinada *não* constante

Valor variável fornecido
pelo usuário

```
jaFiz = 0
while jaFiz < n:

    temp = float(input('Temperatura? '))
    vel = float(input('Velocidade do Vento? '))
    st = sensacaoTermica(temp,vel)
    print('Sensação térmica equivalente: ',st)
    jaFiz = jaFiz +1
```

Uma Solução para n (informado) medições

```
n = int(input('quantas medições foram realizadas?'))
jaFiz = 0
while jaFiz < n:

    temp = float(input('Temperatura? '))
    vel = float(input('Velocidade do Vento? '))
    st = sensacaoTermica(temp,vel)
    print('Sensação térmica equivalente: ',st)
    jaFiz = jaFiz +1
```


Mãos na massa: doses de água!

Um professor, sabendo que a dose diária de água é individual, resolveu calcular a quantidade mínima de litros de água que deve ser ingerida por cada um dos seus n (lido) alunos de uma turma. Esta medida é calculada por:

$$\text{litros de água/dia} = 35\text{ml de água} * \text{peso corporal}/1000$$

Faça um programa que inicialmente obtenha a quantidade de alunos da turma (n). A seguir, para cada um dos alunos, obtenha o peso e mostre a quantidade mínima de litros que o aluno deve consumir.

Exemplo:

Entrada

Quantos alunos?3

Peso do aluno 1: 75

Peso do aluno 2: 50

Peso do aluno 3: 90

Saída

Aluno 1 - Peso: 75.00 - Litros de água: 2.62

Aluno 2 - Peso: 50.00 - Litros de água: 1.75

Aluno 3 - Peso: 90.00 - Litros de água: 3.15


Desenvolvendo “doses de água”

Um professor, sabendo que a dose diária de água é individual, resolveu calcular a quantidade mínima de litros de água que deve ser ingerida por cada um dos seus n (lido) alunos de uma turma. Esta medida é calculada por:

$$\text{litros de água/dia} = 35\text{ml de água} * \text{peso corporal}/1000$$

Faça um programa que inicialmente obtenha a quantidade de alunos da turma (n). A seguir, para cada um dos alunos, obtenha o peso e mostre a quantidade mínima de litros que o aluno deve consumir.

Dados: n° de alunos (n)

n vezes  **Dados** $_{1aluno}$: peso

Respostas $_{1aluno}$: litros de água

Doses de água: uma Solução

```
def AguaIndiv(peso):  
    return peso*35/1000  
  
n=int(input("Quantos alunos?"))  
jaFiz=0  
while jaFiz < n:  
    peso = float(input("Peso do aluno " + str(jaFiz+1)))  
    lAgua = AguaIndiv(peso)  
    print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'  
          %(jaFiz+1,peso,lAgua))  
    jaFiz=jaFiz+1
```

Questionamentos possíveis

1. *Quantos alunos devem ingerir mais de 2l ou menos de 2l ou exatamente 2l?*
2. *Qual a quantidade total de litros de água será consumida pela turma?*
3. *Qual a média consumida por aluno?*
4. *Qual a maior/menor quantidade consumida individualmente? De qual aluno ou quantos alunos a consomem?*

Questionamento 1

Quantos alunos devem consumir mais que 2l?

Contador de alunos com quantidade superior a 2l: **qtSup2l**

- Qual o valor inicial?
- Qual o valor final?
- Incremento?
- Quando incrementar o contador?
- Quando exibir?

Solução do questionamento 1

Quantos alunos devem consumir mais que 2l?

Contador de alunos com quantidade superior a 2l: **qtSup2l**

- Qual o valor inicial? **0**
- Qual o valor final? **Calculado pelo programa**
- Incremento? **+1**
- Quando incrementar o contador? **Quando litros calculados > 2**
- Quando exibir? **Após processar toda a turma**



Questionamento 1: python

```
def AguaIndiv(peso):  
    return peso*35/1000  
  
n=int(input("Quantos alunos?"))  
qtSup2l=0  
jaFiz=0  
while jaFiz < n:  
    peso = float(input("Peso do aluno " + str(jaFiz+1)))  
    lAgua = AguaIndiv(peso)  
    if lAgua>2:  
        qtSup2l=qtSup2l+1  
    print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'  
          % (jaFiz+1,peso,lAgua))  
    jaFiz=jaFiz+1  
print(qtSup2l,'alunos devem ingerir diariamente mais de 2l')
```



Questionamento 2

Total de litros consumidos pela turma?



