

# Introdução à Programação de Computadores para Biologia

## Tipos de dados

Aula 05a

<https://ttdorres.github.io/introprog2021/>

# TIPOS DE DADOS EM PERL

## Variáveis

### 1. Escalares (\$):

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

### 2. Arrays, vetores (@):

### 3. Hashes, vetores associativos (%):

# TIPOS DE DADOS EM PERL

## Variáveis

### 1. Escalares (\$):

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

### 2. Arrays, vetores (@):

```
@genes = ("CG7856", "scpr-B", "CG4294", "Sgt", "CG42308");
```

### 3. Hashes, vetores associativos (%):

```
%genes = ("FBgn0033056" => "CG7856",  
          "FBgn0037888" => "scpr-B",  
          "FBgn0034742" => "CG424",  
          "FBgn0032640" => "Sgt")
```

# 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

## 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 = "ATCGATCGATCGATTGGATC";      # string
```

# VARIÁVEIS ESCALARES

## Valores de Escalares

PERL é diferente de algumas linguagens de programação: o tipo de variável NÃO precisa ser declarado.

Exemplo em C

```
#include <stdio.h>

int main(void){
    int inteiro;    //guarda numeros inteiros
    char caract;    //guarda caracteres
    float real;     //um número real com precisão simples
    double reald;   //um número real com precisão dupla
    void vazio;     //tipo vazio

    inteiro = 5;
    caract = 'C';
    real = 27.25;
    reald = 22.442e2;

    return 1;
}
```

# 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  
$s = 12
```

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

\$t = atribuir um valor octal = 0377

\$u = atribuir um valor hexadecimal = 0xffff

\$s = atribuir um valor binário = 0b1100



# VARIÁVEIS ESCALARES

## Valores

Script: [escalares.pl](http://escalares.pl); exemplo01 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
$s = 0b1100;    # binario

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

# VARIÁVEIS ESCALARES

## Valores

Script: [escalares.pl](#); exemplo01 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";
print "\$s \= $s\n\n";

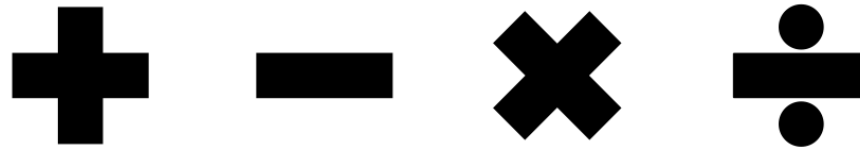
print "$string1\n$string2\n$string3\n\n";

exit;
```

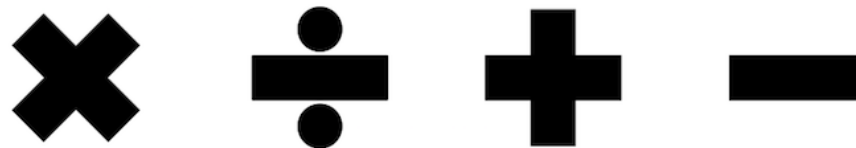
# VARIÁVEIS ESCALARES

## Operações com números

1. Os "operadores da escola" estão disponíveis:



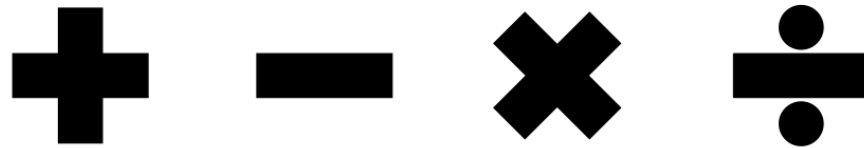
2. Precedência: praticamente igual "da escola"



# VARIÁVEIS ESCALARES

## Operações com números

1. Os "operadores da escola" estão disponíveis:



2. Precedência: praticamente igual "da escola"

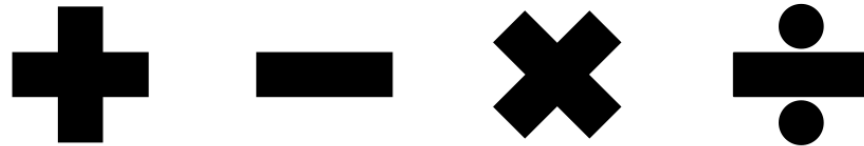


$$2+3*4$$

# 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-decrécimo
/	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 #exemplo02 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
$a = 1;
print "Atribuicao\:           \b$a \b= $a\b\n";

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

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

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

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