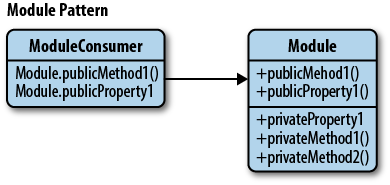
Pesquise sobre padrões de projeto e escolha um para apresentar e descrever o seu funcionamento. Além disso, explique quais as vantagens e desvantagens comparados a outros e mostre suas referências.

**Realize essa atividade no WORD ou no Bloco de Notas, suba esse arquivo para algum repositório e compartilhe o link no campo ao lado para que outros desenvolvedores possam analisá-lo.**

**Module pattern**

Em JavaScript, o Module patterné usado para emular ainda mais o conceito de classes de tal forma que podemos incluir métodos e variáveis públicos/privados dentro de um único objeto, protegendo assim partes específicas do escopo global. Isso resulta em uma redução na probabilidade de nossos nomes de função entrarem em conflito com outras funções definidas em scripts adicionais na página.



O Module patternencapsula o estado e a organização de “privacidade” usando closures. Ele fornece uma maneira de agrupar uma mistura de métodos e variáveis ​​públicos e privados, protegendo partes de vazar no escopo global e colidir acidentalmente com a interface de outro desenvolvedor. Com esse padrão, apenas uma API pública é retornada, mantendo todo o resto dentro do encerramento privado.

Isso nos dá uma solução limpa para blindar a lógica fazendo o trabalho pesado enquanto expõe apenas uma interface que gostaríamos que outras partes do nosso aplicativo usassem. O padrão é bastante semelhante a uma expressão funcional imediatamente invocada, exceto que um objeto é retornado em vez de uma função.

Deve-se notar que não há um sentido explicitamente verdadeiro de “privacidade” dentro do JavaScript porque, ao contrário de algumas linguagens tradicionais, ele não possui modificadores de acesso. As variáveis ​​não podem tecnicamente ser declaradas como públicas nem privadas, então usamos o escopo da função para simular este conceito. Dentro do padrão Module, as variáveis ​​ou métodos declarados só estão disponíveis dentro do próprio módulo, graças ao encerramento. Variáveis ​​ou métodos definidos dentro do objeto de retorno, no entanto, estão disponíveis para todos.

Use o Module Pattern para os seguintes benefícios:

* Manutenibilidade: Os padrões de módulo permitem uma melhor manutenibilidade, pois todo o código relacionado pode ser encapsulado dentro de um único bloco lógico. Esses blocos logicamente independentes são relativamente mais fáceis de atualizar.
* Reutilização: uma única unidade de código pode ser reutilizada em todo o aplicativo. A funcionalidade incluída como um módulo pode ser reutilizada e não precisamos definir as mesmas funções em vários pontos

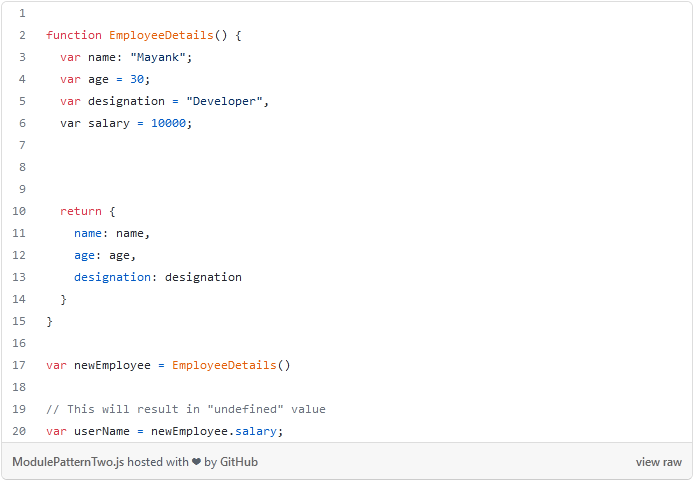
##### As desvantagens de se usar o Module Pattern são:

* É um pouco desgastante mudar a privacidade de um objeto
* Não é possível realizar testes unitários em objetos privados

Implementação do Module Pattern com apenas campos públicos:



Vamos agora expandir o código acima para definir algumas variáveis que não são acessíveis fora do escopo da Função do Módulo.



Agora vamos ver como podemos usar o valor do salário sem precisar expô-lo publicamente:



**Referência Bibliográfica**

KOBELLARZ, Jordan. Guia rápido de padrões de projeto Javascript. Jordan Kobellarz, 30 jan. 2015. Disponível em: <http://jordankobellarz.github.io/javascript/2015/01/30/guia-rapido-de-padroes-de-projeto-javascript.html>. Acesso em: 28 jul. 2022.

OSMANI, Addy. *Learning JavaScript Design Patterns by Addy Osmani*. Disponível em: <https://www.oreilly.com/library/view/learning-javascript-design/9781449334840/ch09s02.html>. Acesso em: 28 jul. 2022.

# GUPTA, Mayank. *Module Pattern in JavaScript*. Disponível em: <https://javascript.plainenglish.io/data-hiding-with-javascript-module-pattern-62b71520bddd>. Acesso em: 28 jul. 2022.