

Introdução à Informática com Software Livre

O que é um Programa?

Um programa (em inglês, software) é uma sequência de instruções específicas que descrevem uma tarefa a ser realizada por um computador, na manipulação de dados, de maneira lógica. É como uma receita de bolo.

O que é um Programa?

Um programa (em inglês, software) é uma sequência de instruções específicas que descrevem uma tarefa a ser realizada por um computador, na manipulação de dados, de maneira lógica.

É como uma receita de bolo.

Mas para escrever um Programa é preciso de uma Linguagem de Programação.

Linguagens de Programação

Assim como nós, seres humanos, os computadores usam diferentes linguagens para se comunicar, que são chamadas de **Linguagens de Programação.**

Linguagens de Programação

Assim como nós, seres humanos, os computadores usam diferentes linguagens para se comunicar, que são chamadas de **Linguagens de Programação**.

Uma Linguagem de Programação é um meio para que seres humanos possam se comunicar com computadores.

São instruções que pessoas e computadores podem entender.

Linguagens de Programação

Algumas linguagens de programação:

-Ada

-Pascal

-BASIC

-FORTRAN

-C

-Python

-Java

-Tcl/Tk

Linguagens de Programação

Algumas linguagens de programação:

-Ada

-Pascal

-BASIC

-FORTRAN

-C

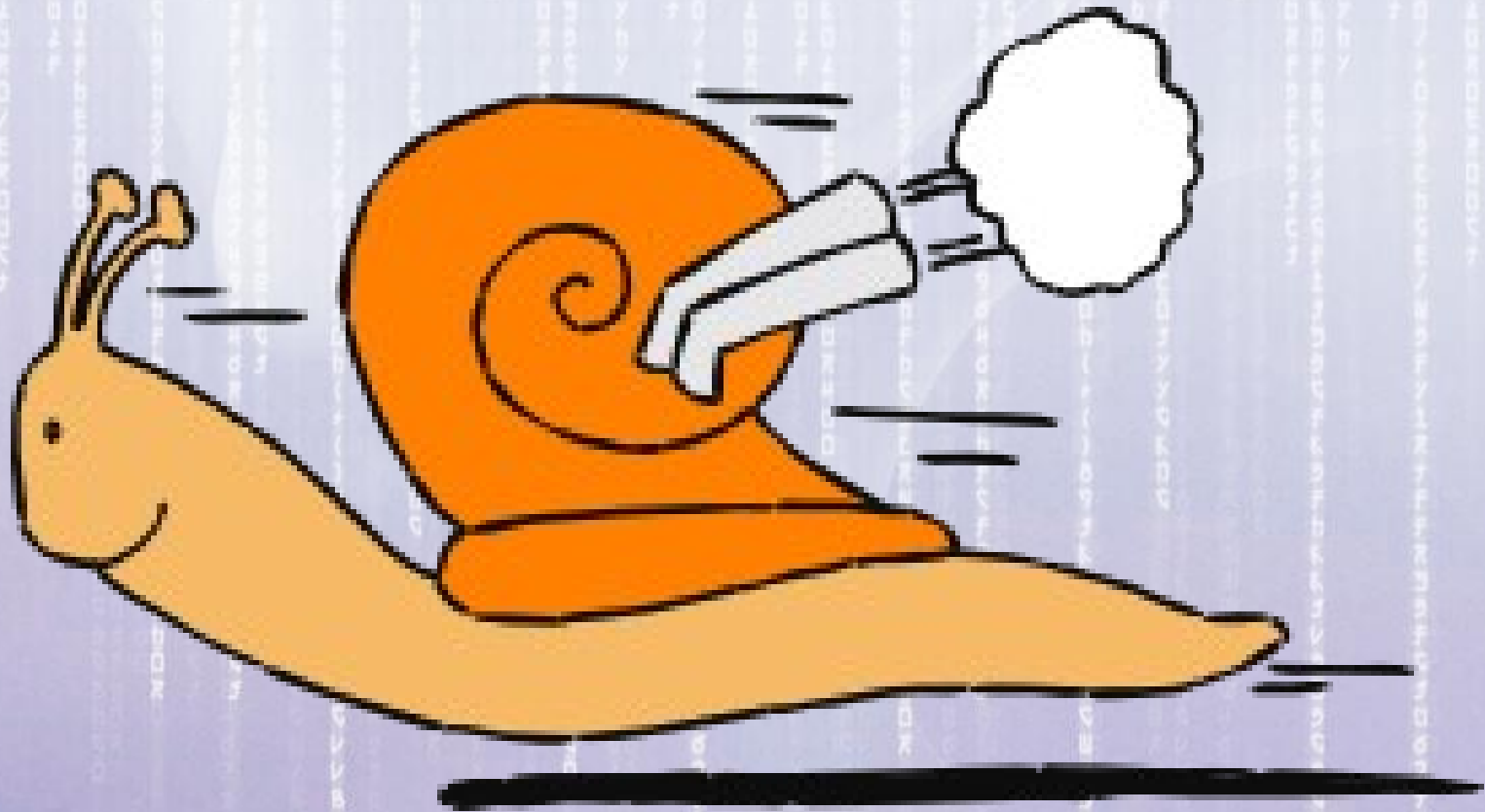
-Python

-Java

-Tcl/Tk

Gravando o Programa em um Arquivo

Como tornar mais rápido o desenvolvimento de um programa?



Gravando o Programa em um Arquivo

Para acelerar o desenvolvimento de um programa podemos gravar todos os comandos em um arquivo e executar todos os comandos deste arquivo automaticamente.



