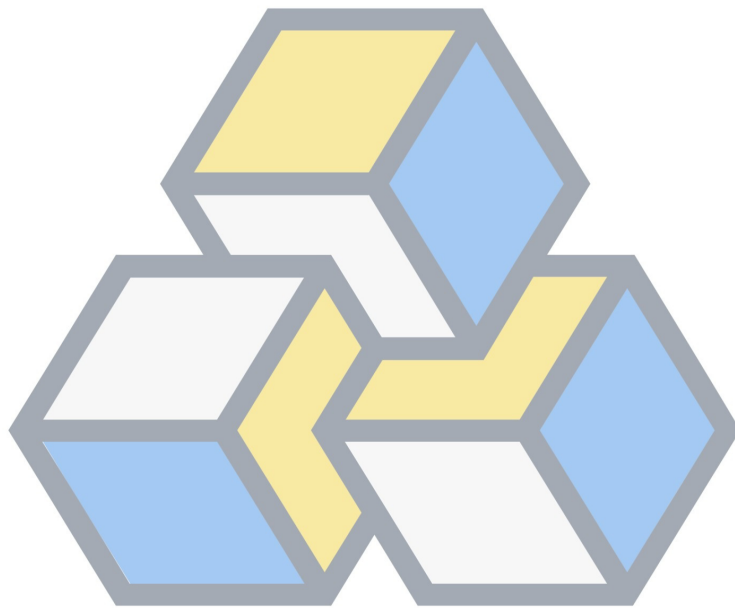


JavaScript Moderno para front-end e back-end

Turma js8374 do curso JS-12



Sumário

1 Exercício: Içamento/Hoisting de declarações	1
1.1 Comentários	1
1.2 Passo a passo com código	1
2 Exercício: Termos de uso, JavaScript e as APIs do Browser que não são JavaScript	4
2.1 Comentários	4
2.2 Passo a passo com código	4
3 Exercício: Escolhendo uma página inicial - variáveis com `let` e comparações	6
3.1 Comentários	6
3.2 Passo a passo com código	6
4 Exercício: Type coercing	7
4.1 Comentários	7
4.2 Passo a passo com código	8
5 Exercício: Módulos com IIFEs	10
5.1 Comentários	10
5.2 Passo a passo com código	12
6 Exercício: Módulos com ESModules	13
6.1 Comentários	13
6.2 Passo a passo com código	13
7 Exercício: Menos *type coercing* e centralizando módulos com `main.js`	16
7.1 Comentários	16
7.2 Passo a passo com código	16
8 Exercício: Organizando a aplicação com módulos.	18
8.1 Comentários	18
8.2 Passo a passo com código	18
9 Exercício: Início da página de configuração	21

9.1 Comentários	21
9.2 Passo a passo com código	21
10 Exercício: Configurando ESModules no VSCode	23
10.1 Comentários	23
10.2 Passo a passo com código	23
11 Exercício: Módulos com mais de um valor exportado	25
11.1 Comentários	25
11.2 Passo a passo com código	25
12 Exercício: Funções: callbacks, encapsulamento e closures	29
12.1 Comentários	29
12.2 Passo a passo com código	29

EXERCÍCIO: IÇAMENTO/HOISTING DE DECLARAÇÕES

1.1 COMENTÁRIOS

O código abaixo que cria a variável depois de usá-la:

```
$inputEndereco.value = paginaInicial
$janelaPrincipal.src = paginaInicial
var paginaInicial = 'http://google.com'
```

É a mesma coisa que esse próximo código:

```
var paginaInicial = undefined
$inputEndereco.value = paginaInicial
$janelaPrincipal.src = paginaInicial
paginaInicial = 'http://google.com'
```

O JS manda a declaração da variável pra cima, faz o **hoisting**. Esse código da página inicial vai executar sem erros porque `undefined` é um valor válido.

"hoisting foi um detalhe de implementação" – Brendan Eich (criador do JavaScript)

Em outras linguagens usar uma variável antes de criá-la resultaria num erro que impediria que o código seguisse com a execução!

ECMAScript não vai mudar o comportamento de algo. ECMAScript não quer quebrar códigos que já usem o `var` com hoisting de propósito.

ECMAScript fez uma nova forma de declarar variáveis que segue as outras linguagens:

```
const paginaInicial = 'http://google.com'
let paginaInicial = 'http://google.com'
```

1.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `index.html` na pasta **raíz do projeto** faça as seguintes alterações:

```
# index.html
```

```

<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
  <input type="image" src="images/libCake/reicarregar.svg">
-  <input type="text" value="blank">
+  <input type="text" id="$inputEndereco" value="blank">
</header>

- <iframe src="blank" frameborder="0"></iframe>
+ <iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

- <script src="scripts/cake-8709f3abc4.js"></script>
+ <script src="scripts/cake-8709f3abc4.js"></script>
+
+ <script src="scripts/pedePaginaInicial.js"></script>

```

2. Nesse código, testaremos o hoisting para ver ele acontecendo. Acesse o console do navegador apertando F12 e execute o código abaixo nele.

Com o seguinte código, veja como nossa página inicial se torna **undefined** e como que no *console* (F12) não há erros de JavaScript por termos usado a variável antes de criá-la.

console.js

```

+
+ $janelaPrincipal.src = paginaInicial
+ $inputEndereco.value = paginaInicial
+
+ var paginaInicial = prompt("Escolha a página inicial")

```

3. Nesse código, veremos como o ECMAScript 2015 tratou o hoisting com suas novas variáveis **let** e **const** . Acesse o console do navegador apertando F12 e execute o código abaixo nele.

