On19 - Todas em Tech - Backend - Mayhhara Morais

OIII, GATINHASSSS

Eu sou a May!
Sou líder técnica da turma On19, engenheira de software backend e, atualmente, trabalho (principalmente) com GoLang na Wildlife Studios.

SAC da May

Mayhhara Morais: linkedinho
Mayhhara Morais: slack
@mayhhara_: instagram
@mayjinboo: twitter

AGENDA DE HOJE

DICA

Hoje temos muito conteúdo e o tempo bem curto.
Por isso, afim de mantermos a fluidez da aula, deixarei disponível um link para o Slido para que vocês adicionem as dúvidas que surgirem.

Sábado, 10/Set

- Git & Github 09h10 - 10h
- Lógica 10h30 - 12h
- Lógica Aplicada 1 13h - 14h30
- Lógica Aplicada 2 15h - 17h

FRASE DO DIA

Pode ir tirando essa expressão de desanimo da cara, antes que eu arranque ela na bicuda pra te lembrar que você é uma máquina de vencer!

Clarice Lispector

SEMANA 1

Introdução
Versionamento de Código, Git e Github

BORA DE CONCEITO?

- **Versionamento de Código**

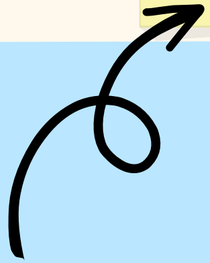
Podemos compreender a expressão “versionamento de código” como um sistema de controle de versão de códigos e scripts específicos.

Um exemplo é a facilidade de gerenciar projetos, dos mais simples aos mais complexos, no **GitHub**.

- **Github**

É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Nele, é possível administrar as mudanças que foram feitas no código, saber quem fez as alterações, bem como gerenciar as ramificações que cada mudança gera.

O termo **OPEN SOURCE** é usado quando tratamos de um código projetado para ser acessado abertamente pelo público: todas as pessoas podem vê-lo, modificá-lo e distribuí-lo conforme suas necessidades.



- **Git**

É o sistema de controle de versão **open source** mais usado no mundo atualmente! Ele é usado para controlar o histórico de alterações de arquivos e principalmente de projetos de desenvolvimento de software. Ele permite mais flexibilidade no fluxo de trabalho, segurança e desempenho.

PRINCIPAIS COMANDOS DO GIT

1. **Git init:** criando seu repositório local!
2. **Git config user.name "Nome da Reprogramer":** configura seu nome
3. **Git config user.email "email@da.reprogramer":** configura seu email do git
4. **Git clone:** baixando o código fonte de um repositório!
5. **Git add:** adicionando alterações no repositório!
6. **Git commit:** confirmando e salvando as alterações no repositórios!
7. **Git push:** enviando as alterações do repositório local para o repositório remoto!
8. **Git pull:** atualizando o repositório local com a versão do repositório remoto!
9. **Git Status:** verificando se há alterações na branch
10. **Git remote:** ligando seu repositório local a um repositório remoto!
11. **Git checkout -b "nome branch":** criando uma nova branch e restaurando arquivos!
12. **Git branch:** listando e deletando branches
13. **Git log:** verifique quais commits foram feitos até agora
14. **Git reset:** desfazendo as alterações de um repositório para um commit anterior!

SEMANA 2

Lógica
Variáveis, Operadores e Condicional

BORA DE CONCEITO?

Variáveis

São caixinhas onde armazenamos informações.
Essas informações podem ser números, frases, valores de verdadeiro ou falso, entre outros variados tipos de dados.

Operadores

São os responsáveis por fazer operações matemáticas entre os valores das variáveis.
Esses operadores podem ser aritméticos, de atribuição e lógicos.

Condicionais

As declarações condicionais nos permitem representar tomadas de decisão, a partir da escolha que deve ser feita, ao resultado obtido dessas escolhas.

VARIÁVEIS

```
let nome = "May"                    string
let idade = 22                      number
let ehFilhaDaBeyonce = true         boolean
let filhos = null                   valor vazio
```

OPERADORES

Aritméticos:
soma(+), subtrai(-), multiplica(*), divide(/), calcula o resto(%), incrementa(++), decrementa(--)

Atribuição:
atribui(=), atribui somando(+=), atribui subtraindo(-=), atribui multiplicando(*=)

Lógicos:
maior que(>), menor que(<), maior ou igual a(>=), menor ou igual a(<=), valor idêntico a(==), valor tipoidêntico a(===), diferente de(!=), e(&&), ou(||)

CONDICIONAIS

if else

```
if (condicao) {
  codigo para executar caso a condição seja verdadeira
} else {
  senão, executar este código
}
```

ternário

```
( condicao ) ? execute isso : se nao, execute isso
```

switch case

```
switch (expressao) {
  case condicao1:
    execute isso
    break;

  case condicao2:
    execute isso no lugar
    break;

  // posso incluir a quantidade que eu quiser

  default:
    execute isso caso nao entre nas condicoes
}
```

SEMANA 3

Lógica Aplicada
Loop, Escopo e Função

BORA DE CONCEITO?

- **Loop**

É como fazer a mesma coisa repetidas vezes - o que é chamado de iteração na linguagem de programação

- **Escopo**

É a acessibilidade de objetos, variáveis e funções em diferentes partes do código. Em outras palavras, o que determina quais são os dados que podem ser acessados em uma determinada parte do código é o escopo.

- **Função**

Um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.

LOOP

for de C

```
for (setup; condição; passo) {
  Enquanto a condição acima for
  verdadeira, o código entre as
  chaves se repetirá em loop
}
```

for of

```
for (let item of array) {
  Enquanto a condição acima for
  verdadeira, o código entre as
  chaves se repetirá em loop
}
```

while

```
while (condição) {
  Enquanto a condição acima for
  verdadeira, o código entre as
  chaves se repetirá em loop
}
```

ESCOPO

```
function primeira() {
  // Escopo da funçao
     primeira
   function segunda() {
    // Escopo da funçao
       segunda
  }
}

function terceira() {
  //Escopo da funçao
     terceira
}
```

O escopo é como se fosse o "corpinho" da funçao, o que ta dentro desse corpo só existe dentro dele.

FUNÇÃO

```
// Ela pode ser em sua forma
completa

const digaOi =
    function(name) {
      return `Oiiii ${name}!!!`; }


// Pode também ser resumida
(arrow function)

const digaOi =
(name) => `Oiiii ${name}!!!`;
```

SEMANA 4

Lógica Aplicada
Objetos, Arrays e Métodos

BORA DE CONCEITO?

Objetos

É um conjunto de dados ou funcionalidades relacionadas.

Arrays

É um conjunto de dados ordenados guardados numa mesma variável.

Métodos

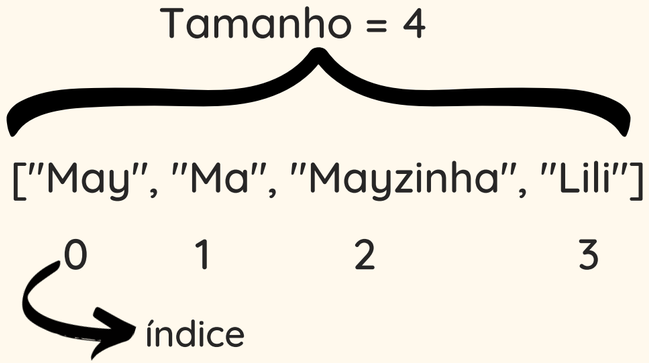
Um método JavaScript é uma propriedade de um objeto que contém uma definição de função.
Ele consiste em um código que pode ser chamado pelo nome de seu método usando a notação de ponto ou a notação de colchetes.

OBJETOS

- O objeto é composto por **membros** (ou **propriedades**)
- Cada **propriedade** possui um **nome**(ou **chave**) e valor para essa **chave**.

```
const pessoa {
   nome = "Mayhhara",
   idade = 22,
   temCNH = true
   apelidos = ["May", "Ma", "Mayzinha", "Lili"]
}
```

ARRAYS



- **Arrays** permitem o armazenamento ordenado de dados semelhantes
- **index** é a posição de cada item na lista, começando sempre em zero
- O último item da lista tem sempre **index = length - 1**

ALGUNS MÉTODOS

pop() - remove um item do fim do array
push() - adiciona um item ao fim do array
shift() - adiciona um item ao início do array
unshift() - remove um item do início do array
sort() - ordena os elementos do próprio array
splice() - altera o conteúdo do array, adicionando novos elementos enquanto remove elementos antigos.
reduce() - cria um acumulador ou reduz o array a um valor único utilizando uma função que cria um valor único.
concat() - retorna um novo array contendo todos os arrays ou valores passados como parâmetro
reverse() - inverte os itens de um array.
filter() - cria um novo array como os elementos que passaram na condição da função que foi data como parâmetro para o método.
find() - retorna o valor do primeiro elemento do array que satisfizer a função provida.
map() - retorna um novo array fazendo modificações em cada um dos itens
Math.random() - retorna um número aleatório entre 0 e 1

LINKS ÚTEIS

Funções: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Functions#function_scope

Loop: https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Building_blocks/Looping_code

Array e Métodos de Array:
https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array

Métodos de String:
https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Useful_string_methods

Método Map:
https://www.freecodecamp.org/portuguese/news/map-em-javascript-como-usar-a-funcao-map-do-js-metodo-de-arrays/

Arrow Function:
https://hcode.com.br/blog/entendendo-arrow-functions-de-uma-vez-por-todas

OBRIGADA!

Muito conteúdo? Sim!
Se ajudem, peçam ajuda e
continuem tentando!
Vai dar certo!!!