

LOCAL e SESSION STORAGE

localStorage - salva os dados **sem data de expiração**. Os dados não serão removidos quando o *browser* for fechado, ou seja, eles ficarão disponíveis enquanto não forem **explicitamente** removidos:

criar:

1) `localStorage.setItem("firstName", "Adam")` //salva uma entrada com a key = "firstName" e value = "Adam"

2) `localStorage.keyName = "keyValue"` // cria uma variável também.

selecionar:

`let nome = localStorage.getItem("firstName");`

`alert(nome)` // vai alertar o valor Adam.

excluir:

1) `localStorage.removeItem("firstName" ou nome)` // remove Adam.

2) `delete localStorage.keyName` // deleta a chave keyName

limpar TUDO:

`localStorage.clear()` // Apaga TODAS as chaves criadas .

`console.log(localStorage.length)` //

sessionStorage - salva os dados apenas para a sessão atual. Os dados são removidos assim que a pessoa fecha a aba (*tab*) ou o *browser*.

- `sessionStorage.setItem("firstName", "Adam")` //salva uma entrada com a key = "firstName" e value = "Adam"

- `let nome = sessionStorage.getItem("firstName");`

`alert(nome)` // vai alertar o valor Adam.

- `sessionStorage.removeItem("firstName" ou nome) // remove Adam.`
- `sessionStorage.clear() // Apaga TODAS as chaves criadas anteriormente.`

É possível salvar outras estruturas em `localStorage` e `sessionStorage`:

```
let organization = {  
  name: "trybe",  
  since: 2019  
}
```

`// object "storage" can be localStorage or sessionStorage`

```
storage.setItem("trybe", JSON.stringify(organization))  
let org = JSON.parse(storage.getItem("trybe"))  
console.log(org) // { name: "trybe", since: 2019 }
```

```
let classes = ["2019/set", "2019/oct"]  
storage.setItem("classes", JSON.stringify(classes))  
let cls = JSON.parse(storage.getItem("classes"))  
console.log(cls) // ["2019/set", "2019/oct"]
```