Com o seguinte código, veja como o console apresenta um erro de JavaScript dizendo que a variável não existe.

console.js

```

+
+ $janelaPrincipal.src = paginaInicial
+ $inputEndereco.value = paginaInicial
+
+ const paginaInicial = prompt("Escolha a página inicial")

```

4. Crie o arquivo **pedePaginaInicial.js** na pasta **scripts** com o seguinte código:

scripts/pedePaginaInicial.js

```

+
+ const paginaInicial = prompt("Escolha a página inicial")
+
+ $janelaPrincipal.src = paginaInicial
+ $inputEndereco.value = paginaInicial
+

```


EXERCÍCIO: TERMOS DE USO, JAVASCRIPT E AS APIS DO BROWSER QUE NÃO SÃO JAVASCRIPT

2.1 COMENTÁRIOS

Se a pessoa não aceitar os termos de uso, precisamos impedir que ela continue usando o Cake.

Para isso, usaremos diversas API's do navegador que nos permitem controlá-lo com JavaScript:

- `prompt`
- `confirm`
- `alert`
- `window.close`
- `localStorage`

Apesar de escrevermos o código com JavaScript, essas API's não fazem parte da especificação *ECMAScript*.

Já os valores retornados por essas API's e os valores que passamos como parâmetros para elas, são todos especificados pela ECMA:

- `template strings` para strings multi-linha e interpolação de variáveis com `` Um texto aqui ${nome} ``
- `booleanos` valores *true* e *false*
- `null` representa algo vazio

2.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `index.html` na pasta **raíz do projeto** faça as seguintes alterações:

`index.html`

```
<meta charset="utf-8">
```

```
<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>
```

```

<header>
  <input type="image" src="images/libCake/recarregar.svg">
  <input type="text" id="$inputEndereco" value="blank">
</header>

<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

<script src="scripts/cake-8709f3abc4.js"></script>

+<script src="scripts/termosDeUso.js"></script>
  <script src="scripts/pedePaginaInicial.js"></script>

```

2. Crie o arquivo **termosDeUso.js** na pasta **scripts** com o seguinte código:

scripts/termosDeUso.js

```

+if(localStorage.getItem('aceitouTermos') === null){
+  // CONST é Variável. Dinâmico
+  const nome = prompt('Qual o seu nome')
+
+  const aceitouTermos = confirm(`
+    Olá ${nome}!
+
+    Antes de usar o Cake, precisamos que
+    você aceite nossos termos de uso:
+
+    • Você aceita que este software foi
+      feito por pessoas que participaram
+      do curso de JavaScript.
+    • Você aceita que o código dessas
+      pessoas pode acessar tudo o que
+      você digitar aqui.
+  `)
+
+  if(!aceitouTermos) {
+    alert(nome + ', não podemos continuar juntos')
+    window.close()
+  } else {
+    localStorage.setItem('aceitouTermos', aceitouTermos)
+  }
+}

```


EXERCÍCIO: ESCOLHENDO UMA PÁGINA INICIAL - VARIÁVEIS COM `LET` E COMPARAÇÕES

3.1 COMENTÁRIOS

Seguimos fazendo a funcionalidade de pedir a página inicial para a pessoa, deixando ela digitar um endereço sem `http://` no começo. Nós completamos o endereço para a pessoa caso ela não digite.

Comparações de igualdade e diferença no JavaScript iguais as de outras linguagens são com três caracteres: `===` e `!==`.

3.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `pedePaginaInicial.js` na pasta `scripts` faça as seguintes alterações:

```
# scripts/pedePaginaInicial.js

-const paginaInicial = prompt("Escolha a página inicial")
+let paginaInicial = prompt("Escolha a página inicial")
+
+if (
+  paginaInicial.substring(0, 7) !== 'http://' &&
+  paginaInicial.substring(0,8) !== 'https://'
+) {
+  // Assignement Atribuição
+  paginaInicial = 'http://' + paginaInicial
+}

$janelaPrincipal.src = paginaInicial
$inputEndereco.value = paginaInicial
```

EXERCÍCIO: TYPE COERCING

4.1 COMENTÁRIOS

Nesse ponto, deixaremos que a pessoa possa digitar nada ou cancelar o `prompt` que pede pela página inicial.

O seguinte código:

```
if (paginaInicial !== null && paginaInicial == '') {}
```

Pode ter o mesmo efeito que esse código:

```
if (paginaInicial) {}
```

Quando digo "pode" ter o mesmo efeito é porque o que está acontecendo ali é a conversão de `paginaInicial` para booleano:

```
Boolean(paginaInicial)
```

Esse código pode ter o resultado `false` para os seguintes valores:

```
Boolean(null)
Boolean(undefined)
Boolean('')
Boolean(0)
Boolean(NaN)
```

Ou seja, fazer `if (paginaInicial)` significa muito mais condições do que `if (paginaInicial !== null && paginaInicial == '')`

Essa conversão de tipos que é feita pelo JavaScript e não pelo nosso código é chamada de *Implicit Type Coercion*

Ela sempre vai acontecer quando algum tipo específico é esperado, no caso do `if()`, um Booleano.

