Fundamentos da Progamação @ LEIC/LETI

Semana 01 - Aula 04

Elementos básicos de programação

Programas, instruções e sequenciação. Execução condicional. Repetição.

Alberto Abad, Tagus Park, IST, 2021

Interpretador de Python

Modo programa (*script*)

```
alberto@macal ~ $ echo "print('Hello world')" > script.py
alberto@macal ~ $ python script.py
Hello world
```

Elementos básicos de programação - Programas

Programas

- Sequência de instruções/expressões num *script*:

```
<program> ::= <definitions>* <instr_or_expres>
```

(Definições na próxima semana)

- Instruções/expressões em "linhas" diferentes do script separadas pela tecla *RETURN*:

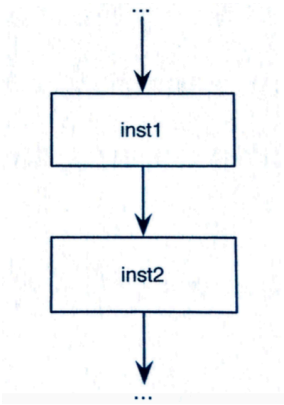
```
<instr_or_expres> :== <instruction> NEWLINE |
                      <expression> NEWLINE |
                      <instruction> NEWLINE <instr_or_expres> |
                      <expression> NEWLINE <instr_or_expres>
```

- Instrução vazia:

```
<empty instruction> :==
```

Elementos básicos de programação - Estruturas de controlo

Sequênciação



- Outras estrutas de controlo: **Seleção** e **Repetição**

Exemplo Programa: Calculadora de preço com IVA

```
valor = eval(input('Valor? '))
iva = eval(input('IVA (%)? '))
imposto = valor*iva/100
print('Valor:', valor, 'Impostos:', imposto, 'PVP:', valor + imposto)
```

In [362]:
```
nome = (input('Nome? '))
sobrenome = (input('Sobrenome? '))
print('O teu nome é:', nome, sobrenome)
```

```
Nome? Han
Sobrenome? Solo
O teu nome é: Han Solo
```

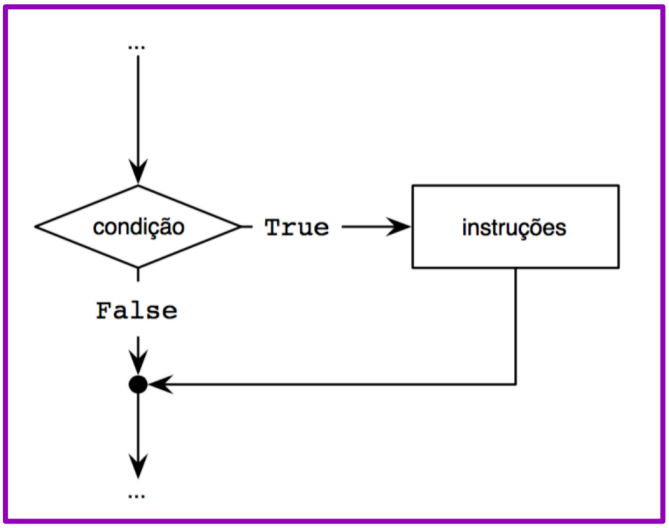

Seleção BNF

```
<if instr> ::= if <condition>: NEWLINE
               <block of instructions>
               <alternative instructions>*
               {<final alternative>}

<alternative instructions> ::= elif <condition>: NEWLINE
                               <block of instructions>

<alternative final> ::= else: NEWLINE
                        <block of instructions>

<block of instructions> ::= INDENT <instructions or expressions> DEDENT

<condition> ::= <expression>
```

- *INDENT* indentaçaõ (TAB ou espaços); *DEDENT* deidentação

Fluxograma *if*

```
if ⟨condição⟩:
    ⟨instruções⟩
```



Elementos básicos de programação - Seleção

Exemplo *if* #1

```
numero = int(input("Numero? "))
if numero % 2 == 0:
    print('Par')
print("Adeus")
```

In [ ]:

Exemplo *if* #2

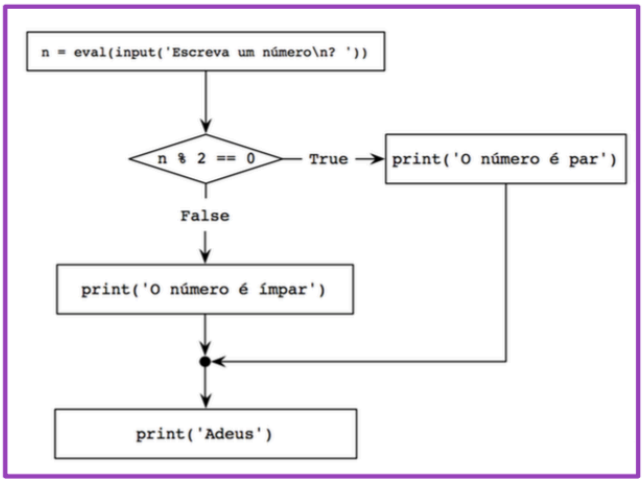
```
numero = int(input("Numero? "))
if numero % 2 == 0:
    print('Par')
    print("Adeus")
```

