Introdução à Programação de Computadores para Biologia

Algoritmos e Introdução ao Perl

Aula 04

https://tttorres.github.io/introprog2024/

ALGORITMOS

O que são?

> "Uma série de instruções para a resolução de um problema"

ALGORITMOS

O que são?

> "Uma série de instruções para a resolução de um problema"

- Exemplo I:
    - Problema: acordar de manhã.

ALGORITMOS

O que são?

> "Uma série de instruções para a resolução de um problema"

- Exemplo I:
  - Problema: acordar de manhã.
  - Solução: descrição dos passos necessários para essa árdua tarefa!

ALGORITMOS

EXEMPLO I: Algoritmo matinal

```
1. Acordar;
2. Se o despertador não tocou,
        2.1) voltar a dormir;
3. Se o despertador tocou,
        3.1) apertar o botão soneca;
4. Repetir o passo 3 por "n" vezes;
5. Levantar e caminhar até o banheiro;
6. Escovar os dentes;
7. Ligar a cafeteira;
8. Se não houver uma xícara limpa;
        8.1) ir ao passo 19;
9. Se houver uma xícara limpa,
        8.2) Colocar água na xícara;
10. Adicionar a água na cafeteira;
11. Repetir os passos 9 e 10 até que o nível na cafeteira seja = a "l"
12. Pegar o café ou capsula de café;
13. Adicionar o café a cafeteira;
14. Repetir o passo 13 por “m” vezes, se não for uma cafeteira de cápsula;
15. Colocar a xícara embaixo da saída do café;
16. Apertar o botão com o desenho da xícara;
17. Esperar 1 min até a saída completa do café;
18. Saborear o café;
19. Desligar a cafeteira;
21. Repetir a função 6;
22. Sair;
```

ALGORITMOS

O que são?

> "Uma receita para a resolução de um problema"

ALGORITMOS

O que são?

> "Uma receita para a resolução de um problema"

- Exemplo II:
    - Problema: preparar um bolo de cenoura
    - Solução: descrição da receita

ALGORITMOS

EXEMPLO II: Bolo de Cenoura

```
1. Separar os seguintes ingredientes:
        1.1) 3 cenouras médias raladas;
        1.2) 4 ovos;
        1.3) 1/2 xícara (chá) de óleo;
        1.4) 2 xícaras (chá) de açúcar;
        1.5) 2 1/2 xícaras (chá) de farinha de trigo;
        1.6) 1 colher (sopa) de fermento em pó;
2. Ligar o forno para pré aquecimento a 180oC;
3. Bater no liquidificador as cenouras, os ovos e o óleo;
4. Transferir o resultado para uma vasilha;
5. Juntar aos poucos a farinha e o açúcar;
6. Misturar bem;
7. Misturar o fermento suavemente com uma colher;
8. Se temperatura do forno estiver a 180oC:
        8.1) Coloque o bolo no forno;
9. Senão, esperar até atingir 180oC e repetir o passo 8;
10. Assar o bolo até que, ao espetar com um palito, o palito fique
    limpo;
11. Tirar do forno;
12. O bolo está pronto.
```

ALGORITMOS

EXEMPLO II: Bolo de Cenoura

- Objetos de consumo (entrada):
  - cenouras
  - ovos
  - farinha
  - óleo
  - açucar
  - fermento
- Objetos de apoio (atores, executores):
  - vasilha
  - liquidificador
  - xícara
  - colher de sopa
  - fogão
  - palito
  - cozinheiro(a)

ALGORITMOS

EXEMPLO II: Bolo de Cenoura

- Objeto produzido (saída):
    - bolo
- Objeto que descreve o processo (receita):
    - Algoritmo

ALGORITMOS

Características

1. É formado por um texto finito;
    - receita dada.
2. É composto por instruções elementares;
    - elementar depende do contexto.
3. É uma receita metódica, passo-a-passo;
    - passo inicial;
    - passo(s) intermediário(s);
    - passo final.
4. Ao executar:
    - partindo de dados válidos, deve sempre terminar;
    - partindo de dados inválidos, pode produzir lixo, ou mesmo não terminar.