Um caso de type coercion acontece quando fazemos uma concatenação: Se página inicial for null: `'http://' + paginaInicial`

``paginaInicial`` será convertida para a string `'null'` e terá como resultado: `'http://null'`

Em uma operação/comparação de igualdade com 2 iguais `==` isso também acontece:

```
const paginaInicial = null
paginaInicial == 'null' // true
```

Já no caso de 3 iguais === , isso não acontece:

```
const paginaInicial = null
paginaInicial === 'null' // true
```

4.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **index.html** na pasta **raíz do projeto** faça as seguintes alterações:

index.html

```
<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
-   <input type="image" src="images/libCake/recarregar.svg">
+   <input type="image" src="images/libCake/recarregar.svg" onclick="window.location.reload()">
    <input type="text" id="$inputEndereco" value="blank">
</header>

<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

<script src="scripts/cake-8709f3abc4.js"></script>

<script src="scripts/termosDeUso.js"></script>
<script src="scripts/pedePaginaInicial.js"></script>
```

2. No arquivo **termosDeUso.js** na pasta **scripts** faça as seguintes alterações:

scripts/termosDeUso.js

```
-if(localStorage.getItem('aceitouTermos') === null){
+if(!localStorage.getItem('aceitouTermos')){
    // CONST é Variável. Dinâmico
    const nome = prompt('Qual o seu nome')

    const aceitouTermos = confirm(`
        Olá ${nome}!

        Antes de usar o Cake, precisamos que
        você aceite nossos termos de uso:

        • Você aceita que este software foi
          feito por pessoas que participaram
          do curso de JavaScript.
        • Você aceita que o código dessas
          pessoas pode acessar tudo o que
          você digitar aqui.
    `)

    if(!aceitouTermos) {
        alert(nome + ', não podemos continuar juntos')
        window.close()
    } else {
        localStorage.setItem('aceitouTermos', aceitouTermos)
    }
}
```

3. No arquivo **pedePaginaInicial.js** na pasta **scripts** faça as seguintes alterações:

scripts/pedePaginaInicial.js

```
-let paginaInicial = prompt("Escolha a página inicial")  
+let paginaInicial = localStorage.getItem('paginaInicial')  
  
-if (-  
-    paginaInicial.substring(0, 7) !== 'http://' &&  
-    paginaInicial.substring(0, 8) !== 'https://'  
-) {  
-    // Assignement Atribuição  
-    paginaInicial = 'http://' + paginaInicial  
+if(!paginaInicial) {  
+    paginaInicial = prompt("Escolha a página inicial")  
+}  
  
-$janelaPrincipal.src = paginaInicial  
-$inputEndereco.value = paginaInicial  
+if(paginaInicial) {  
+    if (  
+        paginaInicial.substring(0, 7) !== 'http://' &&  
+        paginaInicial.substring(0, 8) !== 'https://'  
+    ) {  
+        // Assignement Atribuição  
+        paginaInicial = 'http://' + paginaInicial  
+    }  
  
+    $janelaPrincipal.src = paginaInicial  
+    $inputEndereco.value = paginaInicial  
+  
+    localStorage.setItem('paginaInicial', paginaInicial)  
+}
```

EXERCÍCIO: MÓDULOS COM IIFES

5.1 COMENTÁRIOS

Função Anônima Imediatamente Invocada

Immediately Invoked Function Expression

Vulgo: *IIFE*

Porque? Porque navegadores pegam os arquivos nas tags `<script>` e fazem com que todos rodem no mesmo escopo. Toda variável que esteja 'solta' no arquivo fica global!

Todo `var` dentro de função, só existe dentro daquela função, enquanto ela é executada. Tirando isso, ela sempre estará global, mesmo dentro de blocos de código em `if`, `else`, `while` e `for`. Dizemos que o escopo da `var` é dinâmico.

Já, `const` e `let` têm escopo léxico. Traduzindo: Elas ficam contidas em blocos de código. Variáveis com `const` e `let` não ficam globais se estiverem dentro de um `if`, `else`, `while`, `for` ou `function`.

Por que não deixar global?

Para poder dar o nome das suas variáveis do jeito que quiser. Sem impedir que outros códigos do sistema usem um nome de variável igual. Cada arquivo fica na sua.

Por que `var` permite que você recrie a variável com outro valor. Se dois arquivos usam o mesmo nome de `var`:

```
// Arquivo A
var contador = 2
===
// Arquivo B
var contador = 4
```

O valor da variável `contador` será `4` em todos os arquivos, inclusive no Arquivo A, que queria que a variável tivesse o valor `2`.

Outros códigos não relacionados com o seu podem modificar variáveis no seu código, muitas vezes sem querer. Em casos de nomes de variáveis comuns como `contador` e `i` utilizados em loops, isso pode ser bem ruim. Com `const` e `let` as variáveis não podem ser re-declaradas. O seguinte caso

resultaria em um erro que encerraria a execução do código:

```
// Arquivo A
const numero = 2
===
// Arquivo B
const numero = 4
```

O Arquivo B lançará um erro dizendo que não é possível redeclarar a `const numero`. A mesma coisa aconteceria com o `let`

A solução encontrada para todos os casos:

```
(function() {
    // Qualquer `var`, `const` e `let` aqui dentro não será colocada no escopo global
})();
```

Uma função sem nome que é executada assim que é criada. Ninguém mais consegue chamar ela depois. Executa uma vez e acabou. As variáveis ali dentro estão presas ali dentro. Porém.... se esquecer de colocar o `var` antes do nome da variável:

```
(function() {
    numero = 2
})();
```

O processo de **hoisting** faz com que a variável seja declarada assim, por debaixo dos panos do JS:

```
var numero = undefined
(function() {
    numero = 2
})();
```

Variável `numero` está global! A solução foi dar um jeito de isso dar erro nas versões novas do JS, mas sem quebrar versões antigas. Não podiam fazer esse código dar erro pq muita gente já usava esse código dessa maneira de propósito. A solução, uma string solta no meio do código:

```
(function() {
    "use strict"
    numero = 2
})();
```

O código acima resultará num erro, pois o modo 'strict' do JavaScript foi ativado quando ele encontra a string "use strict". Chamamos isso de módulo. Cada módulo pode fazer as variáveis que quiser sem interferir sem querer com o escopo global. Mas se for de propósito, ainda consegue:

```
let numero
(function() {
    "use strict"
    numero = 2
})();
```

O código roda sem erros!

Se quiser exportar alguma variável para o escopo global:

```
const variavelGlobal = (function() {
    "use strict"
    const numero = 2
    const variavelExportada = numero + 365
    return variavelExportada
})();
```

Criamos uma `variavelGlobal` que tem o valor `367`, resultado de ter rodado o código do nosso módulo.

Esse é o único jeito de criar módulos no ECMAScript antigo.

5.2 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo `aceitouAnteriormente.js` na pasta `scripts` com o seguinte código:

`scripts/aceitouAnteriormente.js`

```
+const aceitouSalvar = (function() {
+    "use strict"
+    const aceitouAnteriormente = localStorage.getItem("aceitouSalvar")
+
+    let aceitouSalvar
+
+    if(!aceitouAnteriormente){
+        aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')
+        if(!aceitouSalvar) {
+            alert('Você pode mudar isso na página de configurações')
+        }
+
+        localStorage.setItem("aceitouSalvar", aceitouSalvar)
+    }
+
+    return aceitouSalvar
+})();
```

EXERCÍCIO: MÓDULOS COM ESMODULES

6.1 COMENTÁRIOS

Pessoal do node não quis isso e adotou um padrão chamado "CommonJS". Nele, todo arquivo já é um módulo. E nenhuma variável é colocada no escopo global.

Foi criado um mecanismo de importação e exportação de valores.

Quando algum arquivo quer acessar informações de outro:

```
// Arquivo a.js
const valorDeOutroArquivo = require('b.js')
```

6.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **index.html** na pasta **raíz do projeto** faça as seguintes alterações:

index.html

```
<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
  <input type="image" src="images/libCake/reicarregar.svg" onclick="window.location.reload()">
  <input type="text" id="$inputEndereco" value="blank">
</header>

<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

<script src="scripts/cake-8709f3abc4.js"></script>

<script src="scripts/termosDeUso.js"></script>
- <script src="scripts/pedePaginaInicial.js"></script>
+ <script type="module" src="scripts/pedePaginaInicial.js"></script>
```

2. No arquivo **aceitouAnteriormente.js** na pasta **scripts** faça as seguintes alterações:

scripts/aceitouAnteriormente.js

```
- const aceitouSalvar = (function() {
-   "use strict"
-   const aceitouAnteriormente = localStorage.getItem("aceitouSalvar")
-
-   let aceitouSalvar
```



```

-
-   if(!aceitouAnteriormente){
-       aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')
-       if(!aceitouSalvar) {
-           alert('Você pode mudar isso na página de configurações')
-       }
-       localStorage.setItem("aceitouSalvar", aceitouSalvar)
-   }
-
-   return aceitouSalvar
- }()
+const aceitouAnteriormente = localStorage.getItem("aceitouSalvar")
+
+let aceitouSalvar
+
+if(!aceitouAnteriormente){
+    aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')
+    if(!aceitouSalvar) {
+        alert('Você pode mudar isso na página de configurações')
+    }
+    localStorage.setItem("aceitouSalvar", aceitouSalvar)
+}
+
+export default aceitouSalvar

```

3. No arquivo **pedePaginaInicial.js** na pasta **scripts** faça as seguintes alterações:

scripts/pedePaginaInicial.js

```

-let paginaInicial = localStorage.getItem('paginaInicial')
-if(!paginaInicial) {
-    paginaInicial = prompt("Escolha a página inicial")
-}
-
-if(paginaInicial) {
-    if (
-        paginaInicial.substring(0, 7) !== 'http://' &&
-        paginaInicial.substring(0, 8) !== 'https://'
-    ) {
-        // Assignement Atribuição
-        paginaInicial = 'http://' + paginaInicial
-    }
-
-    $janelaPrincipal.src = paginaInicial
-    $inputEndereco.value = paginaInicial
-    localStorage.setItem('paginaInicial', paginaInicial)
-}

+import aceitouSalvar from './aceitouAnteriormente.js'

+if(aceitouSalvar === null){
+
+    let paginaInicial = localStorage.getItem('paginaInicial')

+    if(!paginaInicial) {
+        paginaInicial = prompt("Escolha a página inicial")
+    }

+    if(paginaInicial) {
+        if (
+            paginaInicial.substring(0, 7) !== 'http://' &&
+            paginaInicial.substring(0, 8) !== 'https://'
+        ) {

```

```

+         // Assignement Atribuição
+         paginaInicial = 'http://' + paginaInicial
+     }
+
+     $janelaPrincipal.src = paginaInicial
+     $inputEndereco.value = paginaInicial
+
+     localStorage.setItem('paginaInicial', paginaInicial)
+ }
+}

```

EXERCÍCIO: MENOS *TYPE COERCING* E CENTRALIZANDO MÓDULOS COM `MAIN.JS`

7.1 COMENTÁRIOS

Neste exercício, nós centralizamos nossos código JS no módulo `main.js` que faz todos os `import` necessários para que nossa app rode.

Também deixamos de depender do *type coercing* para evitar *bugs* e passamos a verificar exatamente os tipos de valores que esperamos das variáveis nos nossos `if` s.

7.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `index.html` na pasta **raíz do projeto** faça as seguintes alterações:

`index.html`

```
<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
  <input type="image" src="images/libCake/recarregar.svg" onclick="window.location.reload()">
  <input type="text" id="$inputEndereco" value="blank">
</header>

<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

<script src="scripts/cake-8709f3abc4.js"></script>

- <script src="scripts/termosDeUso.js"></script>
-
- <script type="module" src="scripts/pedePaginaInicial.js"></script>
+ <script type="module" src="scripts/main.js"></script>
```

2. Crie o arquivo `main.js` na pasta **scripts** com o seguinte código:

`scripts/main.js`

```
+import './termosDeUso.js'
+import './pedePaginaInicial.js'
```

3. Na pasta **scripts**, renomeie o arquivo **aceitouAnteriormente.js** para **aceitouSalvar.js**
Faça também as seguintes modificações no código:

scripts/aceitouSalvar.js

```
-const aceitouAnteriormente = localStorage.getItem("aceitouSalvar");  
+let aceitouSalvar = JSON.parse(localStorage.getItem("aceitouSalvar"))  
  
-let aceitouSalvar  
-  
-if(!aceitouAnteriormente){  
+if(aceitouSalvar === null){  
    aceitarSalvar = confirm('Você aceita que a gente salve suas informações?')  
    if(!aceitouSalvar) {  
        alert('Você pode mudar isso na página de configurações')  
    }  
  
    localStorage.setItem("aceitouSalvar", aceitarSalvar)  
}  
  
export default aceitarSalvar
```

4. No arquivo **pedePaginaInicial.js** na pasta **scripts** faça as seguintes alterações:

scripts/pedePaginaInicial.js

```
-import aceitouSalvar from './aceitouAnteriormente.js'  
+import aceitarSalvar from './aceitouSalvar.js'  
  
-if(aceitouSalvar === null){  
-  
+if(aceitouSalvar === null || aceitarSalvar === true){  
    let paginaInicial = localStorage.getItem('paginaInicial')  
  
    if(!paginaInicial) {  
        paginaInicial = prompt("Escolha a página inicial")  
    }  
  
    if(paginaInicial) {  
        if (  
            paginaInicial.substring(0, 7) !== 'http://' &&  
            paginaInicial.substring(0,8) !== 'https://'   
        ) {  
            // Assignement Atribuição  
            paginaInicial = 'http://' + paginaInicial  
        }  
  
        $janelaPrincipal.src = paginaInicial  
        $inputEndereco.value = paginaInicial  
  
        localStorage.setItem('paginaInicial', paginaInicial)  
    }  
-  
}
```

EXERCÍCIO: ORGANIZANDO A APLICAÇÃO COM MÓDULOS.

8.1 COMENTÁRIOS

Nesse exercício, fizemos uma grande refatoração para organizar melhor nossa aplicação com *ESModules*.

Criamos um módulo agregador `scripts/pedeInfosIniciais/index.js` que importa todos os módulos que pedem informações na inicialização do navegador.

Também criamos módulos dentro da pasta `storage/` que evitam que a gente duplique o código de acesso às informações do `localStorage`.

Refatoramos todos os nossos códigos para utilizar essa estrutura nova do projeto.

8.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `main.js` na pasta `scripts` faça as seguintes alterações:

```
# scripts/main.js

- import './termosDeUso.js'
- import './pedePaginaInicial.js'
+ import './scripts/pedeInfosIniciais/index.js'
```

2. Crie o arquivo `index.js` na pasta `scripts/pedeInfosIniciais` com o seguinte código:

```
# scripts/pedeInfosIniciais/index.js

+ import './termosDeUso.js'
+ import './pedeAceitouSalvar.js'
+ import './pedePaginaInicial.js'
```

3. Mova o arquivo `aceitouSalvar.js` para a pasta `scripts/pedeInfosIniciais` e o renomeie para `pedeAceitouSalvar.js`. No momento o arquivo está na pasta `scripts`. Faça também as seguintes modificações no código:

```
# scripts/pedeInfosIniciais/pedeAceitouSalvar.js

- let aceitouSalvar = JSON.parse(localStorage.getItem("aceitouSalvar"))
+ import aceitouSalvar from './scripts/storage/aceitouSalvar.js'
```

```

    if(aceitouSalvar === null){
-   aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')
+   // shadowing/sombra no módulo
+   // redeclerando com o mesmo nome
+   const aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')
+
+   if(!aceitouSalvar) {
+       alert('Você pode mudar isso na página de configurações')
+   }
+
+   localStorage.setItem("aceitouSalvar", aceitouSalvar)
- }
-
- export default aceitouSalvar
+ }

```

4. Mova o arquivo **pedePaginaInicial.js** para a pasta **scripts/pedeInfosIniciais** . No momento este arquivo está na pasta **scripts** . Faça também as seguintes modificações no código:

scripts/pedeInfosIniciais/pedePaginaInicial.js

```

- import aceitouSalvar from './aceitouSalvar.js'
+ import aceitouSalvar from './scripts/storage/aceitouSalvar.js'
+ import paginaInicial from './scripts/storage/paginaInicial.js'

    if(aceitouSalvar === null || aceitouSalvar === true){
-   let paginaInicial = localStorage.getItem('paginaInicial')
+   // Sem shadowing
+   let paginaInicialDefault = paginaInicial

-   if(!paginaInicial) {
-       paginaInicial = prompt("Escolha a página inicial")
+   if(!paginaInicialDefault) {
+       paginaInicialDefault = prompt("Escolha a página inicial")
+   }

-   if(paginaInicial) {
+   if(paginaInicialDefault) {
+       if (
-       paginaInicial.substring(0, 7) !== 'http://' &&
-       paginaInicial.substring(0, 8) !== 'https://'
+       paginaInicialDefault.substring(0, 7) !== 'http://' &&
+       paginaInicialDefault.substring(0, 8) !== 'https://'
+       ) {
+           // Assignement Atribuição
-       paginaInicial = 'http://' + paginaInicial
+       paginaInicialDefault = 'http://' + paginaInicialDefault
+       }

-       $janelaPrincipal.src = paginaInicial
-       $inputEndereco.value = paginaInicial
+       $janelaPrincipal.src = paginaInicialDefault
+       $inputEndereco.value = paginaInicialDefault

-       localStorage.setItem('paginaInicial', paginaInicial)
+       localStorage.setItem('paginaInicial', paginaInicialDefault)
+   }
}

```

5. Mova o arquivo **termosDeUso.js** para a pasta **scripts/pedeInfosIniciais** . No momento este

arquivo está na pasta **scripts** . Faça também as seguintes modificações no código:

```
# scripts/pedeInfosIniciais/termosDeUso.js
```

6. Crie o arquivo **aceitouSalvar.js** na pasta **scripts/storage** com o seguinte código:

```
# scripts/storage/aceitouSalvar.js
```

```
+export default JSON.parse(localStorage.getItem('aceitouSalvar'))
```

7. Crie o arquivo **paginaInicial.js** na pasta **scripts/storage** com o seguinte código:

```
# scripts/storage/paginaInicial.js
```

```
+export default localStorage.getItem('paginaInicial')
```

EXERCÍCIO: INÍCIO DA PÁGINA DE CONFIGURAÇÃO

9.1 COMENTÁRIOS

Criamos uma nova página de configuração que será acessada quando a pessoa clicar no ícone da engrenagem ao lado da barra de endereço.

Nessa página de configuração, já visualizaremos as informações salvas no `localStorage`, mas não acessaremos ele diretamente. Podemos usar nossos novos módulos dentro de `storage/`.

9.2 PASSO A PASSO COM CÓDIGO

1. No arquivo `index.html` na pasta **raíz do projeto** faça as seguintes alterações:

`index.html`

```
<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
  <input type="image" src="images/libCake/recarregar.svg" onclick="window.location.reload()">
  <input type="text" id="$inputEndereco" value="blank">
+   <a href="config.html">
+     
+   </a>
</header>

<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>

<script src="scripts/cake-8709f3abc4.js"></script>

<script type="module" src="scripts/main.js"></script>
```

2. No arquivo `config.html` na pasta **raíz do projeto** faça as seguintes alterações:

`config.html`

```
<link rel="stylesheet" href="styles/cake-config-33d6f649ee.css"/>

<h1>Configurações</h1>

<label>
```



```

    Página inicial:
-   <input type="text">
+   <input type="text" id="$inputPaginaInicial">
</label>

<label>
-   Permissão de compartilhamento:
-   <input type="checkbox">
+   Permissão para armazenamento:
+   <input type="checkbox" id="$inputPermitiuSalvar">
</label>

<button>
    Salvar configurações
</button>
<a href="/">Voltar</a>
+
+<script type="module" src="scripts/paginaConfiguracao.js"></script>

```

3. Crie o arquivo **paginaConfiguracao.js** na pasta **scripts** com o seguinte código:

scripts/paginaConfiguracao.js

```

+import paginaInicial from '/scripts/storage/paginaInicial.js'
+import aceitouSalvar from '/scripts/storage/aceitouSalvar.js'
+
+$inputPaginaInicial.value = paginaInicial
+$inputPermitiuSalvar.checked = aceitouSalvar

```

EXERCÍCIO: CONFIGURANDO ESMODULES NO VSCODE

10.1 COMENTÁRIOS

Nos nossos códigos, passamos a importar nossos módulos com um caminho absoluto:

```
import aceitouSalvar from '/scripts/storage/aceitouSalvar.js'
```

O VSCode por padrão acha que endereços iniciados com `/` se referem à pasta raiz do seu computador, e não do nosso projeto. Isso se deve pelo fato de que ESMODULES ainda não são amplamente adotados.

Em projetos que usam `import` e `export`, na maioria das vezes, o código é passado para algum compilador que traduz esses módulos para módulos de padrões mais antigos, como *CommonJS* ou até mesmo *IIFEs*, que comentamos anteriormente. Esse processo de compilação geralmente ocorre na máquina de quem está desenvolvendo, e assim, caminhos iniciados em `/` são considerados também como a partir da pasta raiz do computador.

Com ESMODULES nos browsers, esses nossos imports são URLs relativas sempre ao endereço em que a aplicação está rodando. No nosso caso, relativas à pasta raiz do nosso projeto.

A configuração abaixo diz que a pasta raiz do nosso projeto deve ser considerada sempre que algum endereço de `import` iniciado com `/` for encontrado no nosso código.

Com essa configuração, ganhamos diversas facilidades do editor na hora de acessar, navegar e até mesmo mudar códigos entre nossos módulos.

10.2 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo `jsconfig.json` na pasta **raiz do projeto** com o seguinte código:

```
# jsconfig.json

+{
+  "compilerOptions": {
+    "baseUrl": ".",
+    "module": "esnext",
+    "paths": {
```

```
+      "/"*": ["/.*"]
+      }
+    }
+}
```

EXERCÍCIO: MÓDULOS COM MAIS DE UM VALOR EXPORTADO

11.1 COMENTÁRIOS

Nesse código implementamos o botão de salvar as configurações na página de configurações.

Para evitar código duplicado que altera o `localStorage`, criamos funções *setter* em nossos módulos de `storage/` que serão responsáveis por alterar o `localStorage`.

Refatoramos nosso código anterior para que não haja mais acesso ao `localStorage` fora desses módulos.

11.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **aceitouSalvar.js** na pasta **scripts/storage** faça as seguintes alterações:

scripts/storage/aceitouSalvar.js

```
-export default JSON.parse(localStorage.getItem('aceitouSalvar'))
+export default JSON.parse(localStorage.getItem('aceitouSalvar'))
+
+export function setAceitouSalvar(aceitouSalvar) {
+  localStorage.setItem("aceitouSalvar", aceitouSalvar)
+}
```

2. No arquivo **paginaInicial.js** na pasta **scripts/storage** faça as seguintes alterações:

scripts/storage/paginaInicial.js

```
-export default localStorage.getItem('paginaInicial')
+export default localStorage.getItem('paginaInicial')
+
+export function setPaginaInicial(valor) {
+  localStorage.setItem('paginaInicial', valor)
+}
```

3. No arquivo **config.html** na pasta **raíz do projeto** faça as seguintes alterações:

config.html

```
<link rel="stylesheet" href="styles/cake-config-33d6f649ee.css"/>
```

```

<h1>Configurações</h1>

<label>
  Página inicial:
  <input type="text" id="$inputPaginaInicial">
</label>

<label>
  Permissão para armazenamento:
  <input type="checkbox" id="$inputPermitiuSalvar">
</label>

- <button>
+ <button id="$botaoSalvar">
  Salvar configurações
</button>
<a href="/">Voltar</a>

<script type="module" src="scripts/paginaConfiguracao.js"></script>

```

4. No arquivo **paginaConfiguracao.js** na pasta **scripts** faça as seguintes alterações:

scripts/paginaConfiguracao.js

```

- import paginaInicial from './scripts/storage/paginaInicial.js'
- import aceitouSalvar from './scripts/storage/aceitouSalvar.js'
+ import paginaInicial, { setPaginaInicial } from './scripts/storage/paginaInicial.js'
+ import aceitouSalvar, { setAceitouSalvar } from './scripts/storage/aceitouSalvar.js'

$inputPaginaInicial.value = paginaInicial
$inputPermitiuSalvar.checked = aceitouSalvar
+
+
+// o que vai ser executado quando clicar
+// o que vai ser executado quando o evento de click acontecer
+$botaoSalvar.onclick = salvar
+
+// função de callback
+// hoisting
+// função é um tipo de dado
+// executada em um outro momento do tempo
+function salvar(){
+  setAceitouSalvar($inputPermitiuSalvar.checked)
+  setPaginaInicial($inputPaginaInicial.value)
+}

```

5. No arquivo **pedeAceitouSalvar.js** na pasta **scripts/pedeInfosIniciais** faça as seguintes alterações:

scripts/pedeInfosIniciais/pedeAceitouSalvar.js

```

- import aceitouSalvar from './scripts/storage/aceitouSalvar.js'
+ import aceitouSalvar, { setAceitouSalvar } from './scripts/storage/aceitouSalvar.js'

if(aceitouSalvar === null){
  // shadowing/sombra no módulo
  // redeclarando com o mesmo nome
  const aceitouSalvar = confirm('Você aceita que a gente salve suas informações?')

  if(!aceitouSalvar) {

```

```

        alert('Você pode mudar isso na página de configurações')
    }

-   localStorage.setItem("aceitouSalvar", aceitouSalvar)
+   setAceitouSalvar(aceitouSalvar)
}

```

6. No arquivo **pedePaginaInicial.js** na pasta **scripts/pedeInfosIniciais** faça as seguintes alterações:

scripts/pedeInfosIniciais/pedePaginaInicial.js

```

import aceitouSalvar from './scripts/storage/aceitouSalvar.js'
- import paginaInicial from './scripts/storage/paginaInicial.js'
+ import paginaInicial, {setPaginaInicial} from './scripts/storage/paginaInicial.js'

if(aceitouSalvar === null || aceitouSalvar === true){
    // Sem shadowing
    let paginaInicialDefault = paginaInicial

    if(!paginaInicialDefault) {
        paginaInicialDefault = prompt("Escolha a página inicial")
    }

    if(paginaInicialDefault) {
        if (
            paginaInicialDefault.substring(0, 7) !== 'http://' &&
            paginaInicialDefault.substring(0,8) !== 'https://'
        ) {
            // Assignment Atribuição
            paginaInicialDefault = 'http://' + paginaInicialDefault
        }

        $janelaPrincipal.src = paginaInicialDefault
        $inputEndereco.value = paginaInicialDefault

-       localStorage.setItem('paginaInicial', paginaInicialDefault)
+       setPaginaInicial(paginaInicialDefault)
    }
}

```

7. Para que não haja mais o carregamento da página *blank.html* automaticamente na inicialização do navegador, removeremos alguns atributos na página principal **index.html**, que está na pasta **raíz do projeto**:

index.html

```

<meta charset="utf-8">

<link rel="stylesheet" href="styles/cake-2afdf04e92.css"/>

<header>
    <input type="image" src="images/libCake/recarregar.svg" onclick="window.location.reload()">
-   <input type="text" id="$inputEndereco" value="blank">
+   <input type="text" id="$inputEndereco">
    <a href="config.html">
        
    </a>
</header>

