

Programação de IAs Conversacionais

Módulo 1 - Variáveis



Índice

1. Console
2. Variáveis
3. `console.log()`
4. Concatenação
5. Operadores aritméticos
6. Exercícios





Programação de IAs Conversacionais

Console

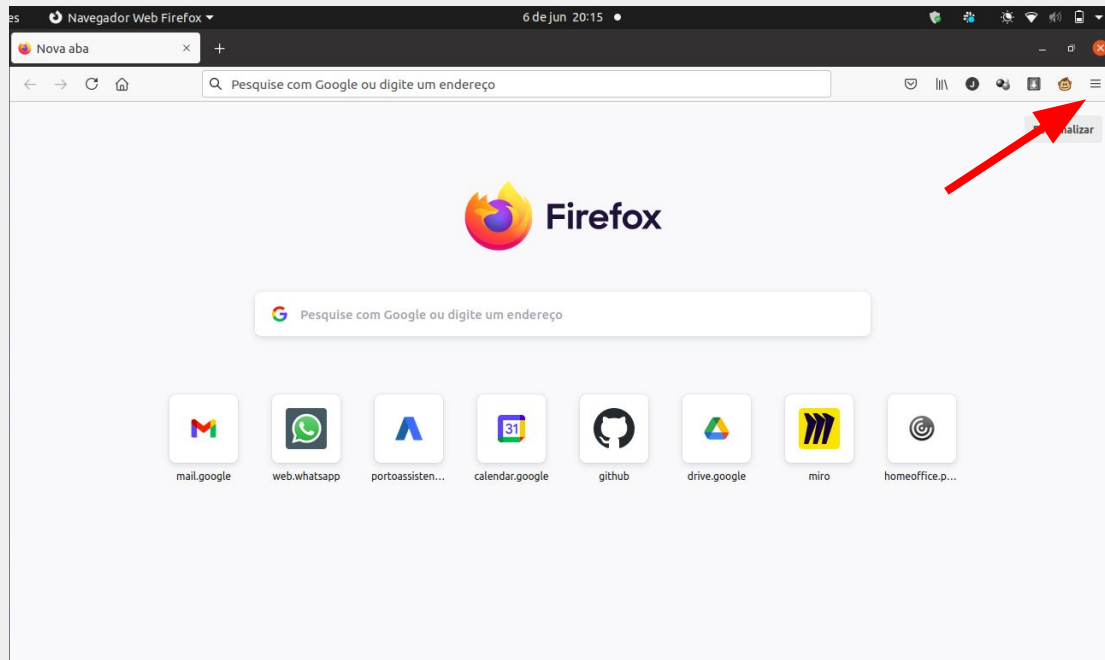


Console

Iremos usar o console do navegador Mozilla Firefox



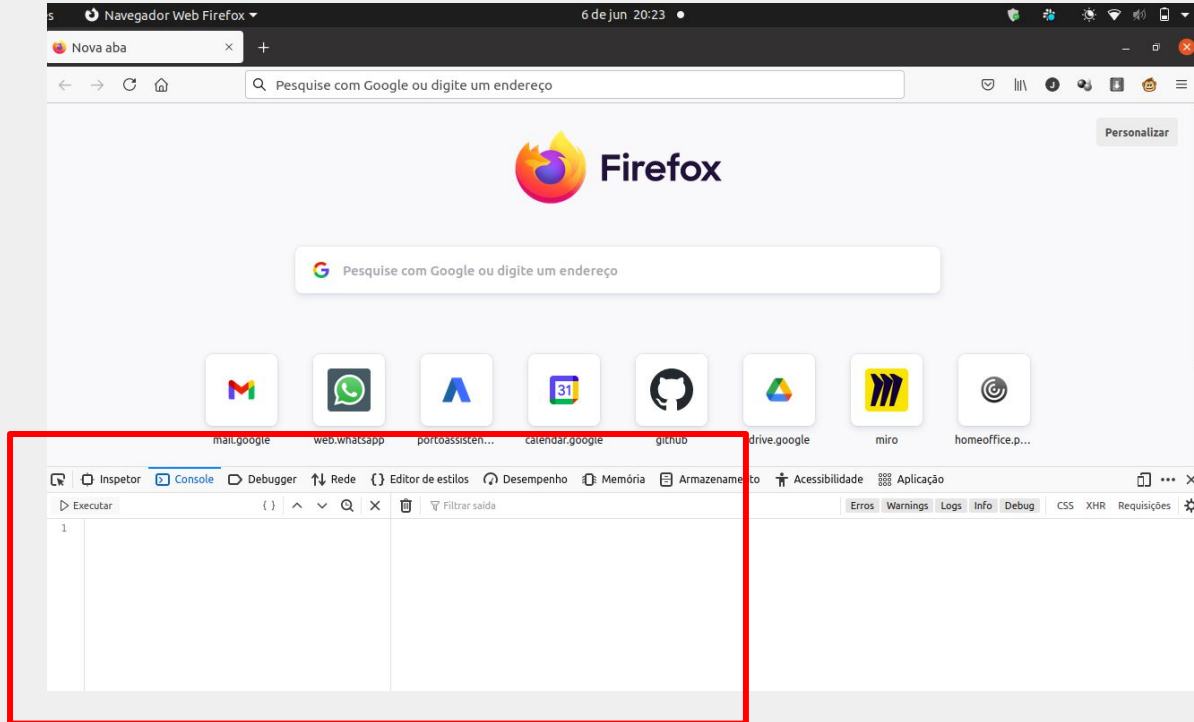
Console



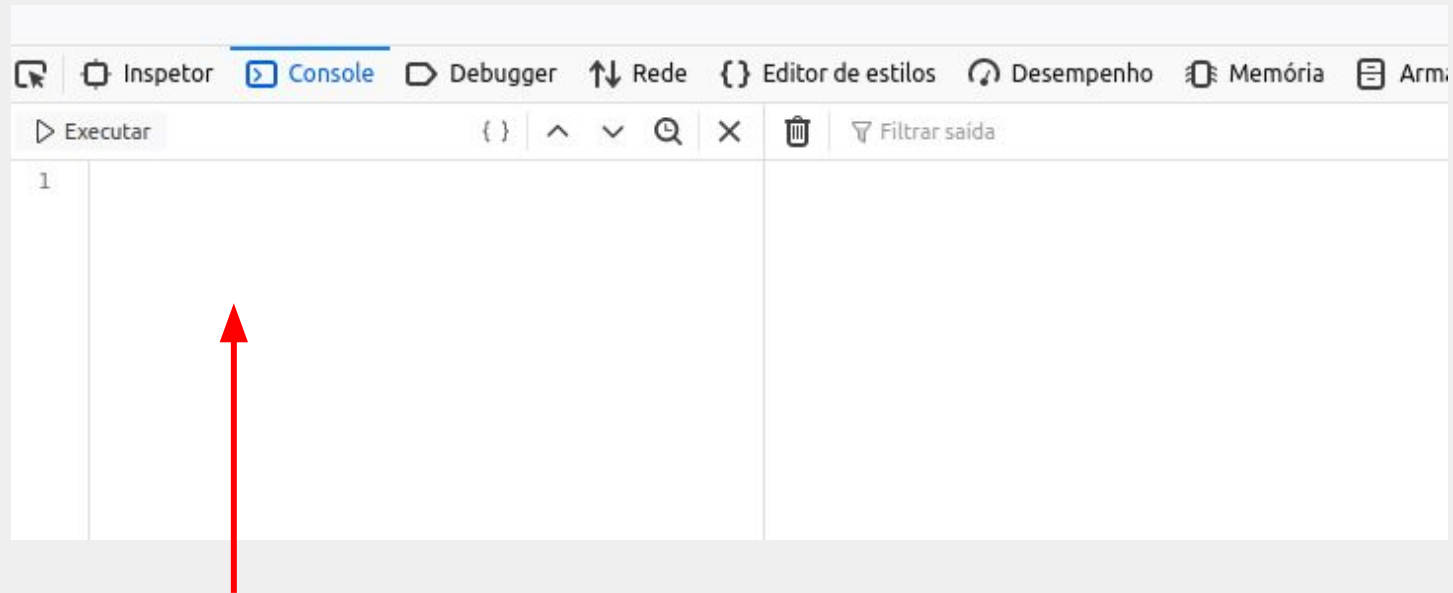
Console

- Clique em mais ferramentas
- Clique em ferramentas de desenvolvedor
- Ou use o atalho:
 ctrl + shift + i
- Ou o botão f12

Console



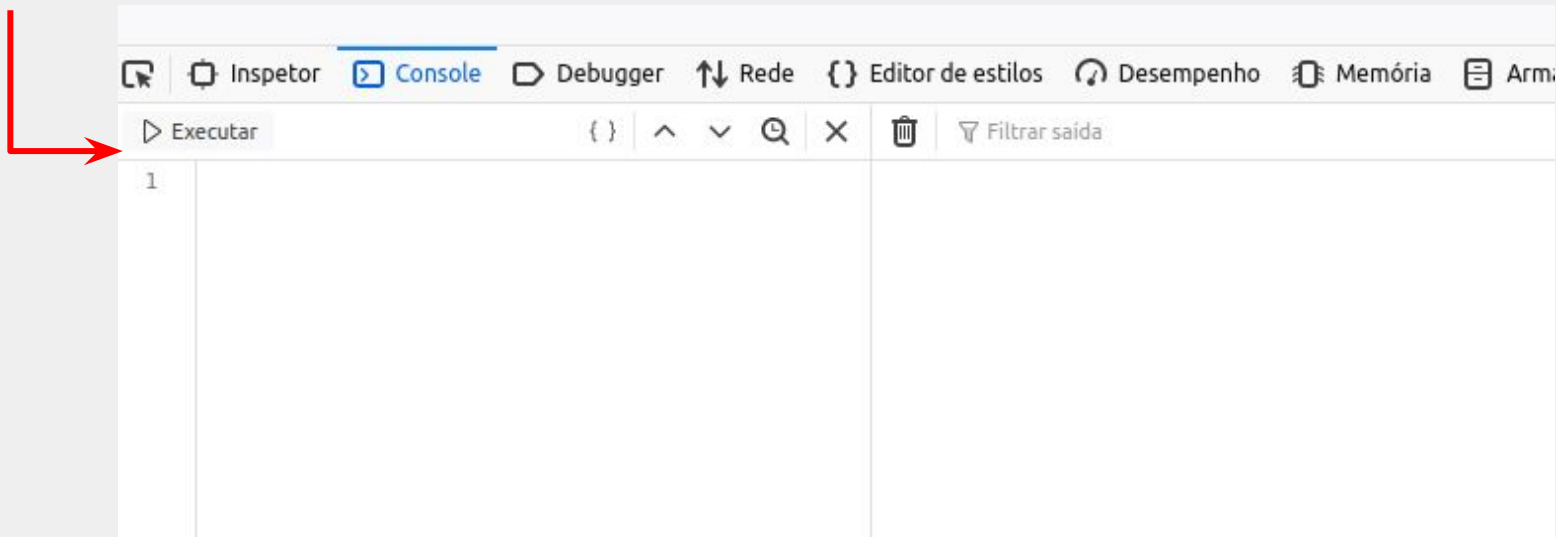
Console



Digitamos o código aqui

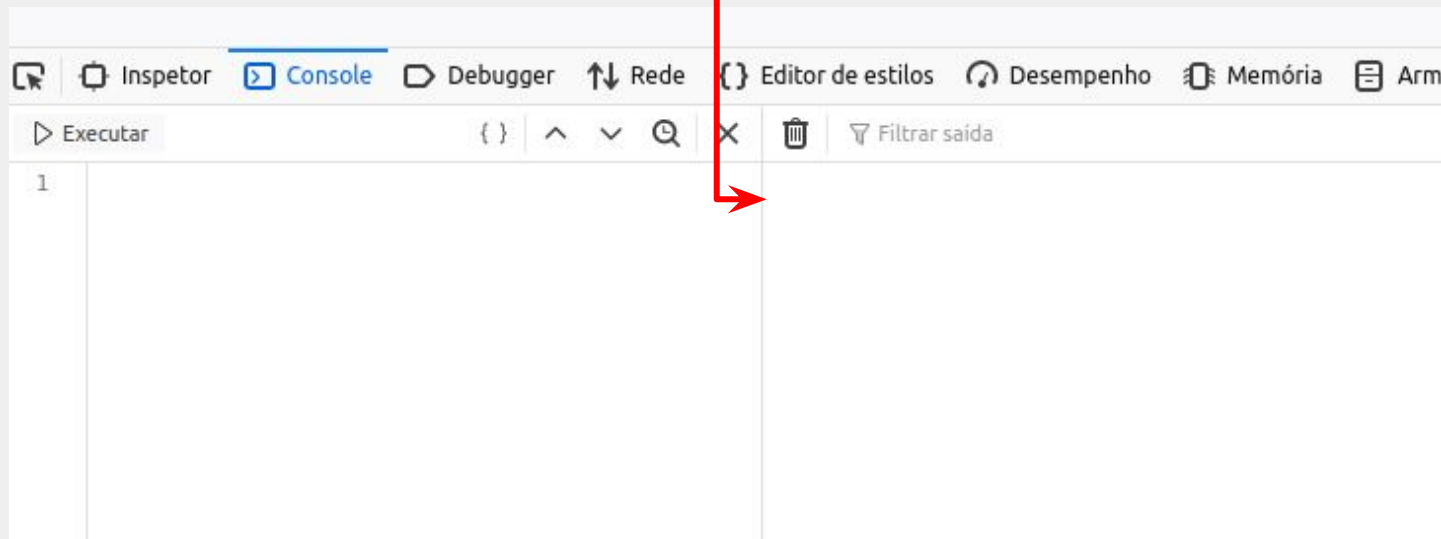
Console

Aqui para executar o código



Console

Aqui sai o resultado do código





Programação de IAs Conversacionais

Variáveis



Introdução

- Onde você guarda seus objetos?
- Guardamos nossas coisas para nos organizarmos
- Na programação também guardamos “coisas”
- Armazenamos o que precisamos nas **Variáveis**

Introdução

Variáveis são como caixas, onde guardamos o que precisamos



Introdução

- Nessas “caixas” colocamos um rótulo
- Caixa de sapatos, caixa de brinquedos
- Facilidade para encontrarmos depois

Introdução

- Tecnicamente variáveis são espaços na memória do computador onde armazenamos diferentes tipos de dados

Regras básicas de uma variável

- Ter um nome
- Ter um tipo
- Ter um valor

Regras para nomes

- Não podem ser iguais a palavras reservadas;
- Devem possuir como primeiro caractere uma letra ou sublinhado '_' (os caracteres seguintes podem ser letras, números e sublinhado) - SnakeCase
- Primeira letra em minúsculo

Regras para nomes

- Podemos usar letras, números e o sublinhado apenas
- O primeiro caractere deve ser uma letra ou um sublinhado
- Não podem conter espaços em branco;
- Não podem conter símbolos (exceto o _)
- CamelCase - denominação em inglês para palavras compostas ou frases, primeira palavra em minúscula e as seguintes em maiúscula. Ex: casaAzul

Palavras reservadas

As palavras reservadas são:

let, var, const, default, function, return, case, if,
switch, void, catch, do, in, this, while, else,
instanceof, throw, for, try, typeof, continue,
debugger, finally, break, case, class, delete, with,
yield, export

Exemplo no pseudocódigo

INÍCIO

VARIÁVEIS

NUM_1, NUM_2, SOMA: INTEIROS;

NUM_1 = 22;

NUM_2 = 11;

SOMA = NUM_1 + NUM_2;

ESCREVA ("A soma de ", NUM_1, " + ", NUM_2, " é: ", SOMA)

FIM

Exemplo no pseudocódigo

INÍCIO

VARIÁVEIS

NUM_1, NUM_2, SOMA: INTEIROS;

NUM_1 = 22;

NUM_2 = 11;

SOMA = NUM_1 + NUM_2;

ESCREVA ("A soma de ", NUM_1, " + ", NUM_2, " é: ", SOMA)

FIM

Exemplo no pseudocódigo

INÍCIO

VARIÁVEIS

NUM_1, NUM_2, SOMA: INTEIROS;

NUM_1 = 22;

NUM_2 = 11;

SOMA = NUM_1 + NUM_2;

ESCREVA ("A soma de ", NUM_1, " + ", NUM_2, " é: ", SOMA)

FIM

Exemplo no pseudocódigo

INÍCIO

VARIÁVEIS

NUM_1, NUM_2, SOMA: INTEIROS;

NUM_1 = 22;

NUM_2 = 11;

SOMA = NUM_1 + NUM_2;

ESCREVA ("A soma de ", NUM_1, " + ", NUM_2, " é: ", SOMA)

FIM

Criando variáveis

Podemos usar **var**, **let** ou **const**

Exemplo:

```
var numeroUm = 1;
```

```
let numeroUm = 1;
```

```
const numeroUm = 1;
```

Obs: Usaremos **var** nesse módulo

Criando variáveis

Diferença entre **var** e **let**

- Uma variável **var** é iniciada no começo do código
- Hoisting (elevada/içada)
- Uma variável **let** só inicia dentro da função declarada

Criando variáveis

const

- Variáveis const são **constantes**, ou seja, não é possível mudar seu conteúdo durante a execução do código

Usando variáveis

- Declarar a variável com **var**
- Definir o nome
- Definir o **tipo** ou atribuir um **valor** diretamente usando o sinal =

```
var minhaVariavel = Number
```

ou

```
var meuNome = "Maria"
```

Usando variáveis

Podemos mudar o valor da variável:

```
var minhaVariavel = 12
```

```
minhaVariavel = 25
```

```
minhaVariavel = "Céu"
```

Usando variáveis

```
console.log()
```

Para imprimir no console o valor da variável usamos o comando `console.log()`

```
console.log(minhaVariavel)  
//irá imprimir o último conteúdo que guardamos:  
Céu
```



Programação de IAs Conversacionais

console.log()



`console.log()`

Console é um objeto do javascript (falaremos sobre objetos mais pra frente)

Em resumo o console é uma espécie de função nativa que possui vários métodos

`.log()` é um dos métodos do objeto console

Outros métodos do console: `.error()`, `.warn()` entre outros...

`console.log()`

`.log` serve para emitir mensagens de depuração do sistema

Em outras palavras, `console.log()` imprime na tela do nosso ambiente de desenvolvimento (IDE) o que passamos dentro dos parênteses

`console.log()`

Cuidado!

Em um projeto real não podemos deixar esse comando no código, usamos apenas quando estamos desenvolvendo

Nas nossas aulas usaremos para exibir texto ou os valores das variáveis



Programação de IAs Conversacionais

Concatenação



Concatenação

Significa encadear, juntar

É a junção de dois ou mais valores

Usado para unir texto/palavras/caracteres

Usa-se o símbolo +

Concatenação no console.log()

```
var nome = "Maria"
```

```
var sobrenome = "Silva"
```

```
console.log(nome + sobrenome)
```

Concatenação no console.log()

Podemos inserir outras palavras ou símbolos no console.log()

```
console.log("Nome: " + nome + " " + sobrenome)
```

```
// retorna Nome: Maria Silva
```

Concatenação na Variável

Concatenação na variável:

Podemos concatenar variáveis e guardar em outra variável

```
var nomeCompleto = nome + " " + sobrenome
```

```
console.log(nomeCompleto) // imprime Maria Silva
```



Programação de IAs Conversacionais

Operadores Aritméticos



Soma +

- Operador de adição
- Une dois valores transformando em um novo valor

$$2 + 2 = 4$$

- Sempre tenta converter o valor em um número quando possível

$$+ \text{"2"} = 2$$

Soma +

Mas e o símbolo da concatenação?

A concatenação usamos apenas em strings

Faça o teste: "2" + 2

Subtração -

- Operador de diminuição

$$5 - 3 = 2$$

- Reduz o valor
- Transforma um número em negativo
-“2” = -2

Multiplicação *

- Operação para multiplicar os valores numéricos

```
var resultado = 2 * 2
```

```
console.log(resultado) //imprime 4
```

Divisão /

- Operação para dividir dois valores numéricos

```
var resultado = 10 / 2
```

```
console.log(resultado) //imprime 5
```

Módulo %

- Devolve o resto da divisão

`var resultado = 8 % 3 //devolve 2`

`var resultado = 10 % 2 //devolve 0`

`var resultado = 3 % 2 //devolve 1`

Incremento ++

- Usado para incrementar 1 ao valor

```
var valor = 2
```

```
valor ++ //retorna 3
```

Decremento --

- Usado para decrementar 1 ao valor (diminuir)

```
var valor = 2
```

```
valor -- //retorna 1
```



Programação de IAs Conversacionais

Exercícios



Exercícios

Declare duas variáveis e atribua nome, tipo e valor
concatene as duas variáveis em uma variável nova
imprima com `console.log()`

Exercícios

Você tem as variáveis:

```
var primeiroNome = 'João'  
var nomeDoMeio = 'Silva'  
var ultimoNome = 'Andrade'
```

crie a variável nomeCompleto e guarde o nome inteiro do João

imprima com console.log()

Exercícios

Você tem as variáveis:

`num1 = 3`

`num2 = 6`

Faça a subtração destes números e imprima com `console.log()`

Exercícios

Com as variáveis:

`num1 = 23`

`num2 = 12`

Faça a soma de `num1 + num2` e depois divida por 2,
imprima com `console.log()`



- ae-search-in-files
- ae-shell
- ae-subprocess-inspector
- ae-term
- test-shutdown-after-startup
- lib
- tcl
- test
- README
- aradia.gemspec
- aradia.todo

```
def square_id(id)
```