ALGORITMOS

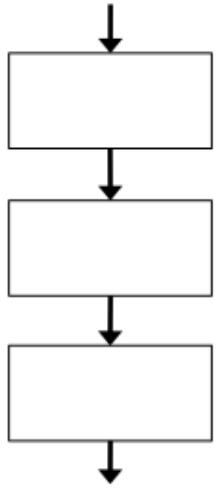
Características

> O enfoque deste curso está nos algoritmos computacionais, ou seja, algoritmos que “descrevem uma sequência de ações que podem ser traduzidas para alguma linguagem de programação”

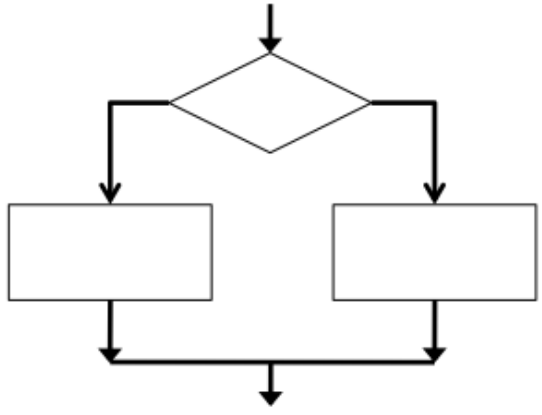
ALGORITMOS

Representações

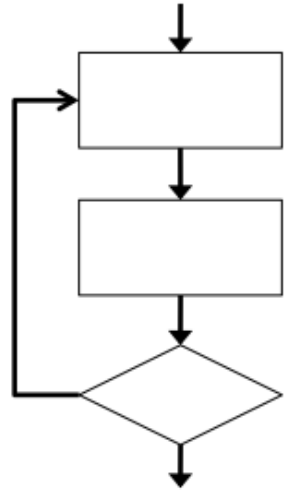
1. Linguagem escrita (português);
    - exemplos I e II.
2. Fluxogramas;
    - representação gráfica.