Total de litros consumidos pela turma = \sum litros de cada aluno

Variável para armazenar \sum de litros de cada aluno

Acumulador de litros: **totLit**

- Qual o valor inicial?
- Qual o valor final?
- Valor a somar?
- Quando atualizar o acumulador?
- Quando exibir?

Solução do questionamento 2

Total de litros consumidos pela turma?



Total de litros consumidos pela turma = Σ litros de cada aluno

Variável para armazenar Σ de litros de cada aluno

Acumulador de litros: **totLit**

- Qual o valor inicial? **0**
- Qual o valor final? **Calculado pelo programa**
- Valor a somar? **+litros do aluno**
- Quando atualizar o acumulador? **Em cada repetição**
- Quando exibir? **Após processar toda a turma**

Questionamento 2: Python

```
def AguaIndiv(peso):  
    return peso*35/1000  
  
n=int(input("Quantos alunos?"))  
qtSup2l=0  
totLit=0  
jaFiz=0  
while jaFiz < n:  
    peso = float(input("Peso do aluno " + str(jaFiz+1)))  
    lAgua = AguaIndiv(peso)  
    if lAgua>2:  
        qtSup2l=qtSup2l+1  
    totLit+=lAgua  
    print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'  
          %(jaFiz+1,peso,lAgua))  
    jaFiz=jaFiz+1  
print(qtSup2l,'alunos devem ingerir diariamente mais de 2l')  
print('Consumo total da turma: ',totLit)
```

Questionamento 3

Qual o consumo médio por aluno?

$\text{Consumo médio} = \sum \text{litros por aluno} / \text{nº de alunos}$

– Quando calcular e exibir?

Solução do questionamento 3

Qual o consumo médio por aluno?

$\text{Consumo médio} = \sum \text{litros por aluno} / \text{nº de alunos}$

– Quando calcular e exibir? **Após processar toda a turma**

Questionamento 3: Python

```
def AguaIndiv(peso):  
    return peso*35/1000  
  
n=int(input("Quantos alunos?"))  
qtSup2l=0  
totLit=0  
jaFiz=0  
while jaFiz < n:  
    peso = float(input("Peso do aluno " + str(jaFiz+1)))  
    lAgua = AguaIndiv(peso)  
    if lAgua>2:  
        qtSup2l=qtSup2l+1  
    totLit+=lAgua  
    print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'  
          %(jaFiz+1,peso,lAgua))  
    jaFiz=jaFiz+1  
print(qtSup2l,'alunos devem ingerir diariamente mais de 2l')  
print('Consumo total da turma: ',totLit)  
print('Consumo médio por aluno: ',totLit/n)
```

N turmas com M alunos?

```
def AguaIndiv(peso):  
    return peso*35/1000  
  
n=int(input("Quantos alunos?"))  
qtSup2l=0  
totLit=0  
jaFiz=0  
while jaFiz < n:  
    peso = float(input("Peso do aluno " + str(jaFiz+1)))  
    lAgua = AguaIndiv(peso)  
    if lAgua>2:  
        qtSup2l=qtSup2l+1  
    totLit+=lAgua  
    print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'  
          %(jaFiz+1,peso,lAgua))  
    jaFiz=jaFiz+1  
print(qtSup2l,'alunos devem ingerir diariamente mais de 2l')  
print('Consumo total da turma: ',totLit)  
print('Consumo médio por aluno: ',totLit/n)
```

Como processar as 5
turmas do professor com
quantidade diferente de
alunos ?

1 turma com M alunos!

5
X

```
n=int(input("Quantos alunos?"))
qtSup2l=0
totLit=0
jaFiz=0
maiorQt = -1
while jaFiz < n:
    peso = float(input("Peso aluno " + str(jaFiz+1)))
    lAgua = AguaIndiv(peso)
    if lAgua>2:
        qtSup2l=qtSup2l+1
    if lAgua>maiorQt:
        maiorQt=lAgua
    totLit+=lAgua
    print('Aluno %d-Peso:%6.2f-Litros de
          água:%6.2f' %(jaFiz+1,peso,lAgua))
    jaFiz=jaFiz+1
print(qtSup2l,'alunos devem ingerir + de 2l/dia')
print('Consumo total da turma: ',totLit)
print('Consumo médio por aluno: ',totLit/n)
print('Maior consumo individual', maiorQt)
```

1 turma

Repetições Aninhadas

✓Uma estrutura de repetição pode ser utilizada dentro de outra estrutura de repetição

Cada laço **while** têm sua própria **variável de controle**

Ideia: N turmas com M alunos

Processando 5 turmas

Contador de repetições de turmas: $nTurmas$

- Qual o valor inicial?
- Qual o valor final (meta)?
- Qual o incremento cada vez que a repetição é executada?

