# Programação Funcional

Objetivo:

* Apresentação do paradigma;
* Conceitos fundamentais;
* JavaScript.

Por que JavaScript como linguagem para este curso?

* Linguagem da WEB;
* Multiparadigma;
* Foco no entendimento do Paradigma.

O que preciso para acompanhar esse curso?

* Conceitos básicos de programação;
* Linguagem JavaScript;
* Papel e caneta para anotar os conceitos;
* Editor de textos (Escrever o código);
* Navegador (Executar o código).

Por que programação funcional

* Organizar uma linha de raciocínio;
* Entender uma nova maneira de abordar os problemas;
* Algumas tecnologias têm como requisito.

Programação Funcional

É um paradigma de programação como a Orientação a Objetos, por exemplo.

* Uma maneira de resolver problemas.

Funções

* Pequenas tarefas dentro de uma função;
* Abstrair um problema e isolar ele dentro da função;
* Não modificar dados fora dela;
* Pequenas e bem especificas no que fazem.

Características principais da função

* Um dado (ou mais) entra em uma função e um novo dado sai dessa função;
* Não altera dados;
* Não guarda estado stateless.

Linguagens funcionais

* JavaScript (multiparadigma);
* PHP (multiparadigma);
* Elixir;
* Haskell.

Pontos Positivos

* Fácil manutenção;
* Fácil uso de testes;
  + Funções isoladas e consistentes;
* Códigos mais confiáveis;

Princípios

# PARADIGMAS

* Programação Imperativa;
* Programação Declarativa;

# DADOS

* Imutabilidade;
* Stateless;

# FUNÇÕES

* Independentes;
* Puras;
* Higher-order;
* First-class;
* Composição.

Programação Imperativa

* O código é pensado e gerado em sequência;
* O código é pensado como um passo-a-passo, com uma receita de bolo;
* Uma coisa depende da outra;
* O estado de um dado é alterado constantemente causando mutações nas variáveis;
* Orientações a objetos é um tipo de paradigma imperativo.

Programação Declarativa

* O código é gerado para fazer algo, não importa como;
* O que fazer e não **como fazer**;
* Não há necessidade de um passo-a-passo no código;
* Programação funcional é um tipo de paradigma declarativo.

Imutabilidade

* Uma variável não vai variar;
* Se você mudar uma variável, você não muda, você cria uma nova.

Stateless

* Não guarda estado;
* A função só conhece dados entregues a ela;
* A resposta não poderá variar.

## Funções

Vamos conhecer algumas características das funções na programação funcional.

* Funções independentes;
* Funções puras;
* First-class functions;
* Higher-order functions;
* Composição de funções.

Funções independentes

* Deverá ter ao menos 1 argumento;
* Deverá retornar algo;
* Nada que acontecer lá dentro afeta o mundo externo;
  + Dados imutáveis;
  + Não guardar estado.
* Não faremos o uso de loops (for, while), mas se houver necessidade de tal, usaremos recursão (a função chama ela mesma).

Funções puras

* Não devera depender de nenhum dado externo a não ser o que foi passado como argumento;
* Não deverá sofrer nenhuma interferência do mundo externo a ela;
* Se o argumento é o mesmo, o retorno não poderá ser diferente;
* Não terá nenhum efeito colateral no seu código;
  + Não irá modificar nenhum dado.
  + Não irá guardar nenhum dado.

First-class function

* Podem estar em qualquer lugar, inclusive, como parâmetro de outras funções;
* A função poderá ser entendida como uma variável.

Higher-order function

* Funções que recebem funções como argumentos;
* Funções que poderão retornar outras funções.

Composição de funções

* Um encadeamento de funções;
* Uma função que retorna um dado e vai para outra função, que retorna um dado e vai para outra função, que retorna….

Conclusão

* Paradigma de programação;
* Linguagens;
* Conceitos fundamentais;
  + Stateless, Imutabilidade, Função pura, Higher-orders.