Estrutura de controle sequencial

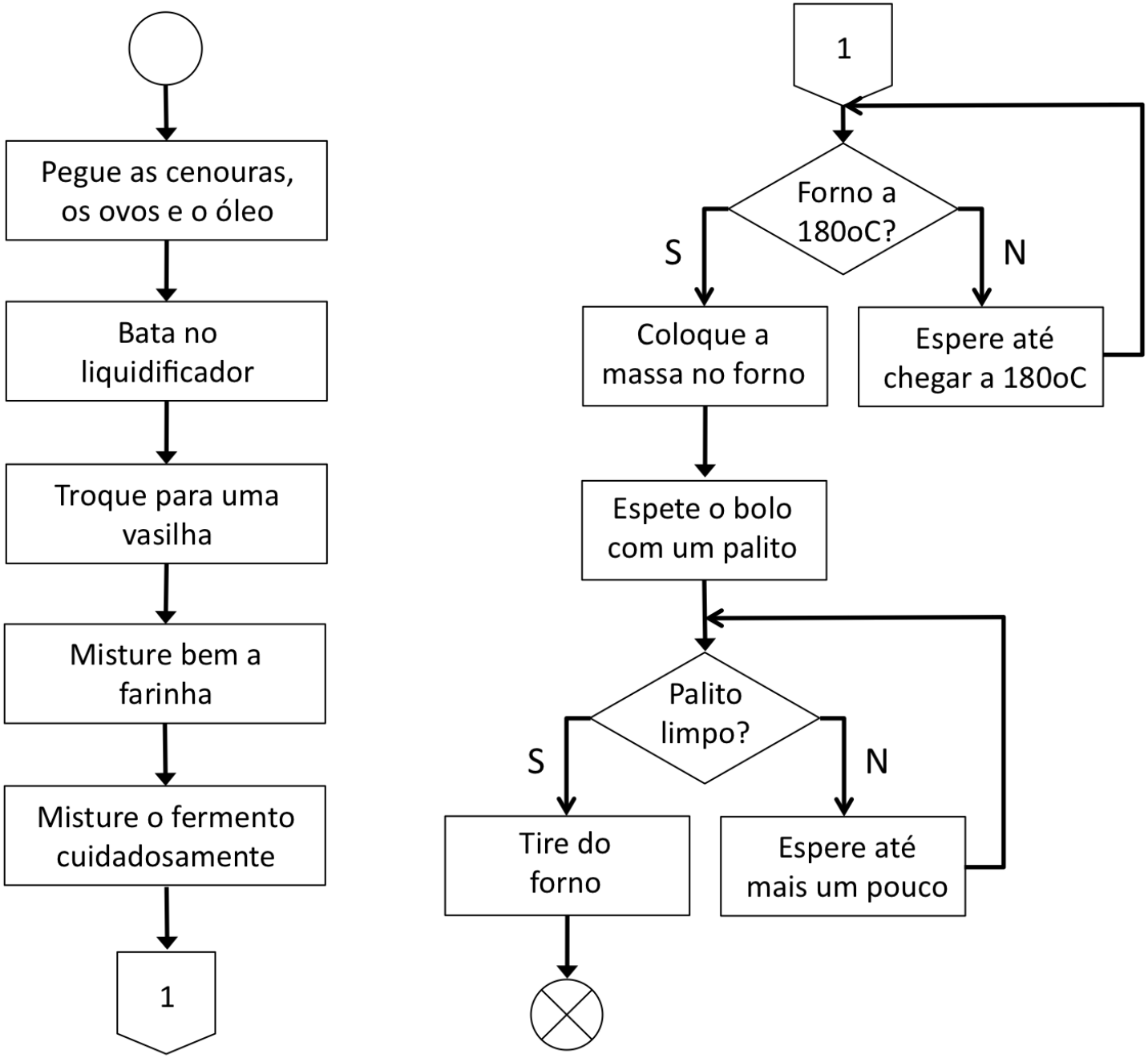


Estrutura de controle condicional



Estrutura de controle repetitiva

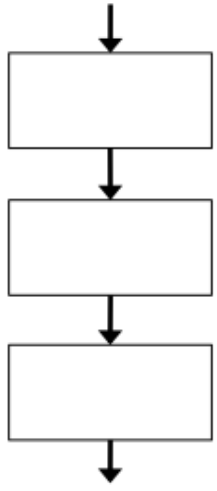
Representações: bolo de cenoura

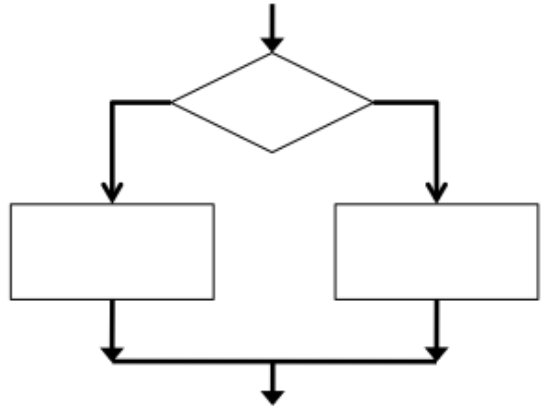


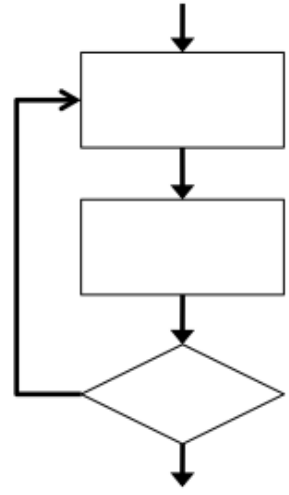
ALGORITMOS

Representações

1. Linguagem escrita (português);
    - exemplos I e II.
2. Fluxogramas;
    - representação gráfica.







3. Linguagens de programação

PERL

“Practical Extraction and Reporting Language”

PERL

“Practical Extraction and Reporting Language”

> "the Swiss Army knife of scripting languages"

> "duct tape that holds the Internet together,"

PERL

“Practical Extraction and Reporting Language”

> Pathologically Eclectic Rubbish Lister
> **Larry Wall**

- Linguagem interpretada;
- muito utilizada em bioinformática;
- menos regras e múltiplas formas de resolver um problema.

PERL

“Practical Extraction and Reporting Language”

> Pathologically Eclectic Rubbish Lister
> **Larry Wall**

- Linguagem interpretada;
- muito utilizada em bioinformática;
- menos regras e múltiplas formas de resolver um problema.

“For programmers, laziness is a virtue.”

