

Introdução à Programação de Computadores para Biologia

Estruturas de controle - Loops

Aula 08

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

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

3. *while*

ESTRUTURAS DE CONTROLE - LOOPS

FOR

Script: [loops.pl](#)

```
#!/usr/bin/perl
# script para testar loop com for

# exemplo

print "\nExemplo:\n";

for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

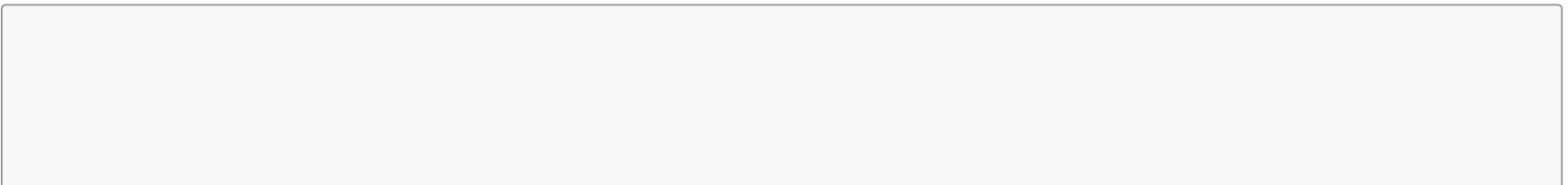
1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

3. *while*



ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [loops.pl](#)

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 01, copiar da página

```
#!/usr/bin/perl
# foreach() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

###

print "\nExemplo 01:\n";

foreach $seq(@seqs) {
    print "$seq\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 02, modificar o exemplo01 para imprimir ID sequencial para cada sequencia("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

###

print "\nExemplo 01:\n";

foreach $seq(@seqs) {
    print "$seq\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 02, modificar o exemplo01 para imprimir ID sequencial para cada sequencia ("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays
print "\nExemplo 02:\n";
$i = 1;
foreach $seq(@seqs) {
    print ">Gene$i\n$seq\n\n";
    ++$i;
}
exit;
```


ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimi-las?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 03, copiar da página

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 03:\n";

#comandos

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 03

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 03:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimir as sequências de forma ORDENADA?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 04, modificar exemplo 03

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 04:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 04, modificar exemplo 04

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 04:\n";

@genes = sort(keys(%seqs));

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

A variável `$_`

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 04:\n";

@genes = sort(keys(%seqs));

foreach (@genes) {
    print ">$_\n$seqs{$_}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como encontrar uma sequência específica? Por exemplo, a sequência *GATAGAG* está presente no hash?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 05, copiar da página

```
# foreach() para encontrar um valor especifico em hashes
print "\nExemplo 05:\n";
$seq_alvo = "GATAGAG"; # esta presente?
foreach $gene (keys(%seqs)) {
    #comandos
}
exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 05

```
# foreach() para encontrar um valor especifico em hashes

print "\nExemplo 05:\n";

$seq_alvo = "GATAGAG"; # esta presente?

foreach $gene (keys(%seqs)) {
    if ($seqs{$gene} eq $seq_alvo) {
        print "Sequencia encontrada: $gene.\n\n";
    }
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

3. *while*

```
while (teste) {  
    bloco de comandos  
}
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR vs WHILE

Script: [loops.pl](#)

Exemplo 06, copiar da página

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR vs WHILE

Script: [loops.pl](#)

Exemplo 06, copiar da página

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

# while
$i = 0;

while ($i < 5) {
    print "$i\n";
    ++$i;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 07, copiar da página

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATAGAG");

# for
$seqs = @seqs;

print "\nExemplo 07 (for):\n";

for ( $i = 0; $i < $seqs; $i++ ) {
    print "$seqs[$i]\n" ;
}
```

```
# formas de imprimir arrays
```

```
@seqs = ("ATGGCGTAGATCG","TAAGCCCCGGTATATTTGACCCT", "GATAGAG");
```

```
# for
```

```
$seqs = @seqs;
```

```
print "\nExemplo 07 (for):\n";
```

```
for ( $i = 0; $i < $seqs; $i++ ) {  
    print "$seqs[$i]\n" ;  
}
```

```
# foreach
```

```
print "\nExemplo 07 (foreach):\n";
```

```
foreach $seq (@seqs) {  
    print "$seq\n" ;  
}
```

```
# while
```

```
#$n_seqs = @seqs;
```

```
#$i = 0;
```

```
#print "\nExemplo 07 (while):\n";
```

```
#while ($i < $n_seqs) {  
#    print "$seqs[$i]\n";  
#}
```

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Usando loop em Perl, crie um array com os números de 10 a 900, contando de 10 em 10.

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Crie um script que vai contar o número de bases G em uma sequência de DNA.

Solução 01

```
#!/usr/bin/perl

#Script para contar a frequencia da base G em uma sequencia de DNA

$DNA = "GATTACACGAT";

$countG = 0;

#criar um array com $DNA
@sequence = split('', $DNA);

#Para cada letra da sequencia, checar se é igual a G
#caso 'verdadeiro' aumentar $countG
foreach(@sequence){
    if($_ eq "G"){
        $countG++;
    }
}

#imprimir o número de Gs
print "Ha $countG bases G\n";
exit;
```

Solução 02

```
#!/usr/bin/perl

#Script para contar a frequencia da base G em uma sequencia de DNA

$DNA = "GATTACACGAT";

$countG = 0;
$currentPos = 0;

#descobrir o tamanho do $DNA
$DNAlength = length($DNA);

#Para cada letra da sequencia, checar se é igual a G
#caso 'verdadeiro' aumentar $countG
while($currentPos < $DNAlength){
    $base = substr($DNA,$currentPos,1);
    if($base eq "G"){
        $countG++;
    }
    $currentPos++;
}

#imprimir o número de Gs
print "Ha $countG bases G\n";
exit;
```

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Crie um script que vai contar o número de bases G ou C em uma sequência de DNA.