Resolvendo com 5 turmas

Processando 5 turmas

Contador de repetições de turmas: $nTurmas$

- Qual o valor inicial? **0**
- Qual o valor final (meta)? **5**
- Qual o incremento cada vez que a repetição é executada? **+1**

Uma Solução para 1 turma

```
def procTurma(n):
    qtSup2l=0
    totLit=0
    jaFiz=0
    while jaFiz < n:
        peso = float(input("Peso do aluno " + str(jaFiz+1)))
        lAgua = AguaIndiv(peso)
        if lAgua>2:
            qtSup2l=qtSup2l+1
            totLit+=lAgua
            print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'
                  %(jaFiz+1,peso,lAgua))
            jaFiz=jaFiz+1
    print(qtSup2l, 'alunos devem ingerir + de 2l/dia')
    print('Consumo total da turma: ',totLit)
    print('Consumo médio por aluno: ',totLit/n)
    return
```

Uma Solução para 5 turmas

```
nTurmas = 0
while (nTurmas<5):
    print('\nTurma ',nTurmas+1)
    n=int(input("Quantos alunos?"))
    procTurma(n)
    nTurmas=nTurmas+1
```

DESAFIO: Exibir no final o total de litros consumido pelas 5 turmas

Qual a maior quantidade consumida individualmente?

Há problemas que solicitam como saída o melhor, o maior, o mais, o menor, o pior, o menor, o menos...

A solução deve "separar" o valor calculado mais adequado de acordo com um critério

↓
Variável para armazenar a maior quantidade consumida

- Qual o valor inicial?
- Qual o valor final?
- Quando atualizar o acumulador?
- Quando exibir?

Uma Solução em Python

```
def procTurma(n):
    qtSup2l=0
    totLit=0
    jaFiz=0
    maiorQt = -1
    while jaFiz < n:
        peso = float(input("Peso do aluno " + str(jaFiz+1)))
        lAgua = AguaIndiv(peso)
        if lAgua>2:
            qtSup2l=qtSup2l+1
        if lAgua>maiorQt:
            maiorQt=lAgua
        totLit+=lAgua
        print('Aluno %d-Peso:%6.2f-Litros de água:%6.2f'
              %(jaFiz+1,peso,lAgua))
        jaFiz=jaFiz+1
    print(qtSup2l,'alunos devem ingerir + de 2l/dia')
    print('Consumo total da turma: ',totLit)
    print('Consumo médio por aluno: ',totLit/n)
    print('Maior consumo individual', maiorQt)
```

Exercício da Tabuada

1. Crie a função **Tabuada** que recebe um número e exibe a tabuada deste número de 1 a 10

Exemplo: numero recebido – 5:

```
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
4 x 5 = 20
5 x 5 = 25
6 x 5 = 30
7 x 5 = 35
8 x 5 = 40
9 x 5 = 45
10 x 5 = 50
```

2. Faça um programa que mostra na tela as tabuadas de todos os números entre 1 a 10.

Tabuada: uma Solução

```
def tabuada(n):
    mult=1
    while mult<=10:
        print('\t%2d x %2d = %d'%(mult,n,mult*n))
        mult=mult+1
    return

num=1
While num<=10:
    print('\n Tabuada do ',num)
    tabuada(num)
    num=num+1
```

Tabuada: números do usuário?

```
def tabuada(n):  
    mult=1  
    while mult<=10:  
        print('\t%2d x %2d = %d'%(mult,n,mult*n))  
        mult=mult+1  
    return  
  
num=1  
while num<=10:  
    print('\n Tabuada do ',num)  
    tabuada(num)  
    num=num+1
```

Como modificar
o programa para
exibir a tabuada
de números
escolhidos pelo
usuário?

Repetição indeterminada

Como exibir a tabuada de números escolhidos pelo usuário?

```
while o usuário quiser :  
    num = int(input('Número? '))  
    print('\n Tabuada do ',num)  
    tabuada(num)
```

COMO PARAR?

Repetição indeterminada: parada?