PERL

“Practical Extraction and Reporting Language”

> Aaaain, mas por que não aprendemos Python?
>
> **Aluno**

- muito utilizada em bioinformática;
- menos regras e múltiplas formas de resolver um problema;
- expressões regulares

PERL

Geany IDE

1. Verificar a instalação do Geany.
2. Se não estiver instalado, fazer o download em: geany.org

PERL

Primeiro Programa

1. No Geany, File > New.
2. File > Save as...
3. Criar a pasta "introprog" no diretório home/aluno (no WSL criar a pasta em /mnt/c/Users/aluno/)
4. Gravar arquivo como "hello.pl"

PERL

Primeiro Programa

5. No arquivo criado escreva:

```
print "Hello world!\n";
```

6. File > Save (ou ctrl+S)

7. No terminal escreva:

```
perl ~/home/aluno/hello.pl
```

👍 👏 🎉

PERL

Primeiro Programa

8. No arquivo hello.pl escreva:

```
# Uhu! Meu primeiro script!
print "Hello world!\n";
```

9. File > Save (ou ctrl+S)

10. No terminal escreva:

```
perl ~/home/aluno/hello.pl
```

PERL

Primeiro Programa

No arquivo hello.pl escreva:

```
#! /usr/bin/perl

# Uhu! Meu primeiro script!
print "Hello world!\n";
```

No terminal (pasta `home/aluno`) escreva:

```
ls -la
```

```
chmod 777 hello.pl
```

```
ls -la
```

```
./hello.pl
```

```
#################################################################
#  PROGRAM: notas.pl                                    25.07.2012
#
#  AUTHOR: Tatiana Torres
#
#  LAST MODIFIED: 17.04.2018
#################################################################
```

Cabeçalho

```
#!/usr/bin/perl
```

Shebang

```
my ($E, $P, $M, $nota_final);
```

Declarações

```
($E, $P, $M) = @ARGV;
$nota_final = ((2*$E)+(3*$P)+(5*$M))/10;
if ($nota_final >= 5) {
   print "Aluno aprovado\n";
} else {
   print "Aluno reprovado\n";
}
```

Instruções

```
# conversao de nota para conceitos, apenas para a Pos-graduacao
```

Declarações

```
if ($nota_final < 5) {
   print "R, Reprovado, sem direito a credito\n";
} elsif ($nota_final <= 7.0) {
   print "C, Regular, com direito a credito\n";
} elsif ($nota_final <= 8.5) {
   print "B, Bom, com direito a credito\n";
} else {
   print "A, Excelente, com direito a credito\n";
}

exit;
```

Pontuação

Introdução à Programação de Computadores para Biologia

Tipos de dados

PERL

Primeiro Programa

Arquivo hello.pl:

```
#! /usr/bin/perl

# Uhu! Meu primeiro script!
print "Hello world!\n";
```

No terminal:

```
$ perl hello.pl Sao Paulo
Hello Sao Paulo!
$
```

TIPOS DE DADOS

Variáveis

1. Escalares ($):

```
my $variavel_escalar = 1;
my $cidade = "Sao Paulo";
my $sequencia = "ATCCTACTGTGCGTCAGGCTAAGCTA";
```

2. Arrays, vetores (@):

```

```

3. Hashes, vetores associativos (%):

```

```

VARIÁVEIS ESCALARES

Variáveis

1. Nomes precedidos de "$":

```
my $cidade = "Sao Paulo";       #correto
my $ cidade = "Sao Paulo";      #incorreto
```

2.

```

```

VARIÁVEIS ESCALARES

Variáveis

1. Nomes precedidos de "$":

```
my $cidade = "Sao Paulo";       #correto
my $ cidade = "Sao Paulo";      #incorreto
```

2. Nomes podem conter uma ou mais letras "A-Z" ou "a-z" incluindo "_" e depois dela(s) números:

```
my $v = 1;              #correto
my $var = 1;            #correto
my $var1 = 1;           #correto
my $var2 = 2;           #correto
my $var_1 = 1;          #correto
my $var 1 = 1;          #incorreto
my $1var = 1;           #incorreto
my $2var = 2;           #incorreto
my $variavel_escalar_criada_para_armazenar_um_numero_real = 1;
```

VARIÁVEIS ESCALARES

Variáveis

3. Variáveis pré-definidas:

```
$_
$1
$2
$^
$/
$\
$,
```

4.

```

```

VARIÁVEIS ESCALARES

Variáveis

3. Variáveis pré-definidas:

```
$_
$1
$2
$^
$/
$\
$,
```

4. Perl diferencia maiúsculas e minúsculas.

```
my $VAR = 1;
my $VAr = 2;
my $Var = 3;
my $var = 4;
```

VARIÁVEIS ESCALARES

Atribuição

Operador de atribuição, "="

```
$var = 1; # atribuindo 1 a variavel $var
```

VARIÁVEIS ESCALARES

Atribuição

Operador de atribuição, "="

```
$var = 1; #atribuindo 1 a variavel $var

1 = $var;
```