In [ ]:

Elementos básicos de programação - Seleção

Fluxograma *if else*

```
if ⟨cond⟩:
    ⟨instruções1⟩
else:
    ⟨instruções2⟩
```



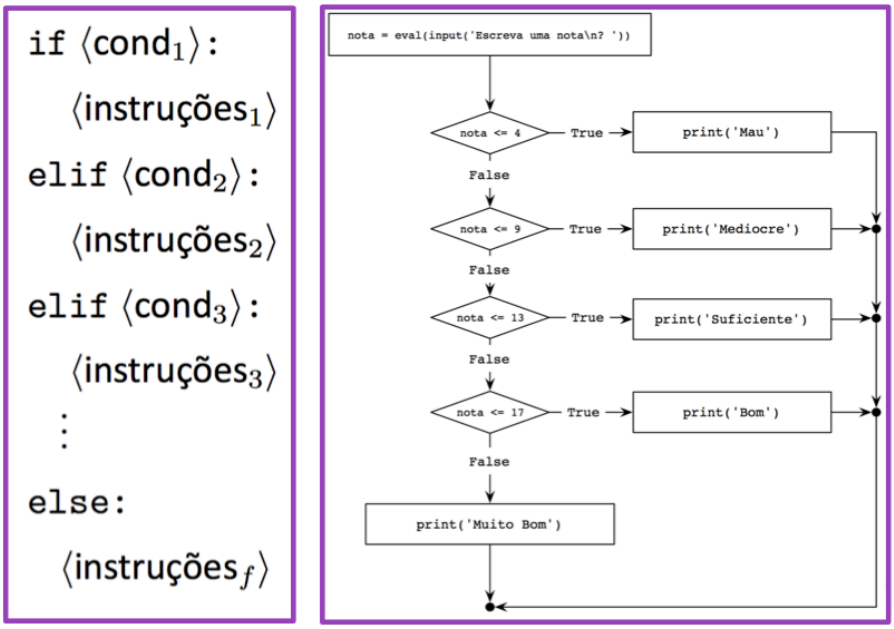
Exemplo *if* #3

```
numero = int(input("Numero? "))
if numero % 2 == 0:
    print('Par')
else:
    print('Impar')
print("Adeus")
```

In [ ]:

Elementos básicos de programação - Seleção

Fluxograma *if elif else*



Elementos básicos de programação - Seleção

Exemplo *if* #4

In [1]:
```
nota = int(input('Introduzir nota?'))

# Completar para printar mensagem de erro se o valor da nota fora i
nválido
if nota<= 4:
    print('Mau')
elif nota <= 9:
    print('Mediocre')
elif nota <=13:
    print('Suficiente')
elif nota <=17:
    print('Bom')
else:
    print('Excelente')
```

```
Introduzir nota?-4
Mau
```


Exemplo *if* #5, Algoritmo: Maior de 2 números

```
 Ler num1 e num2
se num1 > num2
    Escrever "O primeiro número é maior"
senão
    se num2 > num1
        Escrever "O segundo número é maior"
    senão
        Escrever "Os dois números são iguais"
```


Exemplo *if* #5: Algoritmo: Maior de 2 números

In [ ]:
```

```

Exemplo *if* #6: Números pares, impares, positivos e negativos

```
In [ ]: x = int(input('Introduza um número inteiro: '))
```


Repetição (*while*) BNF

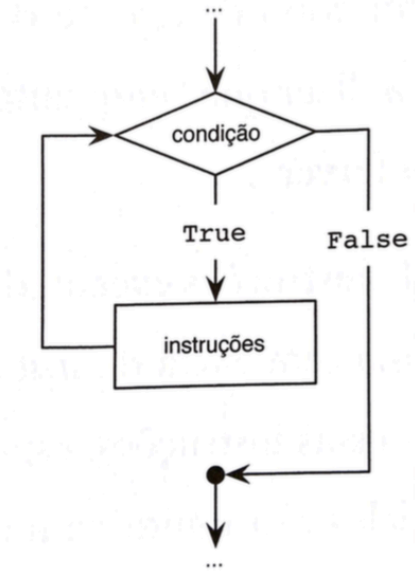
- Repetição enquanto a condição for verdadeira

```
<while instruction> ::=
   while <condition>: NEWLINE
   <block of instructions>
```

- Existem outras intruções de repetição como o `for` (próximas semanas)
- Forçar interrupção do ciclo:

```
<break instruction> ::= break
```