O usuário deve "dizer" ao programa quando parar!!!

Como?

- ✓ introduzindo um valor (via entrada de dados)

Qual valor?

- ✓ um valor pré-estabelecido, reconhecido pelo código do programa como sinalizador do fim (*flag*)

Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número

Ideia da Solução

Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. **O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número**

```
num != 0
while o usuário quiser:
```

```
print("\nBloco de um)
tabuada Comandos
```

Desenvolvendo a Solução

Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. **O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número**

Preenchido
pelo usuário

```
num != 0  
while o usuário quiser:
```

```
print("\nBloco de  
tabuada Comandos")
```

Rascunho da Solução em Python

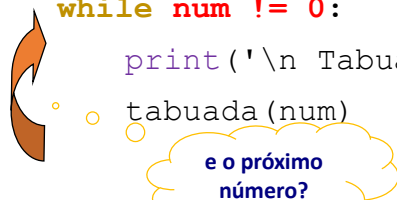
Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. **O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número**

```
num = int(input('Número? ')) #obtem 1º número  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

Solução em Python: números?

Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. **O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número**

```
num = int(input('Número? ')) #obtem 1º número
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    e o próximo número?
```



Números fora e dentro da repetição

Faça um programa que mostra na tela a tabuada de números fornecidos pelo usuário. **O término da entrada de dados ocorre quando o usuário digitar 0 como valor do número**

```
num = int(input('Número? ')) #obtem 1º número
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    num = int(input('Número? ')) #obtem próx
```


Chinês da tabuada (1/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)  
    num = int(input('Número? ')) #próximos
```

num

Chinês da tabuada (2/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)  
    num = int(input('Número? ')) #próximos
```

num

6

Chinês da tabuada (3/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)  
    num = int(input('Número? ')) #próximos
```

num

6

Chinês da tabuada (4/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)  
    num = int(input('Número? ')) #próximos
```

num

6

[illegible]

```
num = int(input('Número? ') #1º
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    num = int(input('Número? ') #próximos
```

[illegible]

```
num = int(input('Número? ') #1º
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    num = int(input('Número? ') #próximo
```

[illegible]

```
num = int(input('Número? ') #1º
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    num = int(input('Número? ') #próximos
```

[illegible]

```
num = int(input('Número? ') #1º
while num != 0:
    print('\n Tabuada do ', num)
    tabuada(num)
    num = int(input('Número? ') #próximos
```

[illegible]

num
6
2
0

num
6
2
0

num
6
2
0

Tabuada: erros possíveis (1/12)

```
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)  
    num = int(input('Número? ')) #próximos
```

NameError: name 'num' is not defined

Tabuada: erros possíveis (2/12)

```
num = int(input('Número? ')) #1º  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (3/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (4/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (5/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (6/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (7/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (8/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (9/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (10/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (11/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

Tabuada: erros possíveis (12/12)

```
num = int(input('Número? ')) #1°  
while num != 0:  
    print('\n Tabuada do ', num)  
    tabuada(num)
```

num

6

loop infinito!!!

Mão na massa: Frascos de essência

Um artesão deseja dividir aproximadamente ao meio o que sobrou de um frasco de essência com uma pipeta de 10ml. Cada vez que ele transfere do frasco original (frasco1) para o novo frasco (frasco 2), 0.35% da essência do frasco original evapora enquanto a evaporação do novo frasco é de 0.12%.

Faça uma função que receba a quantidade inicial do frasco de essência ($40 \leq qt < 230$) e mostre a quantidade final em cada um dos frascos.

Qt Orig	Qt Novo	Orig antes Evap	Novo antes Evap	Evap Orig	Evap Novo	Nova Qt Orig	Nova Qt Novo
200.00	0.00	190.00	10.00	6.65	0.12	183.35	9.88
183.35	9.88	173.35	19.88	6.07	0.24	167.28	19.64
167.28	19.64	157.28	29.64	5.50	0.36	151.78	29.29
151.78	29.29	141.78	39.29	4.96	0.47	136.82	38.81
136.82	38.81	126.82	48.81	4.44	0.59	122.38	48.23
122.38	48.23	112.38	58.23	3.93	0.70	108.44	57.53
108.44	57.53	98.44	67.53	3.45	0.81	95.00	66.72
95.00	66.72	85.00	76.72	2.97	0.92	82.02	75.80

Frascos de essência: ideia

Um artesão deseja dividir aproximadamente ao meio o que sobrou de um frasco de essência com uma pipeta de 10ml. Cada vez que ele transfere do frasco original (frasco1) para o novo frasco (frasco 2), 0.35% da essência do frasco original evapora enquanto a evaporação do novo frasco é de 0.12%.

Faça uma função que receba a quantidade inicial do frasco de essência ($40 \leq qt < 230$) e mostre a quantidade final em cada um dos frascos.

Dados necessários ?

Respostas Exibidas ?

Como Dados → Respostas ?

Frascos de essência: solução

Um artesão deseja dividir aproximadamente ao meio o que sobrou de um frasco de essência com uma pipeta de 10ml. Cada vez que ele transfere do frasco original (frasco1) para o novo frasco (frasco 2), 0.35% da essência do frasco original evapora enquanto a evaporação do novo frasco é de 0.12%.

Faça uma função que receba a quantidade inicial do frasco de essência ($40 \leq qt < 230$) e mostre a quantidade final em cada um dos frascos.

Dados necessários ? Quantidade inicial do frasco1

*Respostas Exibidas ? Quantidade final do frasco1
Quantidade inicial do frasco2*

Como Dados → Respostas ? Transferir frasco1 → frasco 2 de 10 em 10 ml

Frascos: Desenvolvendo a Solução



Tira 10 ml do frasco 1

Coloca 10 ml no frasco 2

Atualiza as quantidades em função da evaporação

QUANDO PARAR?

Frascos: repetição indeterminada

while os frascos não estiverem em equilíbrio :

Tira 10 ml do frasco 1

Coloca 10 ml no frasco 2

Atualiza as quantidades em função da evaporação

Frascos: condição de parada

A diferença entre eles ser inferior a 10ml

while os frascos não estiverem em equilíbrio :

Tira 10 ml do frasco 1

Coloca 10 ml no frasco 2

Atualiza as quantidades em função da evaporação

Frascos: código Python

```
def transfere(frasco1):
    print("%15s %15s %15s %15s %15s %15s %15s %15s"%
          ("Qt Orig","Qt Novo", "Orig antes Evap", "Novo antes
          Evap", "Evap Orig","Evap Novo", "Nova Qt Orig","Nova Qt
          Novo"))
    frasco2=0
    while(frasco1-frasco2>10):
        print("%15.2f %15.2f"%(frasco1,frasco2),end=' ')
        frasco2+=10
        frasco1-=10
        print("%15.2f %15.2f"%(frasco1,frasco2),end=' ')
        evaf1=frasco1*0.035
        evaf2=frasco2*0.012
        frasco1-=evaf1
        frasco2-=evaf2
        print("%15.2f %15.2f %15.2f %15.2f"
              %(evaf1,evaf2,frasco1,frasco2))
    return
```

Frascos: observações no código Python

```
def transfere(frasco1):
    print("%15s %15s %15s %15s %15s %15s %15s %15s"
          %("Qt Orig","Qt Novo", "Orig antes Evap", "Novo antes Evap",
          "Evap Orig","Evap Novo", "Nova Qt Orig","Nova Qt Novo"))
    frasco2=0
    while(frasco1-frasco2>10):
        print("%15.2f %15.2f"%(frasco1,frasco2),end=' ')
        frasco2+=10
        frasco1-=10
        print("%15.2f %15.2f"%(frasco1,frasco2),end=' ')
        evaf1=frasco1*0.035
        evaf2=frasco2*0.012
        frasco1-=evaf1
        frasco2-=evaf2
        print("%15.2f %15.2f %15.2f %15.2f"
              %(evaf1,evaf2,frasco1,frasco2))
    return
```

OBSERVAÇÃO

$a = a+x; \leftrightarrow a += x;$
 $a = a-x; \leftrightarrow a -= x;$
 $a = a*x; \leftrightarrow a *= x;$
 $a = a/x; \leftrightarrow a /= x;$
 $a = a\%x; \leftrightarrow a \% = x;$

O padrão do comando **print()** é mudar para uma nova linha quando termina. Para "quebrar" este padrão, use o parâmetro **end**

Frascos: função Transfere

Chamada da função transfere

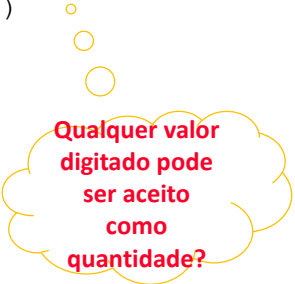
```
frasco1 = float(input("Ml no frasco original? "))  
transfere(frasco1)
```

Faça uma função que receba a quantidade inicial do frasco de essência ($40 \leq qt < 230$)

Transfere: validando dados

Chamada da função transfere

```
frasco1 = float(input("Ml no frasco original? "))  
transfere(frasco1)
```



Qualquer valor
digitado pode
ser aceito
como
quantidade?

Faça uma função que receba a quantidade inicial do frasco de essência ($40 \leq qt < 230$)

Frascos: entrada de dados validada

Chamada da função transfere

```
def qtValida():
    qt = float(input("Quantidade no frasco
original? "))
    while(qt<40 or qt >=230): # Enqto não digitar valor correto
        print('Qt inválida.Digite um valor entre 40 e 230')
        qt = float(input("Quantidade frasco original?"))
    return qt

frasco1 = qtValida()
transfere(frasco1)
```


Exercício: média de uma turma

Faça um programa que obtenha a matrícula , a nota da prova e a média dos trabalhos dos alunos de uma turma, mostrando a média final de cada um.

O término da entrada de dados ocorre quando for introduzido 0 como número de matrícula.

A média de um aluno é calculada da seguinte forma:

$$0.85 * \text{nota da prova} + 0.15 * \text{média dos trabalhos}$$

?????  *Dados* _{1Aluno}: matr, nota Prova e média Trabalhos
Respostas _{1Aluno}: matr, média

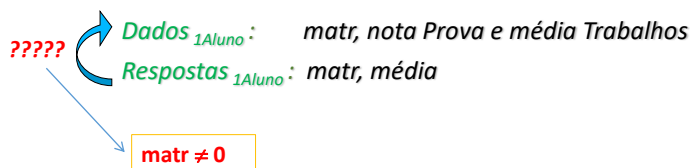
Média: ideias

Faça um programa que obtenha a matrícula, a nota da prova e a média dos trabalhos dos alunos de uma turma, mostrando a média final de cada um.

O término da entrada de dados ocorre quando for introduzido 0 como número de matrícula.

A média de um aluno é calculada da seguinte forma:

$$0.85 * \text{nota da prova} + 0.15 * \text{média dos trabalhos}$$



Média: uma solução

```
def mediaPond( nPr, mTr):  
    return nPr*0.85 + mTr*0.15  
  
matr = int(input("\tMatrícula? 0 finaliza "))  
while matr !=0:  
    nPr = float(input('\t\tNota da Prova?'))  
    mTr = float(input('\t\tMédia de Trabalhos?'))  
    med = mediaPond(nPr,mTr)  
    print("Matricula: %d - Media: %.1f"%(matr,med))  
    matr = int(input("\tMatrícula? 0 finaliza "))
```

Média: valor da nota!

```
def mediaPond(
    return

matr = int(input("\tMatrícula? 0 finaliza "))
while matr !=0:
    nPr = float(input('\t\tNota da Prova?'))
    mTr = float(input('\t\tMédia de Trabalhos?'))
    med = mediaPond(nPr,mTr)
    print("Matricula: %d - Media: %.1f"%(matr,med))
    matr = int(input("\tMatrícula? 0 finaliza "))
```

Qualquer valor digitado pode ser aceito como nota?

Média: ler somente notas válidas

```
def mediaPond( nPr, mTr):
    return nPr*0.85 + mTr*0.15

def NotaValida(mensTipoNota):
    nota = float(input('\t\t' + mensTipoNota))
    while(nota<0 or nota > 10): # Enquanto não digitar valor correto
        print('Nota inválida.Digite valor entre 0 e 10')
        nota = float(input('\t\t' + mensTipoNota))
    return nota

matr = int(input("\tMatrícula? 0 finaliza "))
while matr !=0:
    nPr = NotaValida('Nota da Prova?')
    mTr = NotaValida('Média de Trabalhos?')
    med = mediaPond(nPr,mTr)
    print("Matricula: %d - Media: %.1f"%(matr,med))
    matr = int(input("\tMatrícula? 0 finaliza "))
```

Média: modificações

```
def mediaPond( nPr, mTr):  
    return nPr*0.85 + mTr*0.15  
  
def NotaValida(mensTipoNota):  
    nota = float(input('\t\t' + mensTipoNota))  
    while(nota<0 or nota > 10): # Enqto não digitar valor correto  
        print('Nota inválida.Digite um valor entre 0 e 10')  
        nota = float(input('\t\t' + mensTipoNota))  
    return nota  
  
matr = int(input("\tMatrícula? 0 finaliza "))  
while matr !=0:  
    nPr = NotaValida('Nota da Prova?')  
    mTr = NotaValida('Média de Trabalhos?')  
    med = mediaPond(nPr,mTr)  
    print("Matricula: %d - Media: %.1f"%(matr,med))  
    matr = int(input("\tMatrícula? 0 finaliza "))
```

CALCULAR A
MÉDIA DA
TURMA!

QUAL FOI A
MAIOR MÉDIA
DA TURMA?

Jogo do “par ou ímpar”

Construa um programa para o usuário jogar par ou ímpar com o computador.

Inicialmente o programa deve perguntar ao usuário a quantidade de partidas. Para cada partida devem ser mostrados os números escolhidos e quem venceu.

No final, o programa deve mostrar quantas partidas o usuário venceu, quantas partidas o computador venceu e quem venceu o jogo.

Obs: O usuário sempre será par e deve-se usar as funções abaixo

função **chuteJog()**: pergunta ao usuário um número, retornando-o. Só aceita números entre 0 e 10

função **vencPartida(nJog,nComp)**: recebe os números da partida, verifica e exibe quem venceu, retornando 1 se o jogador venceu ou 0 caso contrário.

função **vencJogo(vJog,vComp)**: recebe a quantidade de vitórias do jogador e do computador, exibindo quem venceu o jogo ou deu empate.

Par ou ímpar: uma Solução

```
import random

def chuteJog():
    num = int(input('\t\t Jogador, escolha um número inteiro: '))
    while(num<0 or num > 10):
        print('Número inválido. Valor deve estar entre 0 e 10')
        num = int(input('\t\t Jogador, escolha um número: '))
    return num

def vencPartida(nJog,nComp):
    tot = nJog+nComp
    print('Jogador: %d x Computador: %d'%(nJog,nComp))
    if tot%2 == 0: # par?
        print('\t\t Par - Jogador venceu')
        return 1
    else:
        print('\t\t Ímpar - Computador venceu')
        return 0
```

Par ou ímpar: continuação

```
def vencJogo(vJog,vComp):
    print('Vitórias Jogador: %d x Vitórias Computador: %d'
          %(vJog,vComp))
    if vJog >vComp:
        print('\t\t Jogador venceu o jogo')
    elif vJog <vComp:
        print('\t\t Computador venceu o jogo')
    else:
        print('\t\t Empate')
    return

nPart = int(input('\t\t Jogador, quantas partidas?'))
cont=0
vJog=0
while(cont< nPart):
    nJog = chuteJog()
    nComp = random.randint(0,10)
    vJog += vencPartida(nJog,nComp)
    cont = cont+1
vencJogo(vJog,nPart-vJog)
```

Par ou ímpar:

Sempre deve-se
jogar n partidas?

```
def vencJogo(vJog,vComp):
    print('Vitórias Jogador: %d x Vit'
          % (vJog,vComp))
    if vJog >vComp:
        print('\t\t Jogador venceu o jogo')
    elif vJog <vComp:
        print('\t\t Computador venceu o jogo')
    else:
        print('\t\t Empate')
    return

nPart = int(input('\t\tJogador, quantas partidas?'))
cont=0
vJog=0
while(cont< nPart):
    nJog = chuteJog()
    nComp = random.randint(0,10)
    vJog += vencPartida(nJog,nComp)
    cont = cont+1
vencJogo(vJog,nPart-vJog)
```

Par ou ímpar para N partidas

Total de Repetições: no máximo n.

Parar quando um dos jogadores venceu metade+1 das partidas

```
nPart = int(input('\t\tJogador, quantas partidas?'))
cont=0
vJog=0
while(cont< nPart):
    nJog = chuteJog()
    nComp = random.randint(0,10)
    vJog += vencPartida(nJog,nComp)
    cont = cont+1
vencJogo(vJog,nPart-vJog)
```

Finalizando *loops* com break

Em algumas situações, é útil forçar a interrupção da iteração.

A instrução **break** força a saída do while sem testar a condição de controle do loop.