VARIÁVEIS ESCALARES

Atribuição

Operador de atribuição, "="

```
$var = 1;    #atribuindo 1 a variavel $var

1 = $var;    #incorreto
```

VARIÁVEIS ESCALARES

Atribuição

Operador de atribuição, "="

```
$var = 1;     #atribuindo 1 a variavel $var

1 = $var;     #incorreto

$greetings = "Hello world!";
```

VARIÁVEIS ESCALARES

Atribuição

Operador de atribuição, "="

```
$var = 1;     #atribuindo 1 a variavel $var

1 = $var;     #incorreto

$greetings = "Hello world!";

$location = <STDIN>;
```

VARIÁVEIS ESCALARES

hello.pl

1. Abrir o Geany, arquivo hello.pl.
2. Copiar #exemplo01 da página da disciplina.
3. No terminal:

```
perl ~/home/aluno/hello.pl
```

VARIÁVEIS ESCALARES

Impressão

Comando *print*

```
#! /usr/bin/perl

# Meu primeiro script
print "Hello world!\n";
```

VARIÁVEIS ESCALARES

Impressão

Comando *print*

```
#! /usr/bin/perl

# Meu primeiro script
print "Hello world!\n";

# Imprimir diretamente o conteudo da variavel
$greetings = "Hello world!";
print $greetings;
```

No terminal:

```
perl ~/home/aluno/hello.pl
```

VARIÁVEIS ESCALARES

Impressão

Comando *print*

```
#! /usr/bin/perl

# Meu primeiro script
print "Hello world!\n";

#exemplo02 – COPIAR DA PAGINA
# Imprimir o conteudo da variavel como parte de uma
# sentença maior
```

VARIÁVEIS ESCALARES

Impressão

Comando *print*

```
#! /usr/bin/perl

# Meu primeiro script
print "Hello world!\n";

#exemplo02
# Imprimir o conteudo da variavel como parte de uma
# sentença maior

print "Eu não suporto mais esse exemplo do $greetings\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

1. No Geany, File > New File.
2. Copiar #exemplo03 da página da disciplina.
3. File > Save as...
4. Gravar arquivo como interpolacao.pl

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

```
#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante
my $greetings = "Hello world!";

# imprimindo
print "$greetings, mais uma vez!\n";


exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

```
#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante
my $greetings = "Hello world!";

# imprimindo
print "$greetings, mais uma vez!\n";
print '$greetings, mais uma vez!\n';

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

Copiar exemplo04 da página da disciplina

```
# exemplo04

#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante

my $greetings = "Hello world!";

# imprimindo
print 'O nome da nossa variavel eh $greetings';
print " e o conteudo dela eh $greetings.\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

E como imprimir "Hello world!" (**COM as aspas**)?

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

E como imprimir "Hello world!" (**COM as aspas**)?

```
# exemplo04

#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante

my $greetings = "Hello world!";

# imprimindo
print 'O nome da nossa variavel eh $greetings';
print " e o conteudo dela eh "$greetings".\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

E como imprimir "Hello world!" (**COM as aspas**)?

Escapando da interpolação:

```
# exemplo04

#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante

my $greetings = "Hello world!";

# imprimindo
print 'O nome da nossa variavel eh $greetings';
print " e o conteudo dela eh \"$greetings\".\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

Copiar exemplo05 na página da disciplina

Escapando da interpolação:

```
# exemplo04

#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante

my $greetings = "Hello world!";

# imprimindo
print 'O nome da nossa variavel eh $greetings';
print " e o conteudo dela eh \"$greetings\".\n";

# exemplo05
print "Ah, se eu ganhasse R$1,00 a cada vez que \"$greetings\" fosse
usado...\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: interpolacao.pl

Copiar exemplo05 na página da disciplina

Escapando da interpolação:

```
# exemplo04

#! /usr/bin/perl
# script para testar interpolacao

# declarando minha variavel constante

my $greetings = "Hello world!";

# imprimindo
print 'O nome da nossa variavel eh $greetings';
print " e o conteudo dela eh \"$greetings\".\n";

# exemplo05
print "Ah, se eu ganhasse R\$1,00 a cada vez que \"$greetings\" fosse us

exit;
```

VARIÁVEIS ESCALARES

Na prática

1. No Geany, File > New File.
2. Copiar #exemplo06 da página da disciplina.
3. File > Save as...
4. Gravar arquivo como tabela.pl

VARIÁVEIS ESCALARES

Na prática

Script: tabela.pl

Exemplo 06:

```
#! /usr/bin/perl
# script para criar uma tabela

# titulo e header
print "Notas da disciplina de Introducao a Programacao\n\n";
print "No USP\tNome\tNota\n";

# imprimindo
print "0001\tMaricotinha\t9,8\n";
print "0002\tJoazinho\t2,0\n";
print "0003\tJujubinha\t9,0\n";
print "0004\tJuquinha\t3,5\n";
print "0005\tMariazinha\t9,5\n";
print "0006\tPedrinho\t2,8\n";

exit;
```

VARIÁVEIS ESCALARES

Caracteres especiais para formatação

CARACTER	FUNÇÃO
\n	Newline
\t	Tabulação (tab)
\u ou \U	Força letras maiúsculas (uppercase) para o primeiro (\u) ou para todos os próximos caracteres (\U)
\l ou \L	Força letras minúsculas (lowercase) para o primeiro (\l) ou para todos os próximos caracteres (\L)
\E	Delimita o final do \U ou \L

VARIÁVEIS ESCALARES

Na prática

Script: tabela.pl

Copiar #exemplo07 da página da disciplina:

```
print "0003\tJujubinha\t9,0\n";
print "0004\tJuquinha\t3,5\n";
print "0005\tMariazinha\t9,5\n";
print "0006\tPedrinho\t2,8\n";

# continuacao do script para criar uma tabela

# alunas aprovadas
$aluna01 = "Maricotinha";
$aluna02 = "Jujubinha";
$aluna03 = "Mariazinha";

print "\nParabens as alunas $aluna01, $aluna02 e $aluna03!\n\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: tabela.pl

Copiar #exemplo07 da página da disciplina:

```
print "0003\tJujubinha\t9,0\n";
print "0004\tJuquinha\t3,5\n";
print "0005\tMariazinha\t9,5\n";
print "0006\tPedrinho\t2,8\n";

# continuacao do script para criar uma tabela

# alunas aprovadas
$aluna01 = "Maricotinha";
$aluna02 = "Jujubinha";
$aluna03 = "Mariazinha";

print "\nParabens as alunas \U$aluna01, $aluna02 e $aluna03!\n\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: tabela.pl

Copiar #exemplo07 da página da disciplina:

```
print "0003\tJujubinha\t9,0\n";
print "0004\tJuquinha\t3,5\n";
print "0005\tMariazinha\t9,5\n";
print "0006\tPedrinho\t2,8\n";

# continuacao do script para criar uma tabela

# alunas aprovadas
$aluna01 = "Maricotinha";
$aluna02 = "Jujubinha";
$aluna03 = "Mariazinha";

print "\n\UParabens as alunas $aluna01, $aluna02 e $aluna03!\n\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: tabela.pl

Copiar #exemplo07 da página da disciplina:

```
print "0003\tJujubinha\t9,0\n";
print "0004\tJuquinha\t3,5\n";
print "0005\tMariazinha\t9,5\n";
print "0006\tPedrinho\t2,8\n";

# continuacao do script para criar uma tabela

# alunas aprovadas
$aluna01 = "Maricotinha";
$aluna02 = "Jujubinha";
$aluna03 = "Mariazinha";

print "\n\UParabens as alunas\E $aluna01, $aluna02 e $aluna03!\n\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

1. No Geany, File > New File.
2. Copiar #exemplo08 da página da disciplina.
3. File > Save as...
4. Gravar arquivo como boanoite.pl

VARIÁVEIS ESCALARES

Na prática

Script: boanoite.pl

```
#! /usr/bin/perl

# Perguntar o nome do usuário
print "Qual seu nome?\n";

# Entrada do usuário
my $nome = <STDIN>;

# Cumprimentar o usuário
print "Boa noite, $nome!\n";

exit;
```

VARIÁVEIS ESCALARES

Na prática

Script: boanoite.pl

No terminal:

```
Darwin:Introprog Tatiana$ ./boanoite.pl
Qual seu nome?
Tatiana
Boa noite, Tatiana
!
Darwin:Introprog Tatiana$
```

VARIÁVEIS ESCALARES

Comando *chomp*

Remove o último caracter se ele for um *newline*

Script: boanoite.pl

```
#! /usr/bin/perl

# Perguntar o nome do usuário
print "Qual seu nome?\n";

# Entrada do usuário
my $nome = <STDIN>;

# Remover newline
chomp($nome);

# Cumprimentar o usuário
print "Boa noite, $nome!\n";
```

VARIÁVEIS ESCALARES

Na prática

Script: boanoite.pl

No terminal:

```
Darwin:Introprog Tatiana$ ./boanoite.pl
Qual seu nome?
Tatiana
Boa noite, Tatiana!
Darwin:Introprog Tatiana$
```

VARIÁVEIS ESCALARES

Valores de Escalares

VARIÁVEIS ESCALARES

Valores de Escalares

PERL tem dois tipos básicos de escalares:

1. Números:

```
$y=1;          # inteiro positivo
$z=-5;         # inteiro negativo
$x = 3.14;     # real em ponto flutuante
$w = 2.75E-6; # real em notação científica
$t = 0377;     # octal
$u = 0xffff;   # hexadecimal
```

2. Strings:

```
$string1 = "Oi, eu sou uma string!";      # string
$string2 = 'Oi, eu tb sou uma string!';   # string
$string3 = "ATCGATCGATCGATCGATTGGATC";    # string
```

VARIÁVEIS ESCALARES

Valores

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como escalares.pl
4. Criar um script que produza a seguinte saída

VARIÁVEIS ESCALARES

Valores

Script: escalares.pl

Output:

```
$y = 1
$z = -5
$x = 3.14
$w = 2.75e-06
$t = 255
$u = 65535

Oi, eu sou uma string!
Oi, eu tb sou uma string
ATCGATCGATCGATCGATTGGATC
```

$t = atribuir um valor octal = 0377

$u = atribuir um valor hexadecimal = 0xffff

VARIÁVEIS ESCALARES

Valores

Script: escalares.pl

Output:

```
$y = 1
$z = -5
$x = 3.14
$w = 2.75e-06
$t = 255
$u = 65535

Oi, eu sou uma string!
Oi, eu tb sou uma string
ATCGATCGATCGATCGATTGGATC
```

$t = atribuir um valor octal = 0377

$u = atribuir um valor hexadecimal = 0xffff

VARIÁVEIS ESCALARES

Valores

Script: escalares.pl; exemplo09 da página

```
#! /usr/bin/perl

# atribuindo valores as variaveis
$y = 1;         # inteiro positivo
$z = -5;        # inteiro negativo
$x = 3.14;      # real em ponto flutuante
$w = 2.75e-6; # real em notação científica
$t = 0377;      # octal
$u = 0xffff;  # hexadecimal

$string1 = "Oi, eu sou uma string!";     # string
$string2 = 'Oi, eu tb sou uma string';  # string
$string3 = "ATCGATCGATCGATCGATTGGATC";  # string
```

VARIÁVEIS ESCALARES

Valores

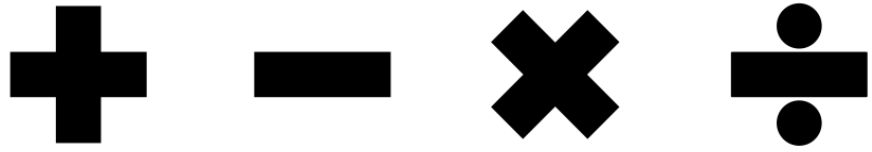
Script: escalares.pl; exemplo09 da página

```
#continuacao

# imprimindo
print "\$y \= $y\n";
print "\$z \= $z\n";
print "\$x \= $x\n";
print "\$w \= $w\n";
print "\$t \= $t\n";
print "\$u \= $u\n\n";

print "$string1\n$string2\n$string3\n\n";