Fluxograma



Elementos básicos de programação - Repetição/*while*

Example *while* #1: Soma números

In [377]:
```
soma = 0
x = eval(input('Introduza um número (negativo para terminar): '))

## Completar codigo
while x >= 0:
    # soma = soma + x
    soma += x # syntactyc sugar
    x = eval(input('Introduza um número (negativo para terminar): '
))


print('Soma total:', soma)
```

```
Introduza um número (negativo para terminar): 1
Introduza um número (negativo para terminar): 2
Introduza um número (negativo para terminar): 3
Introduza um número (negativo para terminar): 4
Introduza um número (negativo para terminar): 5
Introduza um número (negativo para terminar): 6
Introduza um número (negativo para terminar): -1
Soma total: 21
```

Exemplo *while* #2: Soma números (pares e impares)

In [379]:
```
soma = 0
soma_pares = 0
soma_impares = 0
x = int(input('Introduza um número (negativo para terminar): '))

# Completar codigo
while x >= 0:
    if x % 2 == 0: # par
        soma_pares = soma_pares + x
    else:
        soma_impares = soma_impares + x

    # soma = soma + x
    x = eval(input('Introduza um número (negativo para terminar): '
))

soma = soma_pares + soma_impares

print("Soma total:", soma, "\nSoma pares:", soma_pares,"\nSoma impa
res:", soma_impares)
```

```
Introduza um número (negativo para terminar): 4
Introduza um número (negativo para terminar): 5
Introduza um número (negativo para terminar): 6
Introduza um número (negativo para terminar): 7
Introduza um número (negativo para terminar): 8
Introduza um número (negativo para terminar): 9
Introduza um número (negativo para terminar): -1
Soma total: 39
Soma pares: 18
Soma impares: 21
```

Elementos básicos de programação - Repetição/*while*

Exemplo *while* #3: Soma dos dígitos de um número

In [382]:
```
soma = 0
num = int(input("Número? "))

## Completar codigo
while True:
    d = num % 10
    soma = soma + d
    num = num // 10
    if num == 0:
        break


print(soma)
```

```
Número? 4567
22
```


Exemplo *while* #4: Cálculo dos factores primos de um número inteiro

Número	*Divisor*	*Divisível?*	*Escreve*
780	2	Sim	2
390	2	Sim	2
195	2	Não	
195	3	Sim	3
65	3	Não	
65	4	Não	
65	5	Sim	5
13	5	Não	
13	6	Não	
13	7	Não	
13	8	Não	
13	9	Não	
13	10	Não	
13	11	Não	
13	12	Não	
13	13	Sim	13
1			

Elementos básicos de programação - Repetição/*while*

Exemplo *while* #4 - Cálculo dos factores primos de um número inteiro

In [4]:
```
num = int(input("Escreva um inteiro: "))

divisor = 2
print("Fatores Primos:")
```

```
Escreva um inteiro: 6
Fatores Primos:
```

Elementos básicos de programação - Repetição/*while*

Exemplo *while* #5: Pares de divisores

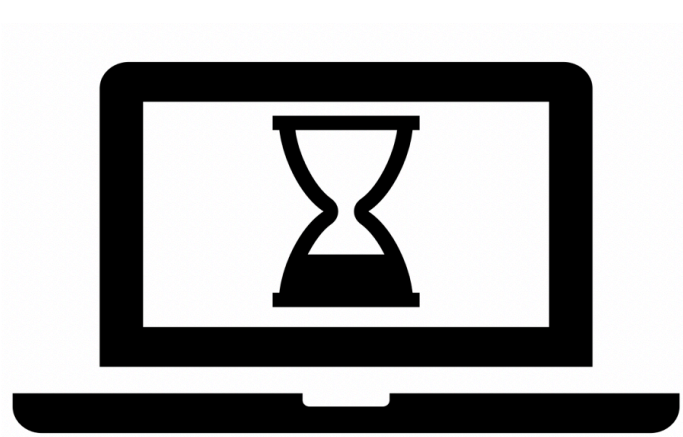
In [6]:
```
n = int(input('Introduza um número inteiro: '))
d = 1

print('Os pares de divisores de', n, 'são')

# print(1,'x',n)
```

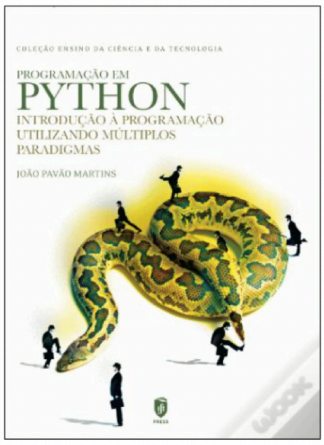
```
Introduza um número inteiro: 25
Os pares de divisores de 25 são
1 x 25
5 x 5
25 x 1
```

In [ ]:
```

```

Elementos básicos de programação - Tarefas próxima semana

- Trabalhar matéria apresentada esta semana
- Ler capítulo 3 (Funções) e 4 (Tuplos e ciclos contados) do livro da UC
- Nas aulas de problemas:
  - Mini-teste BNF no início da primeira aula (L02)
  - L02: Elementos básicos de programanção
  - L03: Funções





In [ ]: