

# Curso: Front-End do Bem



## AULA 20

- Concatenação de Strings  
(+1 exemplo)
  - Template Strings
- Exemplo de Template Strings
  - Template HTML
  - Exemplo de Template HTML
- Sugestões de Prática para essa aula
- O que vamos aprender na aula 21



# Concatenação de Strings (+1 exemplo)

Antes de partirmos para o aprendizado sobre Template Strings, veremos +1 exemplo sobre Concatenação de Strings. E faremos isso, por 2 motivos:

- 1º Para relembrarmos o conceito e praticarmos mais um pouquinho;
- 2º Para compararmos quando usarmos a Template Strings.

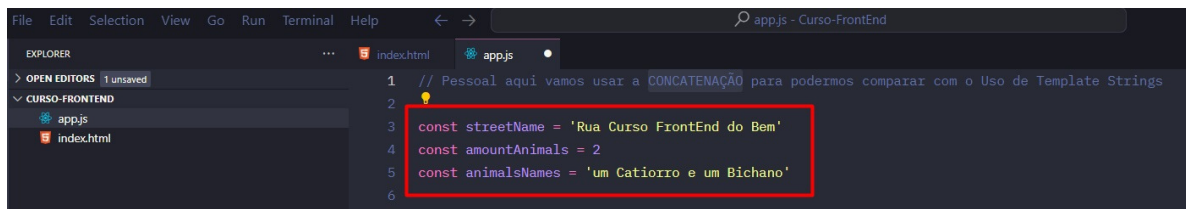
Bora abrir o nosso arquivo app.js e na primeira linha dentro desse arquivo criaremos um comentário: `// Pessoal aqui vamos usar a CONCATENAÇÃO` para podermos comparar com o Uso de Template Strings

E logo abaixo do comentário a segunda coisa que faremos é a declaração de 3 CONSTs. A primeira é uma const `streetName` que recebe uma string `'Rua Curso FrontEnd do Bem'`, daremos um enter para irmos até a linha abaixo e faremos a declaração da nossa segunda const, declararemos a const `amountAnimals` que recebe 2 e daremos um enter para irmos até a linha abaixo e faremos a declaração da nossa terceira const, declaremos a const `AnimalsNames` que recebe `'um Catiorro e um Bichano'`. Ficando assim:

```
const streetName = 'Rua Curso FrontEnd do Bem'  
const amountAnimals = 2  
const animalsNames = 'um Catiorro e um Bichano'
```



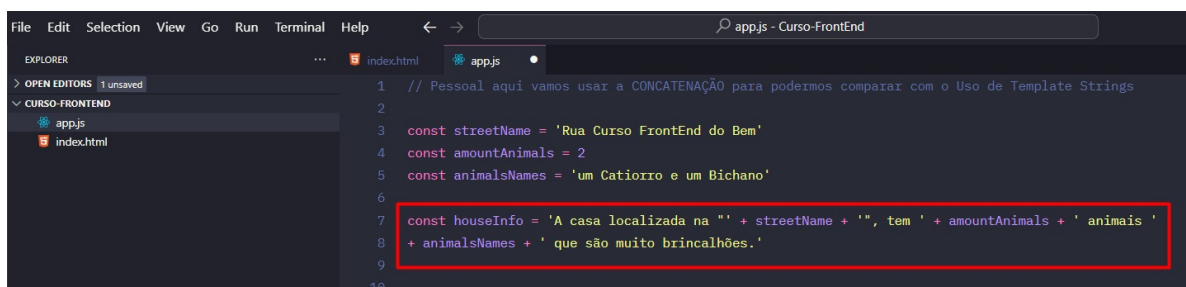
# Concatenação de Strings (+1 exemplo)



```
1 // Pessoal aqui vamos usar a CONCATENAÇÃO para podermos comparar com o Uso de Template Strings
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
```

Criaremos uma String que contem essas 3 consts. Para fazermos isso, criaremos uma const chamada: houseInfo que vai receber 'A casa localizada na "" + streetName + "" tem ' + amountAnimals + ', animais ' + animalsNames + ' que são muito brincalhões.' . Ficando assim:

```
const houseInfo = 'A casa localizada na "" + streetName + "", tem ' + amountAnimals + ' animais ' + animalsNames + ' que são muito brincalhões.'
```

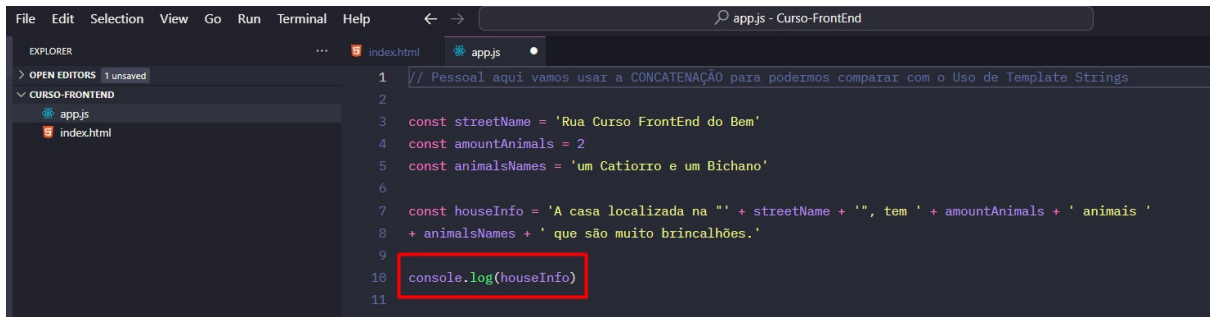


```
1 // Pessoal aqui vamos usar a CONCATENAÇÃO para podermos comparar com o Uso de Template Strings
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = 'A casa localizada na "" + streetName + "", tem ' + amountAnimals + ' animais '
8 + animalsNames + ' que são muito brincalhões.'
9
10
```

Adicionaremos um console.log passando a houseInfo. Ficando assim:

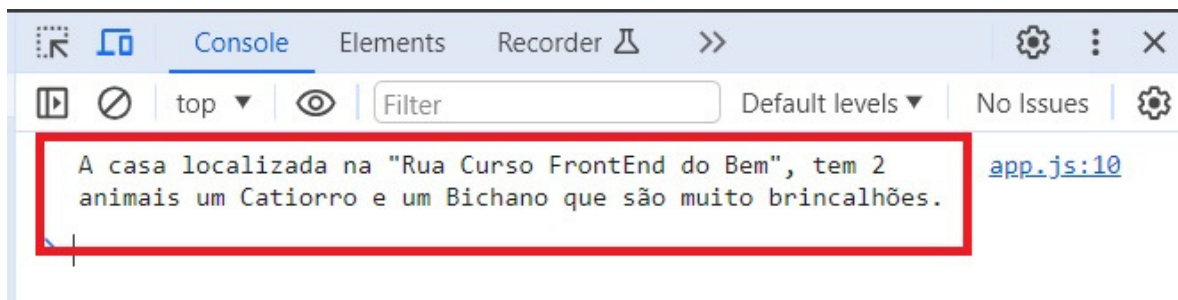
```
console.log(houseInfo)
```

# Concatenação de Strings (+1 exemplo)



```
1 // Pessoal aqui vamos usar a CONCATENAÇÃO para podermos comparar com o uso de Template Strings
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = 'A casa localizada na ' + streetName + ', tem ' + amountAnimals + ' animais '
8 + animalsNames + ' que são muito brincalhões.'
9
10 console.log(houseInfo)
11
```

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que a frase é exibida.



No exemplo acima podemos ver a quantidade de abre e fecha aspas que nós utilizamos dentro de um exemplo considerado "simples" e de concatenações que a string do exemplo (USANDO CONCATENAÇÃO) tem, e com base nisso verificamos que essa string não é muito amigável se quisermos dar alguma manutenção nela e até durante o seu desenvolvimento.

# Template Strings

Veremos uma maneira “diferentona” para escrevermos STRINGS, a qual chamamos de: TEMPLATE STRINGS

Pessoal essa aqui é a TEMPLATE STRINGS, TEMPLATE STRINGS esse aqui é o Pessoal.

Uma das coisas mais “camaradas” que uma Template String possibilita é a de inserirmos as nossas amigas variáveis dentro da string sem ter que fechar uma string, declarar um +, abrir outra string, declarar outro +, fechar string e etc... Parece até o Daniel San em Karatê Kid (põe casaco, tira casaco, põe casaco....) .

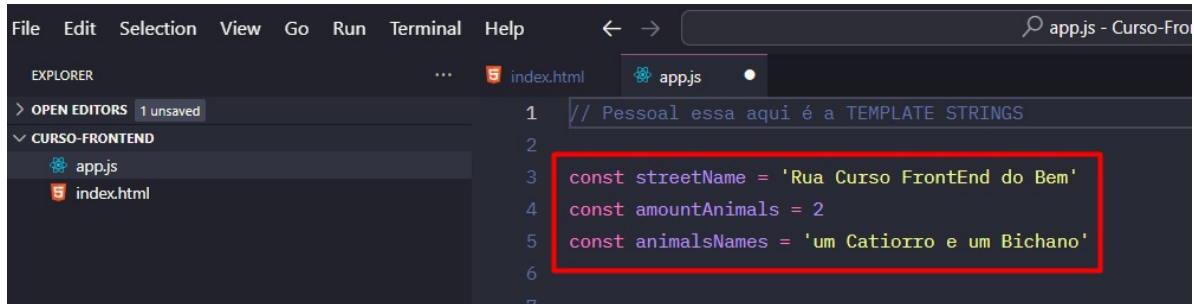
Então Boraaaaaaa abrir o nosso arquivo app.js e na primeira linha dentro desse arquivo criaremos um comentário: `// Pessoal essa aqui é a TEMPLATE STRINGS`

E logo abaixo do comentário a segunda coisa que faremos é a declaração de 3 CONSTS. A primeira é uma const `streetName` que recebe 'Rua Curso FrontEnd do Bem', daremos um enter para irmos até a linha abaixo e faremos a declaração da nossa segunda const, declararemos a const `amountAnimals` que recebe 2 e daremos um enter para irmos até a linha abaixo e faremos a declaração da nossa terceira const, declararemos a const `AnimalsNames` que recebe 'um Catiorro e um Bichano'. Ficando assim:

```
const streetName = 'Rua Curso FrontEnd do Bem'  
const amountAnimals = 2  
const animalsNames = 'um Catiorro e um Bichano'
```



# Template Strings

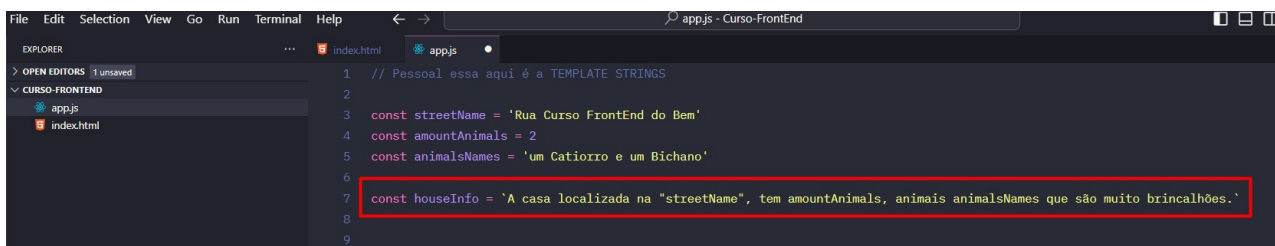


```
File Edit Selection View Go Run Terminal Help
EXPLORER
  > OPEN EDITORS 1 unsaved
    CURSO-FRONTEND
      app.js
      index.html
index.html
1 // Pessoal essa aqui é a TEMPLATE STRINGS
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7
```

Declaremos uma const chamada: houseInfo que receberá: abertura e fechamento de crases `` e dentro dessas crases escreveremos a mesma mensagem utilizada em Concatenação de Strings (conforme vimos na página 3 dessa aula) `A casa localizada na "streetName", tem amountAnimals, animais animalsNames que são muito brincalhões.` Ficando assim:

```
const houseInfo = `A casa localizada na "streetName", tem amountAnimals,
animais animalsNames que são muito brincalhões.`
```

**IMPORTANTÍSSIMO:** Na streetName, na amountAnimals e na animalsNames o que desejamos é que essas 3 variáveis sejam inseridas aqui. Se deixarmos desse jeitinho essa const houseInfo = `A casa localizada na "streetName", tem amountAnimals, animais animalsNames que são muito brincalhões.` é uma string considerada normal que cumprirá a mesma função de uma string que possui abertura e fechamento de aspas E não é isso que queremos, certoooooooooooo!!!!

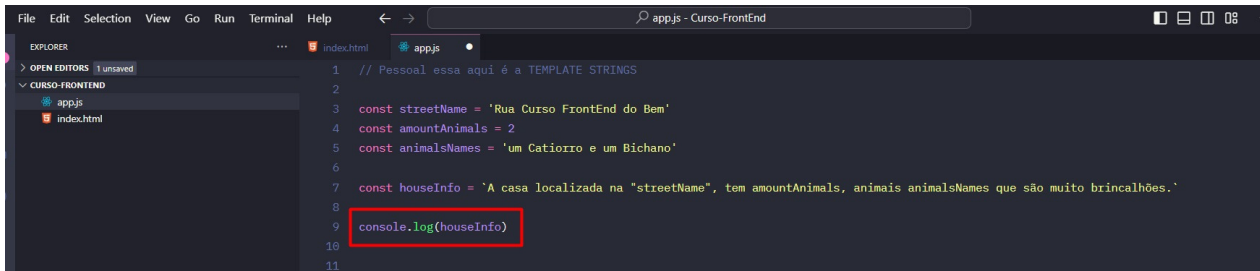


```
File Edit Selection View Go Run Terminal Help
EXPLORER
  > OPEN EDITORS 1 unsaved
    CURSO-FRONTEND
      app.js
      index.html
index.html
1 // Pessoal essa aqui é a TEMPLATE STRINGS
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = `A casa localizada na "streetName", tem amountAnimals, animais animalsNames que são muito brincalhões.`
8
9
```

# Template Strings

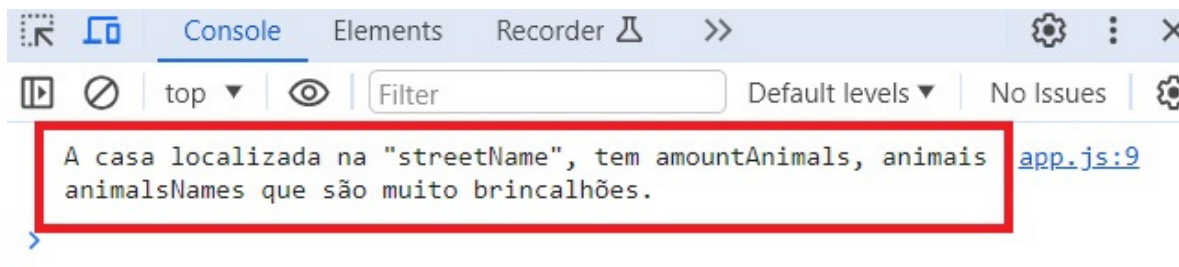
Adicionaremos um `console.log` passando a `houseInfo`. Ficando assim:

```
console.log(houseInfo)
```



```
1 // Pessoal essa aqui é a TEMPLATE STRINGS
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = `A casa localizada na "${streetName}", tem amountAnimals, animais animalsNames que são muito brincalhões.`
8
9 console.log(houseInfo)
```

E ao salvarmos o arquivo `app.js` com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que `streetName`, `amountAnimals` e `animalsNames` foram exibidas como strings, só que com palavras(literalmente) e não inseridas como gostaríamos que fossem.

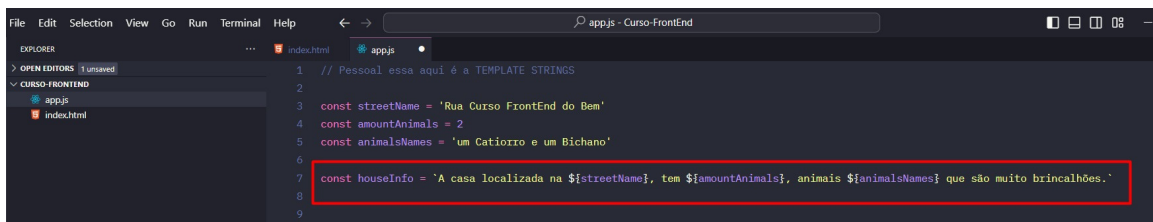


# Template Strings

Igor, Igor... então como vamos fazer para que essas variáveis sejam inseridas da forma correta?

Ótima pergunta: Dentro de uma Template Strings pra inserirmos uma variável, precisamos envolvê-la com um \$ e {}, pegaremos a `streetName`, declararemos um \$ abriremos e fecharemos chaves {} e inserimos a `streetName` dentro dessas chaves `"${streetName}"`, faremos a mesma coisa com a `amountAnimals`, declararemos um \$ abriremos e fecharemos chaves {} e inserimos a `amountAnimals` dentro dessas chaves `"${amountAnimals}"`, faremos igualzinho com a `animalsNames`, declararemos um \$ abriremos e fecharemos chaves e inserimos a `animalsNames` dentro dessas chaves `"${animalsNames}"`. Ficando assim:

```
const houseInfo = `A casa localizada na ${streetName}, tem  
${amountAnimal}, animais ${animalsNames} que são muito brincalhões.`
```



The screenshot shows a code editor with a dark theme. The Explorer panel on the left shows a project named 'CURSO-FRONTEND' with files 'app.js' and 'index.html'. The main editor area shows the following code:

```
1 // Pessoal essa aqui é a TEMPLATE STRINGS
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = `A casa localizada na ${streetName}, tem ${amountAnimals}, animais ${animalsNames} que são muito brincalhões.`
8
9
```

The line 7 is highlighted with a red box.

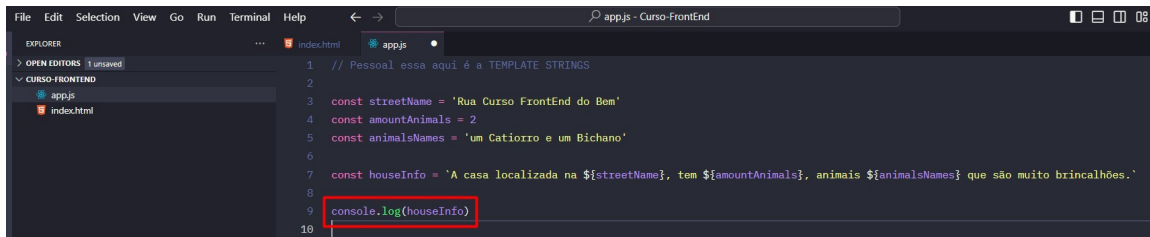
E dando aquela olhada marota na linha acima podemos verificar que existe uma boa diferença de sintaxe entre uma TEMPLATE STRINGS (facilitando d+++++) a nossa leitura e uma STRING NORMAL com abertura e fechamento de aspas.



# Template Strings

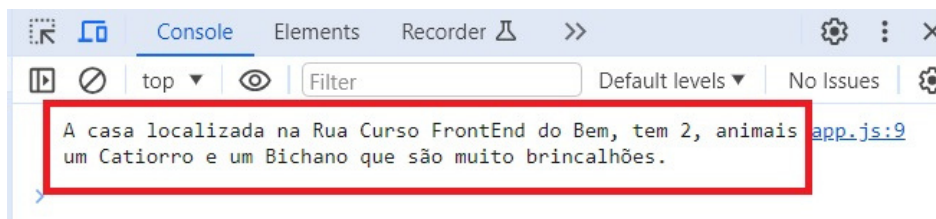
Adicionaremos um `console.log` passando a `houseInfo`. Ficando assim:

```
console.log(houseInfo)
```



```
1 // Pessoal essa aqui é a TEMPLATE STRINGS
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6
7 const houseInfo = `A casa localizada na ${streetName}, tem ${amountAnimals}, animais ${animalsNames} que são muito brincalhões.`
8
9 console.log(houseInfo)
10
```

E ao salvarmos o arquivo `app.js` com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que os valores das 3 consts foram inseridos corretamente dentro da nossa string.



# Template HTML

Um outro jeito legal pra usarmos Template Strings é na criação do brother: Template(s) HTML (Pessoal esse aqui é o Template HTML... Template HTML esse aqui é o Pessoal). Vamos imaginar ludicamente assim como quem não quer nada (despretensioso mesmo) que receberemos algumas informações de forma dinâmica vindo de um lugar específico e esse lugar é um banco de dados com muitassssssssssss informações (não vamos nos preocupar com isso nesse momento, gostaria de deixar essa sementinha plantada em Vossas Cabeças, tanto Template HTML quanto Banco de Dados... esses 2 veremos no decorrer das aulas) e gostaríamos de criar um Template HTML que vai pegar essas informações de forma dinâmica do Banco de Dados e exibir esse Template HTML na página do Navegador(Chrome).

E aqui que vem o “jump of the cat”(pulo do gato).... Podemos usar também a TEMPLATE STRINGS para isso.

Bora criar mais um comentário:// Pessoal esse aqui é o brother TEMPLATE(S) HTML

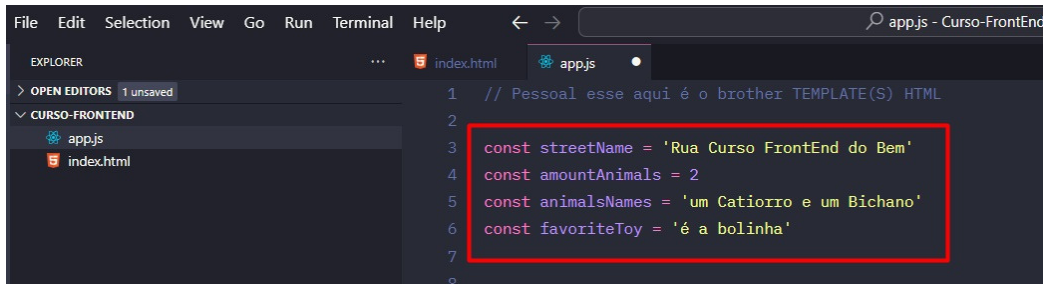
Então abaixo do comentário // Pessoal esse aqui é o brother TEMPLATE(S) HTML . Usaremos as 3 CONSTs dos exemplos anteriores, só que aqui acrescentaremos +1 const, a qual chamaremos de favoriteToy que receberá ‘é a bolinha’ .

```
const streetName = 'Rua Curso FrontEnd do Bem'  
const amountAnimals = 2  
const animalsNames = 'um Catiorro e um Bichano'  
const favoriteToy = 'é a bolinha'
```

Ficando assim:



# Template HTML



Criaremos uma const chamada: houseInfoTemplate que receberá uma template string, abriremos e fecharemos crase ` , ah antes.....simbora dar um enter maroto aqui para quebrarmos uma linha dentro da abertura e fechamento das crases, para deixarmos nossas informações visualmente mais legíveis.

```
const houseInfoTemplate = `
```

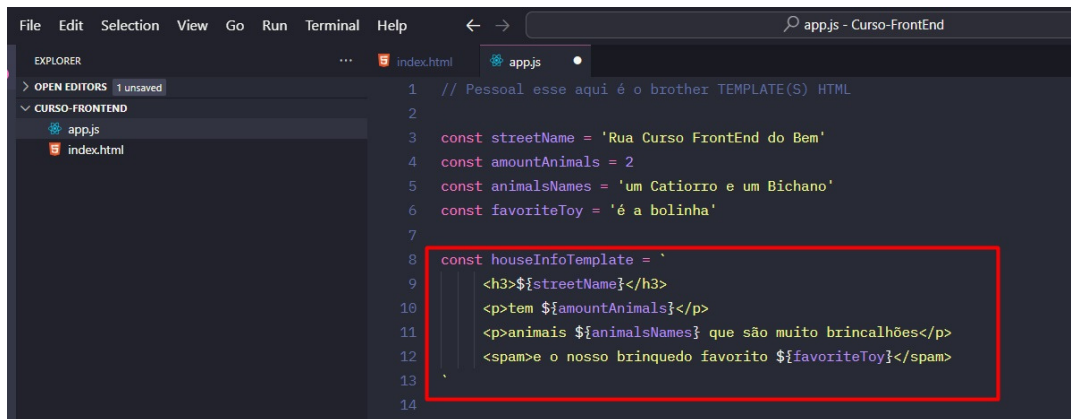
```
  Criaremos um <h3> e fecharemos esse </h3> e dentro desse h3 teremos a
  "${streetName}" e na linha abaixo desse h3, criaremos o nosso primeiro
  <p> e fecharemos esse </p> e dentro desse p escrevermos: tem e a
  "${amountAnimals}" e na linha abaixo desse p, declaremos um segundo
  <p> e fecharemos esse segundo </p> e dentro dele escreveremos:
  animais "${animalsNames}" que são muito brincalhões e na linha abaixo
  desse segundo p, declararemos um spam <spam> e fecharemos esse
  </span> e dentro dele escrevermos: e o nosso brinquedo favorito
  "${favoriteToy}"
`
```

Ficando assim:

```
const houseInfoTemplate = `
  <h3>${streetName}</h3>
  <p>tem ${amountAnimals}</p>
  <p>animais ${animalsNames} que são muito brincalhões</p>
  <spam>e o nosso brinquedo favorito ${favoriteToy}</spam>
`
```



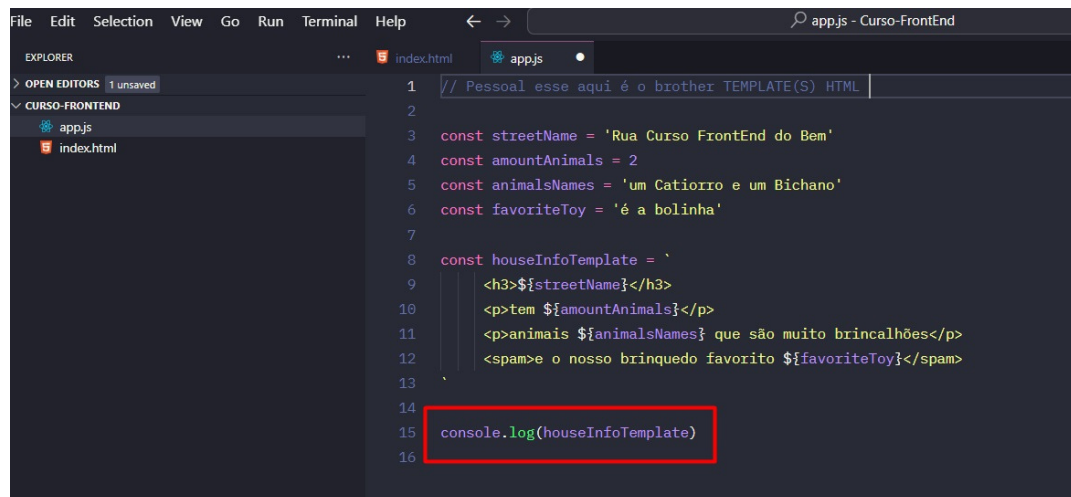
# Template HTML



```
1 // Pessoal esse aqui é o brother TEMPLATE(S) HTML
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6 const favoriteToy = 'é a bolinha'
7
8 const houseInfoTemplate = `
9   <h3>${streetName}</h3>
10   <p>tem ${amountAnimals}</p>
11   <p>animais ${animalsNames} que são muito brincalhões</p>
12   <spam>e o nosso brinquedo favorito ${favoriteToy}</spam>
13 `
14
```

Adicionaremos um `console.log` passando a `houseInfoTemplate`. Ficando assim:

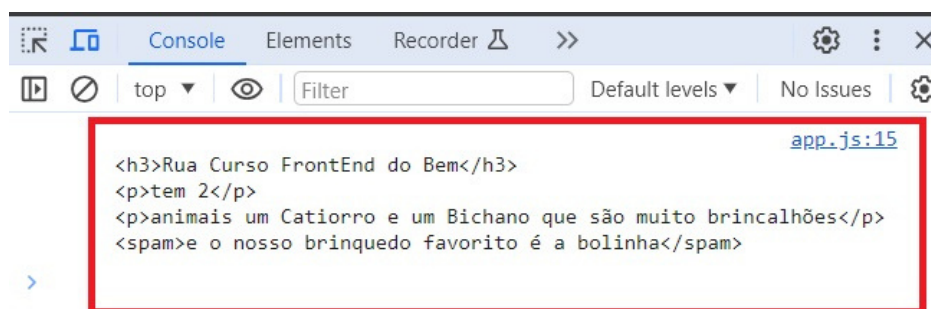
`console.log(houseInfoTemplate)`



```
1 // Pessoal esse aqui é o brother TEMPLATE(S) HTML
2
3 const streetName = 'Rua Curso FrontEnd do Bem'
4 const amountAnimals = 2
5 const animalsNames = 'um Catiorro e um Bichano'
6 const favoriteToy = 'é a bolinha'
7
8 const houseInfoTemplate = `
9   <h3>${streetName}</h3>
10   <p>tem ${amountAnimals}</p>
11   <p>animais ${animalsNames} que são muito brincalhões</p>
12   <spam>e o nosso brinquedo favorito ${favoriteToy}</spam>
13 `
14
15 console.log(houseInfoTemplate)
16
```

# Template HTML

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver o nosso Template HTML(houseInfoTemplate) sendo exibido no console do Navegador(Chrome)



Igor, Igor, Igor, Igor..... surgiu uma dúvida aqui: quando eu uso Template Strings e quando eu uso as strings.... igual o exemplo da concatenação?

Essa pergunta é muito boa, o Igor aqui fica feliz com esse tipo de questionamento: Normalmente quando não precisamos inserir variáveis na string ou fazemos uma quebra de linha não tem muito sentido usar Template Strings. E se durante a criação da string percebermos que ela necessita ter variáveis ou necessita ter aspas ou é um Template Html, a Template Strings vai facilitar e muitooooo tanto a manutenção quanto o código que você estiver escrevendo.

Só reforçando, tanto as Strings (aspas simples e duplas) quanto as Template Strings e não podemos deixar de lado o Template HTML, nós ainda falaremos desses 3 carinhas no decorrer das nossas aulas. Fiquem tranquilos quanto a isso!!!!



# Template HTML

Uma última observação que eu gostaria de passar aqui nessa aula é a seguinte: Vocês notaram que quando falamos de Template HTML, além das constantes e das crases, também voltaram a aparecer um `<h3>`, dois parágrafos `<p>` e um `<spam>`

Lembram que vimos isso nas aulas de HTML (o primeiro pilar desse curso) e que agora será que elas “surgiram do nada” dentro do JavaScript (o terceiro pilar desse curso).

A resposta é Nãooooooooooooooooooooo, os 3 pilares desse curso estão interligados, por isso eu “bato nessa tecla”, aprender a base da base é importantíssimo (que é o intuito desse curso), pois assim quando os pilares começarem a se juntar, tudo vai ficar mais claro para vocês, mesmo quando as coisas começarem a ficar mais complexas.... ah eu lembro que na aula 5 do Curso FrontEnd do Bem eu aprendi que o conceito é esse e o exemplo é esse, então se eu juntar isso com o conteúdo da aula 20 por exemplo, o resultado será esse que estamos vendo aqui (Template HTML).

É exatamente assim que as coisas começam a se encaixar dentro do mundo da programação!!!



**/igor-rebolla**

# Sugestões de prática para essa aula:

1. Refazer e Entender os exemplos dessa aula;
2. Usar valores definidos por vocês para praticar mais sobre: Template Strings e Template HTML;
3. Acrescentar 1 nova const da sua preferência em Template Strings e remover 2 consts em Template HTML.

Colocar isso em prática no VSCode e ver o resultado no console do Navegador.

Programação é prática, curiosidade e repetição!!!!



# O que vamos aprender na aula 21:

Na vigésima aula do curso(Sétima aula de JavaScript) o que veremos: É surpresa – Parte 2 (Vem coisa boa por ai!!!)

**Feedbacks dessa vigésima aula são bem-vindos.**

Nos vemos na vigésima primeira aula que será disponibilizada no dia 11/04/2024 às 8h30 (Horário de Brasília) – Dentro do meu perfil no LinkedIn.



**/igor-rebolla**