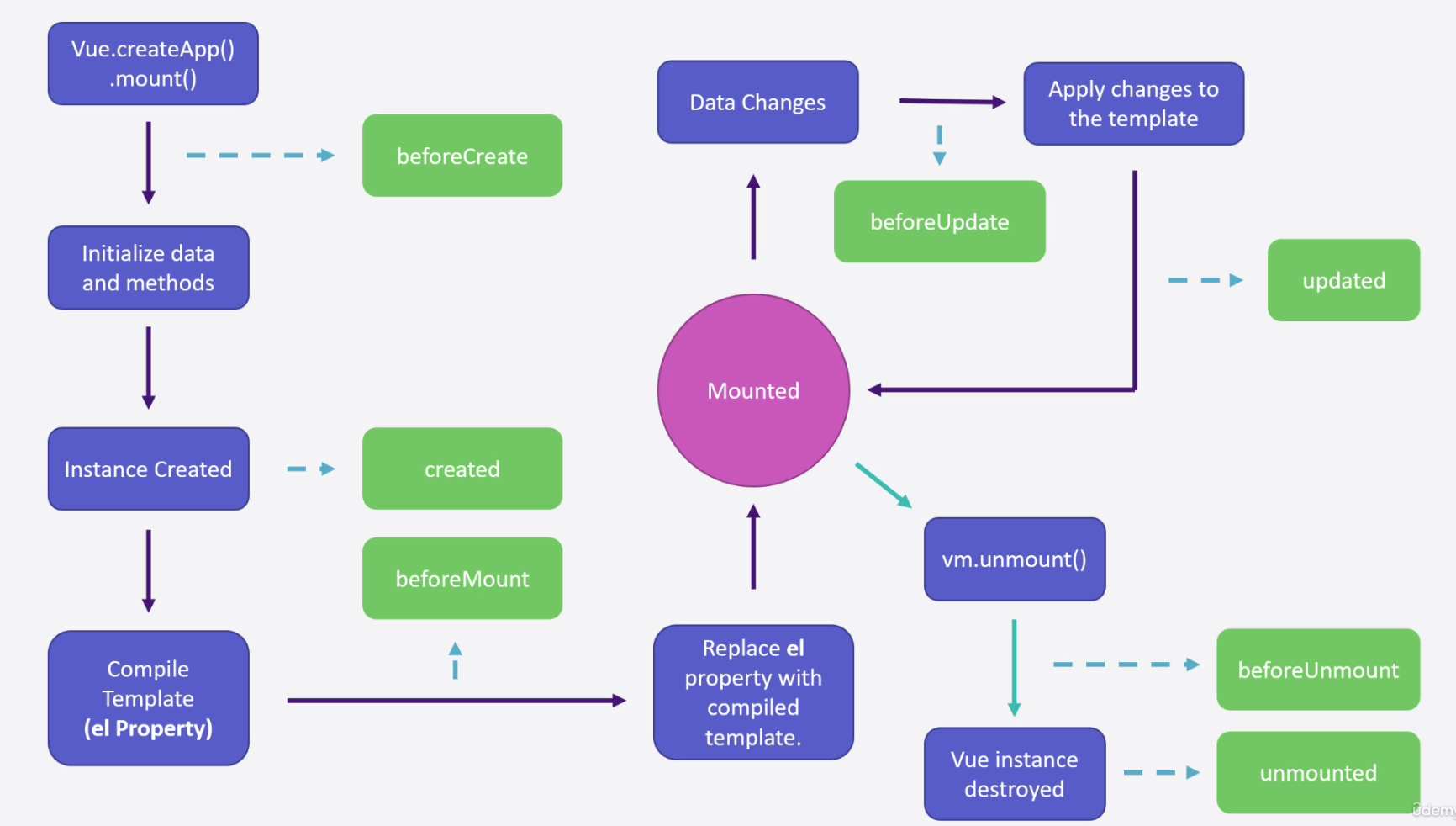
**Understanding Lifecycle Hooks**

****

**DIFFERENCE BETWEEN CREATED AND MOUNTED**

The mounted lifecycle is only triggered once the template is compiled and inserted into the document. We’ll have access to the template if we need to make changes to it. If you need the template rendered on the document, you need to use the mounted lifecycle. If you only need the data, then you can use the created lifecycle.

**BEFOREUPDATE**

We don’t have access to which property was updated. This hook will get triggered upon any changes to the data.

