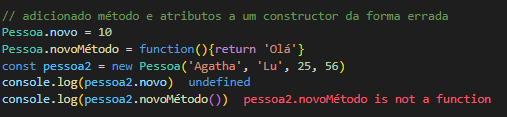
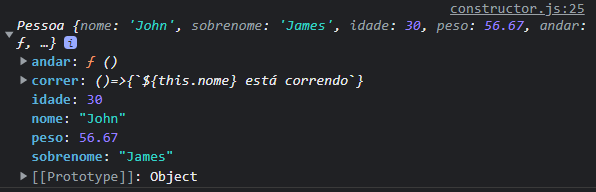
**Prototypes**

**Você não pode adicionar novos atributos e métodos a um constructor da forma abaixo**

**Ex.:** perceba que, mesmo adicionando um atributo e método ao construtor antes de criar o objeto, o novo atributo e método ficam, respectivamente, indefinidos e não-existente.

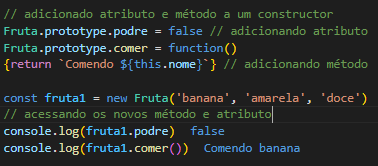
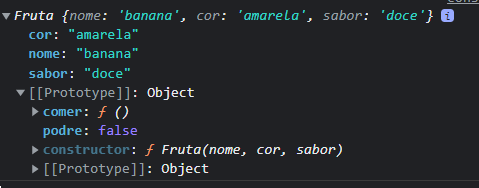


**Todo objeto vai herdar do objeto prototype, ou seja, todo objeto terá o objeto prototype**



**Como o prototype, é possível adicionar atributos e métodos a um constructor**

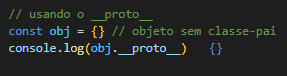
Eles serão adiconados ao objeto prototype de cada objeto do constructor, como uma forma de dizer o que foi mudado depois da criação doconstructor

**Ex.: perceba que o atributo podre e o método comer() não estão dentro do objeto Fruta em si, mas dentro do objeto prototype, que por sua vez está dentro de Fruta.**

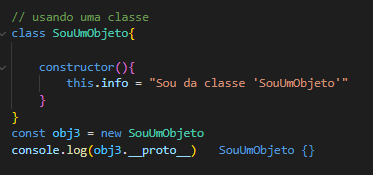
**Propriedade \_\_proto\_\_**

É uma propriedade que todo objeto tem e refere-se ao construtor daquele objeto.

**Ex.1:** objetos literais não têm construtores

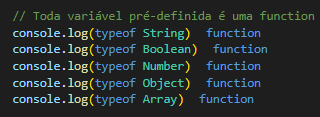


**Ex. 2:** estamos nos referindo à classe que criou o objeto



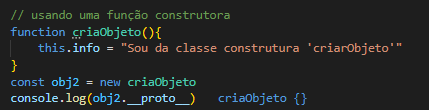
**Propriedade prototype**

É uma propriedade que toda variável pré-definida do JS tem:

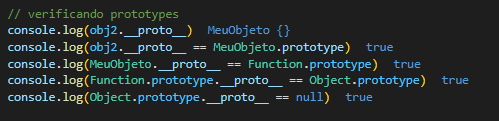
**Ex.:**

**Criando um objeto a partir da classe construtora**

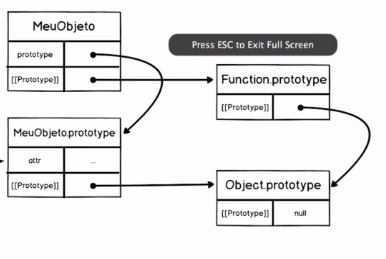
**Ex. 1:** estamos nos referindo à função construtura do objeto ‘obj2’



**Ex. 2:**

* O protótipo do **obj2** aponta para função **MeuObjeto**, ou seja, ele aponta para o **atributo prototype** dela mesmo
* O protótipo da **função MeuObjeto** aponta para o **atributo prototype de Function**
* O protótipo de **Function** aponta para o **atributo prototype de Object**
* O **protótipo de Object.prototype aponta para null**

**Esquema dos Protótipos**



**Criando métodos ou atributos usando o prototype**