Gravando o Programa em um Arquivo

Para acelerar o desenvolvimento de um programa podemos gravar todos os comandos em um arquivo e executar todos os comandos deste arquivo automaticamente.

Mas como fazer isso?



Gravando o Programa em um Arquivo

Abrir o editor de texto em
Aplicações -> Acessórios -> Editor de Texto
e digitar no cabeçalho os comandos:

```
#!/bin/sh
```

```
#A próxima linha reinicia usando o wish \  
exec wish "$0" "$@"
```

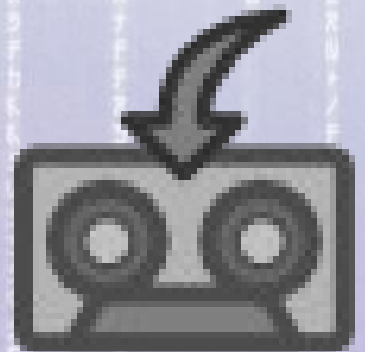
Gravando o Programa em um Arquivo

Abrir o editor de texto em
Aplicações -> Acessórios -> Editor de Texto
e digitar no cabeçalho os comandos:

```
#!/bin/sh
```

```
#A próxima linha reinicia usando o wish \  
exec wish "$0" "$@"
```

Depois salvar o arquivo com o nome
meu_programa.tcl



Executando o Programa Gravado

Para fazer o programa funcionar é necessário tornar o arquivo `meu_programa.tcl` **executável**.

Para fazer isso digite no terminal:

```
chmod +x meu_programa.tcl
```



Executando o Programa Gravado

Para fazer o programa funcionar é necessário tornar o arquivo `meu_programa.tcl` **executável**.

Para fazer isso digite no terminal:

```
chmod +x meu_programa.tcl
```

Então para executar o programa basta digitar:

```
./meu_programa.tcl
```



Recheando o Programa

Por enquanto o programa `meu_programa.tcl` só possui o cabeçalho.

Ou seja não faz nada.



Recheando o Programa

Por enquanto o programa `meu_programa.tcl` só possui o cabeçalho.

Vamos “recheiar” o programa incluindo alguns comandos (instruções)?



Recheando o Programa

Nós já aprendemos sobre **Variáveis** e como **fazer contas** com Tcl/Tk.

Agora vamos conhecer *strings* e **listas**.



String

Uma string é uma sequência de letras e números.

Uma palavra ou uma frase é uma *string*.

Para criar uma string com Tcl/Tk, basta colocar entre aspas (“”).

Vamos criar duas strings no programa meu_programa.tcl e salvar:

```
set s1 “há muitas moradas”  
set s2 “na casa de meu Pai”
```


String

Como combinar essas strings para formar uma frase?

String

Como combinar essas strings para formar uma frase?

Experimente o comando:

```
puts $s1$s2
```

ou

```
puts $s2$s1
```


String

Como combinar essas strings para formar uma frase?

Experimente o comando:

```
puts $s1$s2
```

ou

```
puts $s2$s1
```

Dica: acrescentar espaço em `s1` ou `s2`.

String

Agora armazene o conteúdo das variáveis s1 e s2 na variável s3.

E em seguida exibir o conteúdo de s3.

String

Agora armazene o conteúdo das variáveis `s1` e `s2` na variável `s3`.

E em seguida exibir o conteúdo de `s3`.

Resposta:

```
set s3 $s1$s2
```

```
puts $s3
```

Dica: acrescente um sinal sonoro no final do seu programa com o comando “`puts \a`”

Fazendo o programa *Ler* uma string

Para fazer o seu programa ler uma string.

incluir os comandos no arquivo meu_programa.tcl

```
puts "Por favor, digite o seu nome:"
```

```
gets -stdin nome
```

```
puts "Bom dia $nome"
```


Listas

Uma lista é uma sequência ordenada de elementos tais como strings, outras listas, e outras variáveis.

Vamos criar uma lista com times de futebol?



Listas

Para criar uma lista, você pode usar o comando **list**:

```
set times [list flamengo botafogo vasco fluminense]
```

ou

```
set times [list \  
  flamengo \  
  botafogo \  
  vasco \  
  fluminense  
]
```

E depois veja o conteúdo da lista **times**

Tamanho da Lista

O comando **length** retorna o número de elementos de uma lista.

Vamos guardar o número de elementos da lista `times` na variável `numero_times`:

```
set numero_times [length $times]
```

Tamanho da Lista

O comando **length** retorna o número de elementos de uma lista.

Vamos guardar o número de elementos da lista `times` na variável `n_times`:

```
set n_times [length $times]
```

Depois execute o comando:

```
puts "Oi $nome. A lista contém $n_times times"
```


Adicionando elementos na Lista

O comando **lappend** adiciona elementos a uma lista.

Vamos adicionar um time na nossa lista de times?

Adicionando elementos na Lista

O comando **lappend** adiciona elementos a uma lista.

Vamos adicionar um time na nossa lista de times?

Então adicione os comandos ao seu programa:

```
lappend times guarani
```

```
puts "Lista depois do comando lappend: $times"
```


Posição dos Elementos na Lista

O comando **lindex** permite obter um único elemento da lista.

Como obter apenas o primeiro e o terceiro elemento a lista de times?

Dica: Os elementos da lista são numerados a partir do “0”.

Posição dos Elementos na Lista

O comando **lindex** permite obter um único elemento da lista.

```
set time1 [lindex $time 0]
```

```
set time3 [lindex $time 2]
```

```
puts "Primeiro time da lista: $time1"
```

```
puts "Terceiro time da lista: $time2"
```