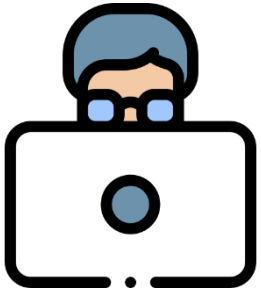

Deloitte.
Project
Coding
100-days-of-code
.git
FAQ.md
log.md
r1-log.md
README.md
resources.md
rules.md
atom-packages
browser_persistence
c01
FlashcardsExpress
freecodecamp_tribute
JavaScript-Authentication
.git
models
public
routes
index.js
views
.gitignore
app.js
package.json
README.md
LocalWeatherFCC
node-weather-zipcode
nodeschool
NodeWeather
portfolio
log.md
index.js
var express = require('express');
var router = express.Router();
var User = require('../models/user');
// GET /register
router.get('/register', function(req, res, next) {
return res.render('register', { title: 'Sign Up' });
});
// POST /register
router.post('/register', function(req, res, next) {
if (req.body.email &&
req.body.name &&
req.body.favoriteBook &&
req.body.password &&
req.body.confirmPassword) {
// confirm that user typed same password twice
if (req.body.password !== req.body.confirmPassword) {
var err = new Error('Passwords do not match.');
err.status = 400;
return next(err);
}
// create object with form input
var userData = {
email: req.body.email,
name: req.body.name,
favoriteBook: req.body.favoriteBook,
password: req.body.password
};
// use schema's `create` method to insert document into Mongo
User.create(userData, function (error, user) {
if (error) {
JavaScript-Authentication-Mongo-Express/routes/index.js 1:1
LF UTF-8 JavaScript 0 files

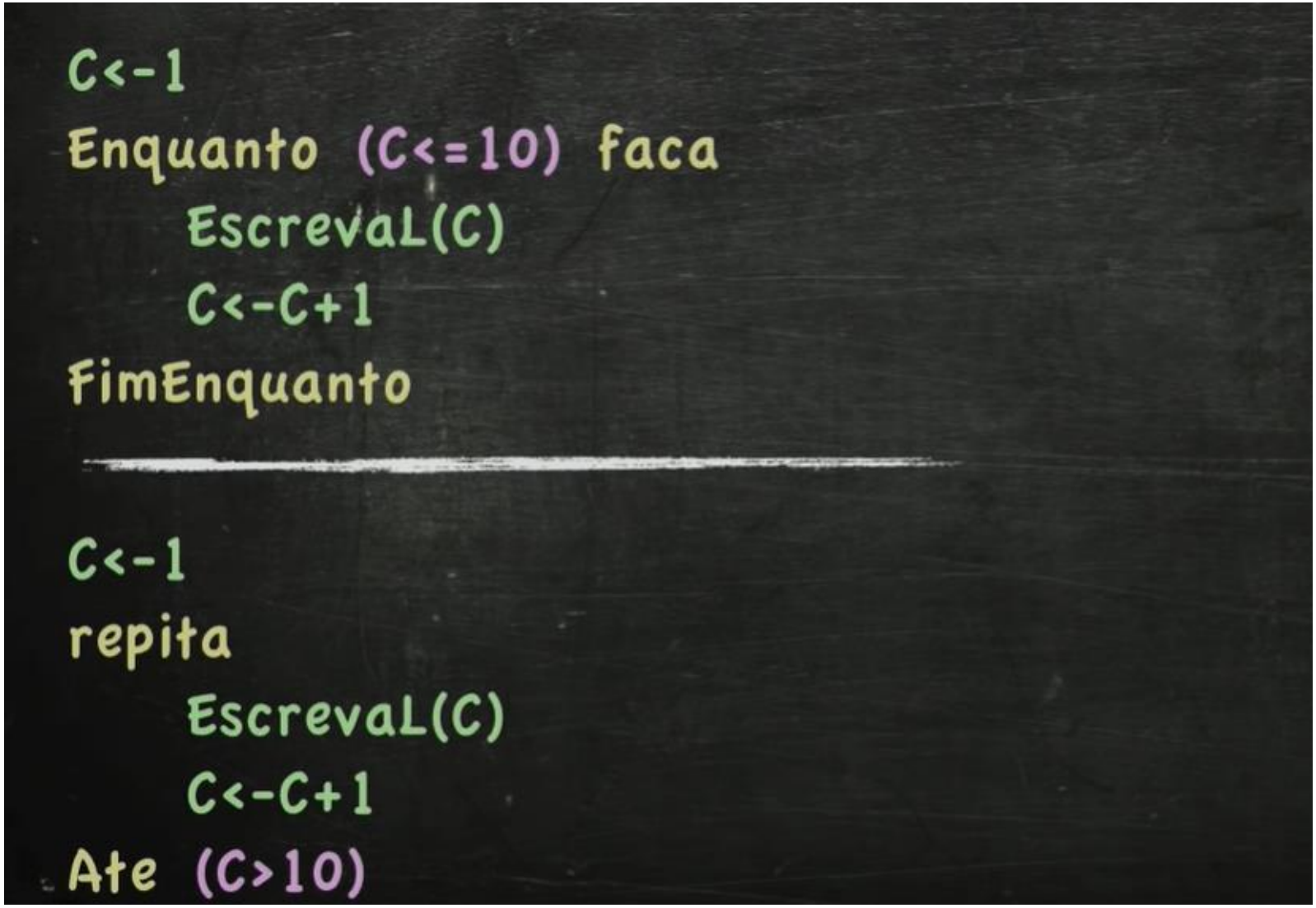
Estruturas de Repetição



Relembrando as estruturas de repetição vistas, temos a **Enquanto faca** e **Repita ate**.

Ambas usam expressõs lógicas para determinar até quando um bloco do código irá ser executado.

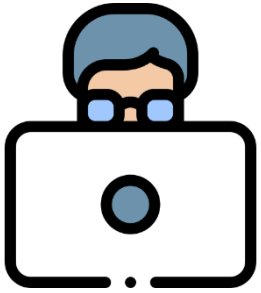
Estruturas de Repetição



```
C<-1
Enquanto (C<=10) faca
    EscrevaL(C)
    C<-C+1
FimEnquanto
```

```
C<-1
repita
    EscrevaL(C)
    C<-C+1
Ate (C>10)
```

Estruturas de Repetição



A última estrutura de repetição que veremos é a estrutura **Para... Faça**

Ela usa uma variável de controle, o que permite que você execute o bloco inteiro uma quantidade determinada de vezes.

Estruturas de Repetição

Para variavel <- inicio ate fim [passo salto] faca

    bloco

FimPara

Parâmetro opcional

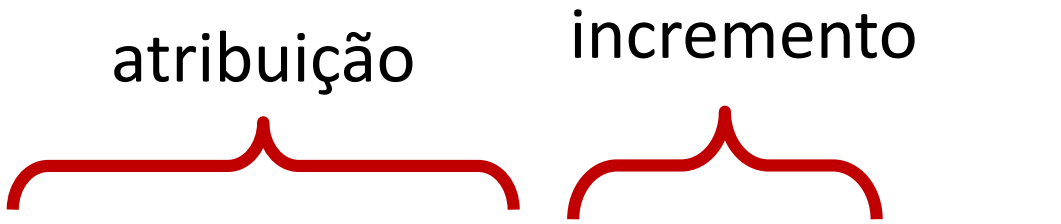
Exemplo

Vamos ver como exemplo o mesmo algoritmo para contar até 10 usando esta estrutura:

```
Para C <- 1 ate 10  passo 1  faca
    EscrevaL ( C )
FimPara
```

Estruturas de Repetição

Ele é mais simples que os anteriores, porque já vem com a atribuição incluida e com incremento incluido.




```
Para C <- 1 ate 10  passo 1  faca
    EscrevaL ( C )
FimPara
```

Observação

O padrão de incremento no passo é 1, ou seja, poderiamos omitir que o resultado neste caso seria o mesmo:

```
Para C <- 1 ate 10  faca
    EscrevaL ( C )
FimPara
```

Estruturas de Repetição

```
1  algoritmo "EXEMPLO CONTADOR ATÉ 10"
2  var
3
4     C:inteiro
5
6  inicio
7
8     Para C<-1 ate 10 faca
9        EscrevaL(C)
10    FimPara
11
12 fimalgoritmo
```

Exercício Prático

Faça um contador que conte de 1 a 10 e pule de 2 em 2.

Exercício Prático

Faça um contador de solicite 10 numeros inteiros e some, mostrando o resultado desta soma

(usando esta nova estrutura de repetição)

Observação

Na maioria dos casos podemos usar esta estrutura como substituição ou alternativa para as já vistas.

Mas há um caso em que ela **não é eficiente**: quando pedimos se **o usuário deseja continuar [S/N],** como foi visto nos exercícios das aulas anteriores.

Nestes casos somente o **Repita Ate** é eficiente.

Exercício Prático

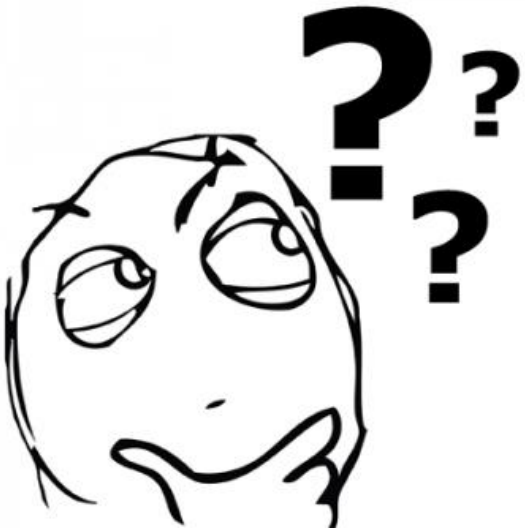
Mostrar números pares

Solicite que o usuário digite um valor e mostre somente os números pares de 0 a este valor digitado.
(usando a estrutura Para Faça)

Exercício Prático

Quantos valores estão entre 0 e 10?

Solicite que o usuário digite 6 valores e faça o programa mostrar quantos estão entre 0 e 10



Desafio

Quantos valores estão entre 0 e 10?

Solicite que o usuário digite 6 valores e faça o programa mostrar quantos estão entre 0 e 10.

Adicional: exibir a soma entre os números ímpares digitados.