```

```
-<iframe src="blank" frameborder="0" id="$janelaPrincipal"></iframe>-  
+<iframe frameborder="0" id="$janelaPrincipal"></iframe>  
  
<script src="scripts/cake-8709f3abc4.js"></script>  
  
<script type="module" src="scripts/main.js"></script>
```

EXERCÍCIO: FUNÇÕES: CALLBACKS, ENCAPSULAMENTO E CLOSURES

12.1 COMENTÁRIOS

TODO

12.2 PASSO A PASSO COM CÓDIGO

1. No arquivo **aceitouSalvar.js** na pasta **scripts/storage** faça as seguintes alterações:

scripts/storage/aceitouSalvar.js

```
-export default JSON.parse(localStorage.getItem('aceitouSalvar'))
+// Encapsulamento
+export let aceitouSalvar = JSON.parse(localStorage.getItem('aceitouSalvar')) //null

-export function setAceitouSalvar(aceitouSalvar) {
-   localStorage.setItem("aceitouSalvar", aceitouSalvar)
+
+// Ambiente onde ela foi criada
+// Acesso a variaveis do ambiente
+// function setAceitouSalvar é uma "Closure"
+export function setAceitouSalvar(valor) {
+   aceitouSalvar = valor
+   localStorage.setItem("aceitouSalvar", valor)
+ }
```

2. No arquivo **paginaInicial.js** na pasta **scripts/storage** faça as seguintes alterações:

scripts/storage/paginaInicial.js

```
-export default localStorage.getItem('paginaInicial')
+export let paginaInicial = localStorage.getItem('paginaInicial')

+export function setPaginaInicial(valor) {
+   paginaInicial = valor
+   localStorage.setItem('paginaInicial', valor)
+ }
```

3. No arquivo **paginaConfiguracao.js** na pasta **scripts** faça as seguintes alterações:

scripts/paginaConfiguracao.js

```
-import paginaInicial, { setPaginaInicial } from './scripts/storage/paginaInicial.js'
```



```

-import aceitarSalvar, { setAceitouSalvar } from './scripts/storage/aceitouSalvar.js'
+import * as storagePaginaInicial from './scripts/storage/paginaInicial.js'
+import * as storageAceitouSalvar from './scripts/storage/aceitouSalvar.js'

-$inputPaginaInicial.value = paginaInicial
-$inputPermitiuSalvar.checked = aceitarSalvar
+$inputPaginaInicial.value = storagePaginaInicial.paginaInicial
+$inputPermitiuSalvar.checked = storageAceitouSalvar.aceitouSalvar

// o que vai ser executado quando clicar
// o que vai ser executado quando o evento de click acontecer
$botaoSalvar.onclick = salvar

// função de callback
// hoisting
// função é um tipo de dado
// executada em um outro momento do tempo
function salvar(){
-   setAceitouSalvar($inputPermitiuSalvar.checked)
-   setPaginaInicial($inputPaginaInicial.value)
+   storageAceitouSalvar.setAceitouSalvar($inputPermitiuSalvar.checked)
+   storagePaginaInicial.setPaginaInicial($inputPaginaInicial.value)
}

```

4. No arquivo **pedeAceitouSalvar.js** na pasta **scripts/pedeInfosIniciais** faça as seguintes alterações:

scripts/pedeInfosIniciais/pedeAceitouSalvar.js

```

-import aceitarSalvar, {setAceitouSalvar} from './scripts/storage/aceitouSalvar.js'
+import * as storageAceitouSalvar from './scripts/storage/aceitouSalvar.js'

-if(aceitouSalvar === null){
+if(storageAceitouSalvar.aceitouSalvar === null){
    // shadowing/sombra no módulo
    // redeclarando com o mesmo nome
    const aceitarSalvar = confirm('Você aceita que a gente salve suas informações?')

    if(!aceitouSalvar) {
        alert('Você pode mudar isso na página de configurações')
    }

-   setAceitouSalvar(aceitouSalvar)
+   storageAceitouSalvar.setAceitouSalvar(aceitouSalvar)
}

```

5. No arquivo **pedePaginaInicial.js** na pasta **scripts/pedeInfosIniciais** faça as seguintes alterações:

scripts/pedeInfosIniciais/pedePaginaInicial.js

```

-import aceitarSalvar from './scripts/storage/aceitouSalvar.js'
-import paginaInicial, {setPaginaInicial} from './scripts/storage/paginaInicial.js'
+import { aceitarSalvar as storageAceitouSalvar } from './scripts/storage/aceitouSalvar.js'
+import { paginaInicial, setPaginaInicial } from './scripts/storage/paginaInicial.js'

-if(aceitouSalvar === null || aceitarSalvar === true){
+if(storageAceitouSalvar === null || storageAceitouSalvar === true){

```

```

// Sem shadowing
let paginaInicialDefault = paginaInicial

if(!paginaInicialDefault) {
    paginaInicialDefault = prompt("Escolha a página inicial")
}

if(paginaInicialDefault) {
    if (
        paginaInicialDefault.substring(0, 7) !== 'http://' &&
        paginaInicialDefault.substring(0,8) !== 'https://'
    ) {
        // Assignement Atribuição
        paginaInicialDefault = 'http://' + paginaInicialDefault
    }

    $janelaPrincipal.src = paginaInicialDefault
    $inputEndereco.value = paginaInicialDefault

    setPaginaInicial(paginaInicialDefault)
}
}

```