

MAIS JAVASCRIPT

Factory Function

Factory Function

São funções que retornam um objeto sem a necessidade de utilizarmos a palavra chave new. Possuem basicamente a mesma função que constructor functions / classes.

```
function createButton(text) {  
  function element() {  
    const buttonElement = document.createElement('button');  
    buttonElement.innerText = text;  
    return buttonElement;  
  }  
  return {  
    element: element,  
    text: text,  
  }  
}  
  
const comprarBtn = createButton('Comprar');
```

Métodos / Variáveis privadas

Uma das vantagens é a possibilidade de criarmos métodos / variáveis privadas.

```
function criarPessoa(nome, sobrenome) {  
  const nomeCompleto = `${nome} ${sobrenome}`;  
  
  function andar() {  
    return `${nomeCompleto} andou`;  
  }  
  
  function nadar() {  
    return `${nomeCompleto} nadou`;  
  }  
  
  return {  
    nome,  
    sobrenome,  
    andar,  
    nadar,  
  }  
}  
  
const designer = criarPessoa('André', 'Rafael');
```

Ice Factory

Podemos impedir que os métodos e propriedades sejam modificados com `Object.freeze()`. Ideia inicial de Douglas Crockford.

```
'use strict';

function criarPessoa(nome, sobrenome) {
  const nomeCompleto = `${nome} ${sobrenome}`;
  function andar() {
    return `${nomeCompleto} andou`;
  }
  return Object.freeze({
    nome,
    sobrenome,
    andar,
  });
}

const designer = criarPessoa('André', 'Rafael');
```

Constructor Function / Factory Function

Uma das vantagens da Factory Function é a possibilidade de iniciarmos a mesma sem a utilização da palavra chave new. Também é possível fazer isso com uma Constructor Function.

```
function Pessoa(nome) {  
  if (!(this instanceof Pessoa)) // ou (!new.target)  
    return new Pessoa(nome);  
  this.nome = nome;  
}  
  
Pessoa.prototype.andar = function() {  
  return `${this.nome} andou`;  
}  
  
const designer = Pessoa('André');
```

Exemplo Real

```
function $$ (selectedElements) {  
  const elements = document.querySelectorAll(selectedElements);  
  
  function on (onEvent, callback) {  
    elements.forEach(element => {  
      element.addEventListener(onEvent, callback);  
    });  
    return this; // retornar this irá funcionar da mesma forma  
  }  
  
  function hide () {  
    elements.forEach(element => {  
      element.style.display = 'none';  
    });  
    return this;  
  }  
  
  function show () {  
    elements.forEach(element => {  
      element.style.display = 'initial';  
    });  
  }  
}
```

```
}

function addClass(className) {
  elements.forEach(element => {
    element.classList.add(className);
  });
  return this;
}

function removeClass(className) {
  elements.forEach(element => {
    element.classList.add(className);
  });
  return this;
}

return Object.freeze({
  elements,
  on,
  hide,
  show,
  addClass,
  removeClass,
```

```
const buttons = $$('button');  
buttons.hide().show().addClass('ativo').removeClass('ativo');
```