<LAB365> SENAI

SEJAM BEM-VINDOS!

Jornada DEV AM. CE. RR

Formação gratuita
em Desenvolvimento
Front-End SENAI

Começamos em breve!

A aula vai iniciar em **5 minutos**.

Relaxe e aproveite a música enquanto aguardamos todos chegarem!

Lógica com JavaScript #03

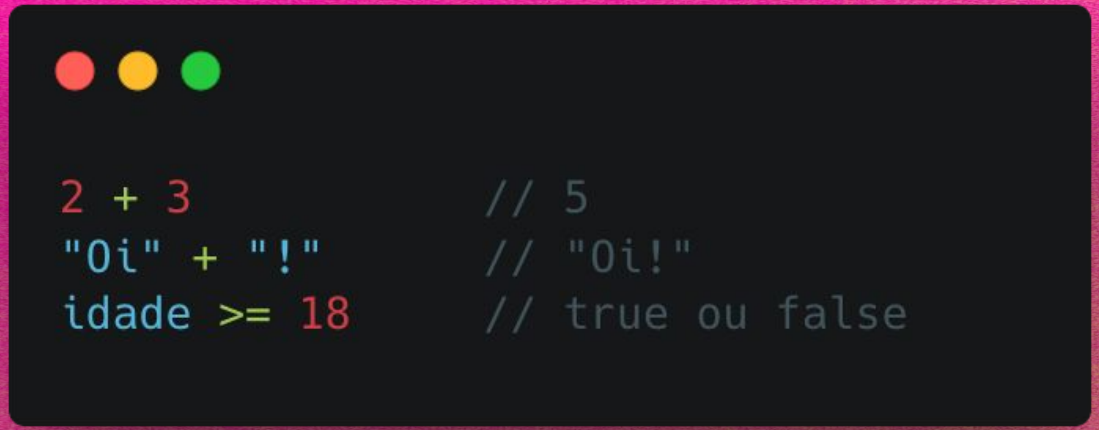
SENAI

Missão de hoje

- Expressões e tipo indefinido
- Retorno implícito
- Chamada de função condicional
- Operações com Strings
- Lógica booleana
- If/Else

Expressões

- São blocos de código que **resultam em um valor**
- Podem ser números, textos, comparações, funções

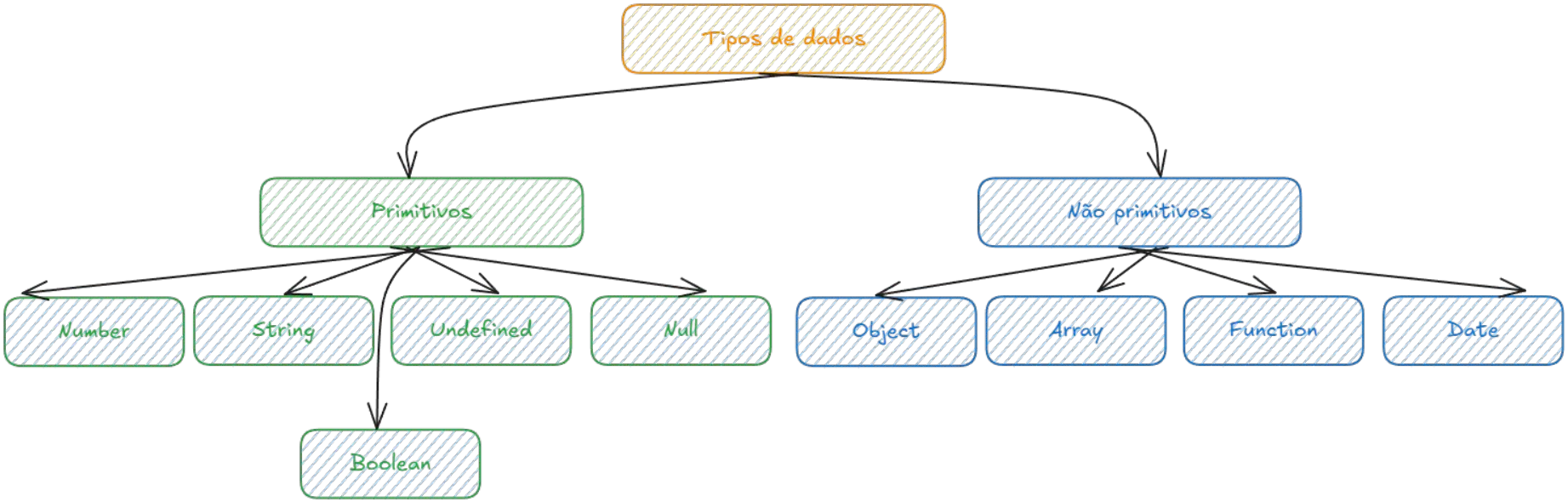


<LAB365>

Tipos de dados

SENAI

TIPOS DE DADOS



Primitivo - Glossário do MDN Web Docs: Definições de termos relacionados à Web | MDN

Estrutura de dados do Javascript - JavaScript | MDN

TIPOS DE DADOS

- Vamos entender alguns tipos de dados primitivos:
  - String;
  - Boolean;
  - Number;
  - **Undefined;**
  - Null;

Undefined: Indefinido

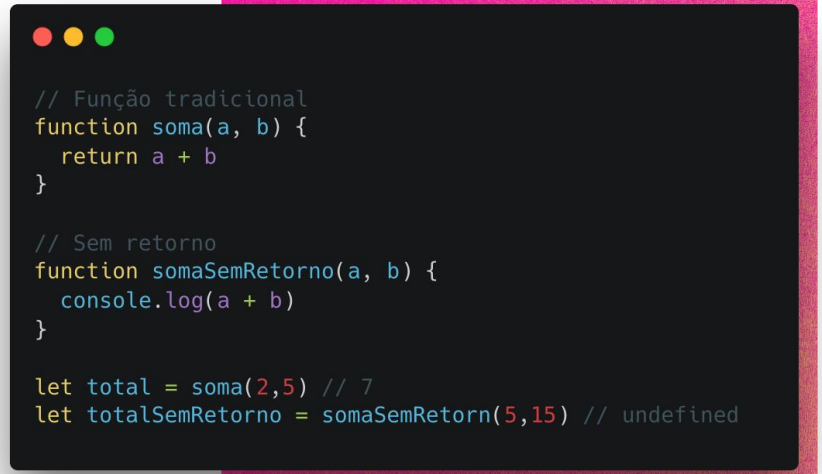
- O tipo undefined significa “não definido” e é atribuído de forma padrão pelo JavaScript a variáveis que não foram inicializadas com nenhum valor.

Retorno implícito

Em JavaScript, toda função gera um resultado.

Mas... e se não colocarmos return?

- Nesse caso, o retorno padrão será undefined.
- Esse comportamento é chamado de **retorno implícito**.



```
// Função tradicional
function soma(a, b) {
  return a + b
}

// Sem retorno
function somaSemRetorno(a, b) {
  console.log(a + b)
}

let total = soma(2,5) // 7
let totalSemRetorno = somaSemRetorn(5,15) // undefined
```

Chamada de Função Condicional

Muitas vezes só queremos **executar uma função se uma condição for verdadeira**.

Duas formas principais:

- Usando **if** → mais legível em condições complexas.

Isso evita chamadas desnecessárias e deixa o código mais limpo.

```
let logado = true

// Forma tradicional
if (logado === true) {
  console.log("Bem-vindo!")
}
```

Operações com Strings

Podemos **juntar textos** em JavaScript de duas formas principais:

Concatenação:

```
let nome = "Maria"
console.log("Olá " + nome)
// Olá Maria
```

Interpolação (template string):

```
let nome = "Maria"
console.log(`Olá ${nome}`)
// Olá Maria
```

Vantagem do Template String

- Torna o código **mais legível** e fácil de entender.
- Evita usar vários + para juntar textos.
- Permite incluir **variáveis e expressões direto dentro da string**.

Concatenação tradicional

```
let nome = "Luca"
let idade = 13
let lugar = "Portorosso"

console.log("Meu nome é " + nome + ", tenho " + idade + " anos e
moro em " + lugar + ".")
// Meu nome é Luca, tenho 13 anos e moro em Portorosso.
```

Template String

```
let nome = "Luca"
let idade = 13
let lugar = "Portorosso"

console.log(`Meu nome é ${nome}, tenho ${idade} anos e moro em ${lugar}.`)
// Meu nome é Luca, tenho 13 anos e moro em Portorosso.
```

Então, por que usar Template String?

- Código mais **curto** e **legível**.
- Fácil de **inserir variáveis** dentro do texto.
- Dá para usar **expressões direto**:

```
console.log(`Ano que vem terei ${idade + 1} anos!`)
```

Caso de uso: Convite de Casamento.

- Vamos abrir uma empresa de convites de casamento, e temos nosso texto padrão. Porém, escrever cada um dos convites manualmente é impraticável;
- Dado o texto padrão, identifique quais variáveis podemos utilizar no lugar do texto, para que possamos alterar o nome dos convidados e dos noivos sem precisar escrever todo o texto novamente:

Caro(a) Fulano(a)!

Você está convidado(a) para o casamento de Beltrano(a) e Ciclano(a), a ser realizado no dia 05/12/2022, às 16 horas.

Contamos com a sua presença!

Atenciosamente,

os(as) noivos(as)

Caro(a) Fulano(a)!

Você está convidado(a) para o casamento de Beltrano(a) e Ciclano(a), a ser realizado no dia 05/12/2022, às 16 horas.

Contamos com a sua presença!

Atenciosamente,

os(as) noivos(as)

Vamos praticar

- **Expressões no console:**
  - Comparação – Idade mínima para dirigir
  - Saldo em conta
  - Entrada de texto (.trim())
- Chamada de função condicional
- ex.1 Chamada de função condicional
- Lista de compras
  - apresentando múltiplos valores em um único texto.
- Mensagem de aniversário

Dúvidas ?

Intervalo!

Finalizamos o nosso primeiro período de hoje.

Nos vemos em 20min.

Início: 20:00
Retorno: 20:15

<LAB365>

Condições encadeadas

SENAI

Revisão

Estruturas Condicionais

As estruturas condicionais permitem que seu código execute diferentes blocos de instruções com base em condições específicas. **É como tomar decisões no dia a dia**: se chove, pegamos o guarda-chuva; se não, deixamos em casa.

Por que usar ?

- Controlar o fluxo de execução do programa.
- Executar código diferente com base em condições.
- Criar lógica de negócios complexa.

Comandos de decisão

Vamos abordar os comandos mais comuns, que são:

1. SE (if);
2. **SE/SE NÃO (if/else);**
3. **SE aninhado (else if);**

```
// Exemplo 1: if
let idade = 20
if (idade >= 18) {
  console.log("Maior de idade")
}

// Exemplo 2: if/else
let temperatura = 15
if (temperatura > 20) {
  console.log("Dia quente")
} else {
  console.log("Dia frio")
}

// Exemplo 3: if/else if/else
let nota = 8
if (nota >= 9) {
  console.log("Excelente")
} else if (nota >= 6) {
  console.log("Bom")
} else {
  console.log("Precisa melhorar")
}
```

If/Else

Estrutura básica de decisão

Se a idade for maior ou igual a 18, o programa executa este bloco.

```
if (idade >= 18) {
  console.log("Maior de idade")
} else {
  console.log("Menor de idade")
}
```

Caso contrário (idade menor que 18), este bloco será executado.

Else if

Permite múltiplas condições

Se a nota for maior ou igual a 9, este bloco será executado.

```
if (nota >= 7) {
  console.log("Aprovado")
} else if (nota >= 5) {
  console.log("Recuperação")
} else {
  console.log("Reprovado")
}
```

Se a condição anterior não for verdadeira, mas a nota for maior ou igual a 6, executa este bloco.

Se nenhuma das condições anteriores for satisfeita, executa este bloco.

If/else sem else: Early return

Se a idade for menor que 18, a função retorna imediatamente este valor e não executa mais nada.

```
function verificaIdade(idade) {
  if (idade < 18) {
  return "Menor de idade"
  }

  return "Maior de idade"
}
```

Quando usamos return, a função é encerrada e o código após essa linha não será executado.

Se a condição anterior não for verdadeira, a função continua e retorna este valor.

Vamos praticar

- Positivo ou negativo ?
- Avaliação do aluno: Recurso de ramificação (notas)
- Função com retorno condicional (Impar ou par?)
  - Ramificado
  - Early return
- **Completando o retorno da função condicional**
- **Mensagem votação**
- **Escrevendo seu próprio else**
- **Escrevendo seu próprio else 2**

Dúvidas ?

Boa noite!


<LAB365>
SENAI