exit;
```

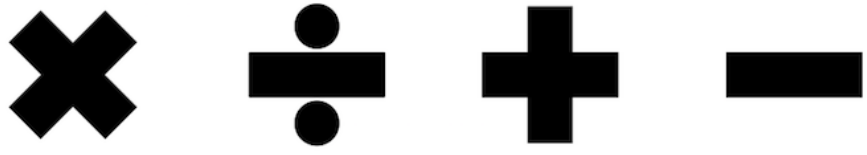
VARIÁVEIS ESCALARES

Operações com números

1. Os "operadores da escola" estão disponíveis:



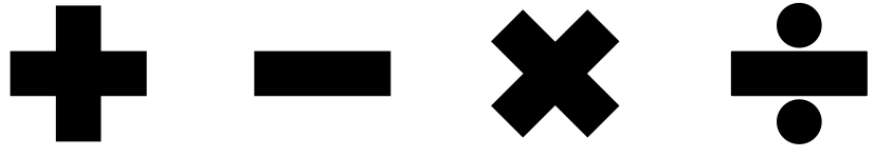
2. Precedência: praticamente igual "da escola"



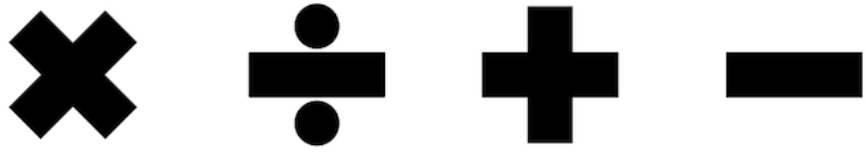
VARIÁVEIS ESCALARES

Operações com números

1. Os "operadores da escola" estão disponíveis:



2. Precedência: praticamente igual "da escola"

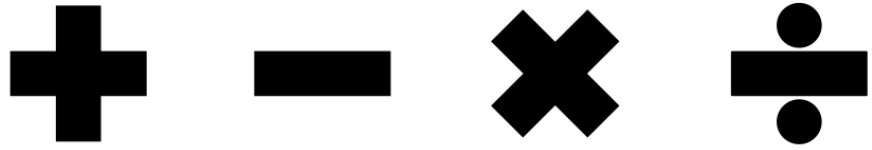


2+3*4

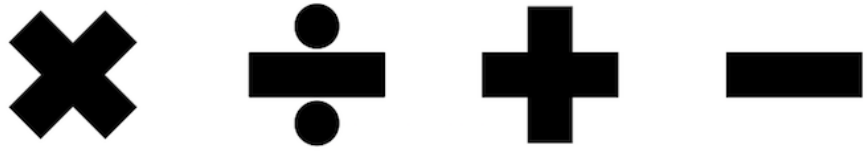
VARIÁVEIS ESCALARES

Operações com números

1. Os "operadores da escola" estão disponíveis:



2. Precedência: praticamente igual "da escola"



Sempre use parênteses!!!

VARIÁVEIS ESCALARES

Operações com números

CARACTER	FUNÇÃO
+	Adição
=	Atribuição
+=	Atribuição após soma
-=	Atribuição após subtração
++	Auto-acréscimo
--	Auto-decréscimo
/	Divisão
%	Módulo (Resto da divisão)
*	Multiplicação
**	Potenciação (Exponenciação)
sqrt()	Raiz quadrada
-	Subtração

VARIÁVEIS ESCALARES

Operações com números

1. No Geany, File > New File.
2. File > Save as...
3. Copiar #exemplo10 na página da disciplina.
4. Gravar arquivo como operacoes.pl.

VARIÁVEIS ESCALARES

Operações com números

Script: operacoes.pl

```
#! /usr/bin/perl
# script para testar operacoes matematicas

# testando
$a = 1;
print "Atribuicao\:                \$a \= $a\n";

#++$a;
#print "Auto\-acrescimo\:           \$a \= $a\n";

#--$a;
#print "Auto\-decrescimo\:          \$a \= $a\n";

#$b = 3 + 1;
#print "Soma\:                     \$b \= $b\n";

#$c = $a + $b;
#print "Soma\:                     \$c \= $c\n";
```

VARIÁVEIS ESCALARES

Operações com números

Script: operacoes.pl

Output:

```
Atribuicao:                $a = 1
Auto-acrescimo:            $a = 2
Auto-decrescimo:           $a = 1
Soma:                      $b = 4
Soma:                      $c = 5
Multiplicacao:             $d = 20
Divisao:                   $e = 5
Raiz quadrada:             $f = 2
Equacao:                   $g = 25
Modulo:                    $h = 1
Modulo:                    $i = 0
Potenciacao:               $j = 25
Adicao e atribuicao:       $j = 30
Subtracao e atribuicao: $j = 25
```