**VUE COMPILING THE TEMPLATE**

Utilizando como exemplo o elemento html a seguir:

<h1 class=”blue”> Hello World </h1>

O Vue irá compilar o seguinte:

{

tag: ‘h1’,

attributes: {

class: “blue”

},

content: “Hello World”

}



**PATCHING é o nome dado ao processo de atualização, adição ou remoção de elementos da página, isso inclui os atributos ou conteúdo de um elemento.**

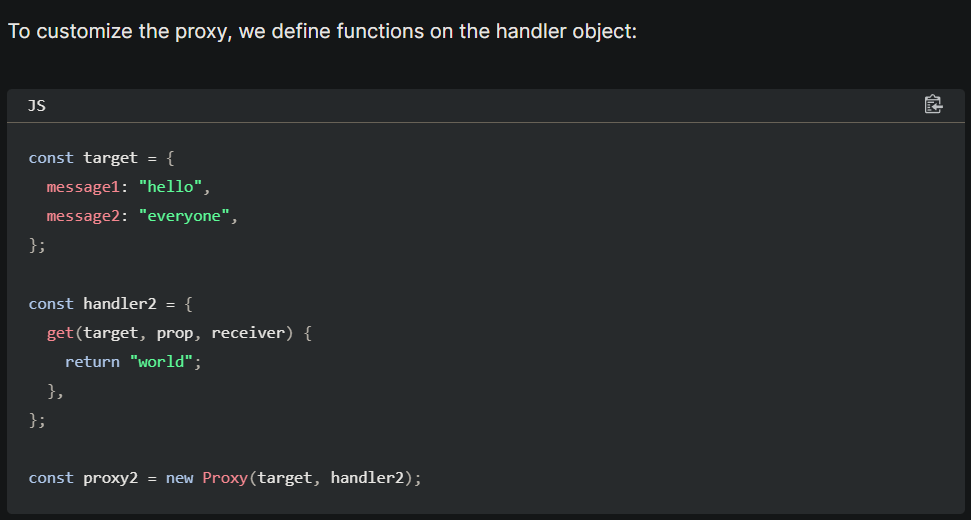
**PROXY**

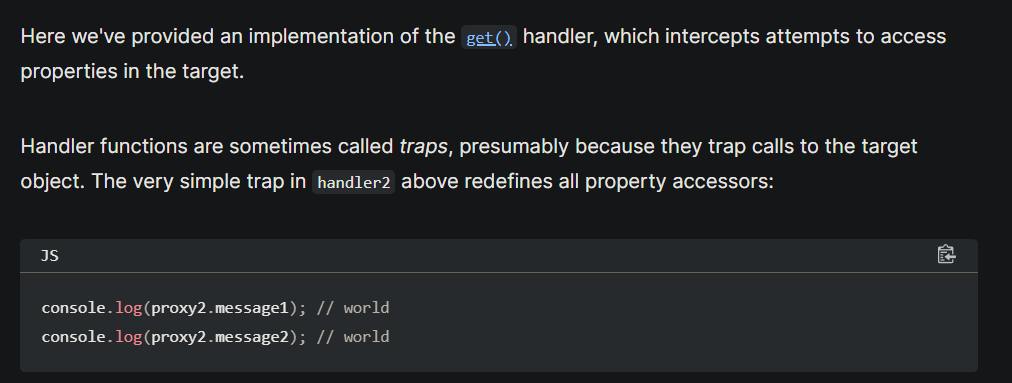
O Proxy permite você criar um objeto que pode ser usado no lugar do objeto original, ou redefinir operações como get, set e propriedades. São comumente utilizados para acessar propriedades log, validar e formatar inputs.

Possui dois parâmetros:

* target: objeto original que você quer procurar
* handler: operações que serão interceptadas e como redefinir as operações interceptadas.







**TEMPLATES**

Eles podem ser criados de duas maneiras: pela instância do Vue ou pelo documento html.

Não é recomendado pelo HTML por duas razões: a primeira é que muitos editores não conseguem pegar sintaxe HTML dentro do JavaScript. A segunda é que você não pode fazer facilmente templates de várias linhas.

**VUE COMPILER**

Pode-se utilizar o Vue com compilador ou sem ele. A principal vantagem em usar o Vue sem o compilador é a redução de tamanho do arquivo. Sem o compilador, o Vue consegue reduzir 30% do tamanho original.

Ao usar a versão sem o compilador, você continua tendo que acessar as funcionalidades como diretivas, expressões, bindings e eventos.

**COMPONENTS**

São criados após instanciar a Vue.createApp(), e antes de montá-lo.

vm.component(‘hello’, {

})

Nos componentes, o primeiro parâmetro sempre é o nome dele e o segundo é a configuração a ser feita no componente. Quando for chamar no html o componente criado, o seu nome é exatamente o que foi atribuído lá no JS como primeiro parâmetro.