Par ou ímpar: com break

```
nPart = int(input('\t\tJogador, quantas partidas?'))
cont=0
vJog=0
meio =nPart//2
while cont< nPart :
    cont = cont+1
    nJog = chuteJog()
    nComp = random.randint(0,10)
    vJog += vencPartida(nJog,nComp)
    if vJog > meio or cont-vJog > meio: # se alguém venceu
        break                        #interrompe o jogo

vencJogo(vJog, cont-vJog)
```


Exercícios diversos: repetição

1. O vencedor do sorteio de um carro do dia por um supermercado será o primeiro consumidor cujo número de itens comprados seja múltiplo de 93. Caso não haja vencedor neste dia, o carro será doado ao estado.

Faça um programa que leia o número de itens dos participantes e verifique se houve um vencedor.

Término da entrada de dados: nº de itens == 0

2. Escreva uma função que receba strings x e s e devolva o índice da posição a partir da qual x ocorre em s ou False se não ocorrer
3. Faça uma função que exiba os números gerados pela função randint(), até que um número múltiplo de 13 seja gerado, considerando que:

- ✓ Múltiplos de 3 e 5, pares, não são somados e quando ocorrerem, um novo número deve ser gerado. Caso este seja múltiplo de 3 ou 5 o ciclo deve ser interrompido, senão é utilizado pelo programa.

Obs: No final, seu programa deve exibir a soma dos números gerados

Duas Soluções Possíveis Ex. 1

```
nItens = int(input('Qtos itens? '))
while nItens > 0 :
    if nItens%93 == 0:
        print("\nUAU!!O carro é seu")
        break
    else:
        print("\nNão foi desta vez!!!!")
        nItens = int(input('Qtos itens?'))
if nItens <=0:
    print( "\nCarro doado ao estado")
```

```
nItens = int(input('Qtos itens? '))
while nItens > 0 and nItens%93 != 0 :
    print("\nNão foi desta vez!!!!")
    nItens = int(input('Qtos itens?'))
if nItens <=0:
    print( "\nCarro doado ao estado")
else:
    print("\nUAU!!!O carro é seu")
```

Uma Solução Ex. 2

```
def mostra(x,s):
    tam=len(s)
    ult=len(x)-tam+1 #+1 para incluir a última fatia
    i=0
    while i<ult:
        if x[i:i+tam]==s:
            return i
        i=i+1
    return False

print(mostra('abacate','ate'))
print(mostra('abacate','cot'))
```

Uma Solução Ex. 3

```
import random
def soma():
    tot=0
    num=random.randint(1,300)
    while (num%13)!=0:
        if (num%3==0) and (num%5==0) and (num%2==0):
            print(num, ' desconsiderado')
            num=random.randint(1,300)
            if (num%3==0) or (num%5==0):
                break

        print(num)
        tot+=num
        num=random.randint(1,300)

    print(num)
    return tot

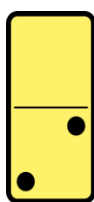
print(soma())
```

Exercício: jogo do dominó

Crie uma função para mostrar os valores das 28 peças do jogo de dominó:

0 – 0	0 – 6
0 – 1	1 – 1
0 – 2	1 – 2
0 – 3	...
0 – 4	5 – 6
0 – 5	6 – 6

Dominó: ideia da solução



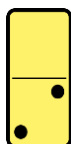
Um laço controla a
1ª metade da peça

(m1)

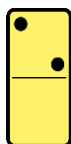
Um laço controla a
2ª metade da peça

(m2)

Problema:



=



0 – 0	1 × 0	2 × 0	3 × 0	4 × 0	5 × 0	6 × 0
0 – 1	1 – 1	2 × 1	3 × 1	4 × 1	5 × 1	6 × 1
0 – 2	1 – 2	2 × 2	3 × 2	4 × 2	5 × 2	6 × 2
0 – 3	1 – 3	2 – 3	3 – 3	4 × 3	5 × 3	6 × 3
0 – 4	1 – 4	2 – 4	3 – 4	4 – 4	5 × 4	6 × 4
0 – 5	1 – 5	2 – 5	3 – 5	4 – 5	5 – 5	6 × 5
0 – 6	1 – 6	2 – 6	3 – 6	4 – 6	5 – 6	6 – 6

```
def exhibe(m1,iniciom2):  
    while (iniciom2 <= 6):  
        print(m1,"-",iniciom2)  
        iniciom2 += 1  
    return  
  
# Controla 1a. metade do dominó  
while (m1 <= 6):  
    exhibe(m1,m1)  
    m1 = m1 + 1
```