

# Profissão: Engenheiro Front-End



# Orientação a Objetos com JavaScript

## GUIA DA AULA 6



# Encapsulamento



**Sobre  
encapsulamento**



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



## Sobre encapsulamento

O encapsulamento no permite ter maior controle sobre os atributos de um objeto, podem até mesmo validar os valores que o atributo irá receber.

Por padrão os atributos que utilizam `this.NOME_ATRIBUTO` são públicos, podem ser atribuídos ou lidos de qualquer maneira.



Para aplicar um controle de acesso às propriedades precisamos mudá-las para variáveis e criar métodos que alterem e accessem esta variável:

```
function Funcionario() {  
  let _salario = 0;  
  this.retornaSalario = function() {  
    return _salario;  
  }  
  this.atribuiSalario = function(valor) {  
    _salario = valor;  
  }  
}  
const funcionarioA = new Funcionario();
```



Acessamos o atributo e fazemos a atribuição de valor através de métodos:

```
funcionarioA.atribuiSalario(1000)
```

```
funcionarioA.retornaSalario() // 1000
```

O `_` antes do nome da variável é uma convenção utilizada para quando o atributo é privado. Dessa maneira podemos até adicionar uma validação na atribuição do valor, verificando se o número é positivo e se o tipo do dado é **number**.



Os métodos quando possuem a tarefa de retornar e atribuir valor a um atributo, são chamados de **getters** e **setters**, respectivamente. Dessa maneira o nome correto para os método seria:

**getSalario** e **setSalario**

Também podemos formatar o retorno do atributo:

```
getSalario() {  
  if (_salario === undefined) return "O salário ainda não foi  
atribuído";  
  return `O salário é de R$ ${_salario}`;  
}
```

