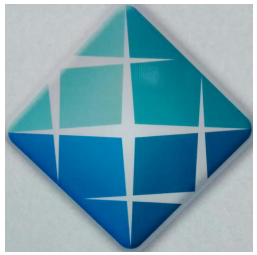
Paradigmas de Linguagens de Programação em Python

Aula 01

Evandro J.R. Silva[1]

[1] Bacharelado em Ciência da Computação
Estácio Teresina



Sumário

01
Razões para estudar conceitos de linguagens de programação

Razões para estudar conceitos de linguagens de programação

- Aumento da capacidade de expressar ideias
  - O pouco conhecimento sobre tipos de estruturas de controle, estruturas de dados e abstrações terá como consequência limitações no desenvolvimento de software.
  - Um maior conhecimento, por outro lado, permite o aumento da diversidade dos seus processos mentais, ou seja, a maneira como você vai pensando na soluções possíveis para diversos desafios.
  - Caso seja necessários, o programador pode trazer/simular recursos de outras linguagens para a que estiver em uso.

Razões para estudar conceitos de linguagens de programação

- Aumento da capacidade de expressar ideias
- Embasamento para escolher linguagens adequadas.

Razões para estudar conceitos de linguagens de programação

- Aumento da capacidade de expressar ideias

- Embasamento para escolher linguagens adequadas.
- Aumento da habilidade de aprender novas linguagens de programação.

Razões para estudar conceitos de linguagens de programação

- Aumento da capacidade de expressar ideias

- Embasamento para escolher linguagens adequadas.
- Aumento da habilidade de aprender novas linguagens de programação.
- Melhor entendimento dos detalhes de implementação
  - Entender o motivo pelo qual uma linguagem foi implementada de determinada forma, leva à habilidade de usá-la de forma mais inteligente, conforme a linguagem foi projetada para ser usada.

Razões para estudar conceitos de linguagens de programação

- Aumento da capacidade de expressar ideias

- Embasamento para escolher linguagens adequadas.
- Aumento da habilidade de aprender novas linguagens de programação.
- Melhor entendimento dos detalhes de implementação

- Melhor uso de linguagens já conhecidas.

02
Domínios de Programação

Domínios de Programação

- Computadores são utilizados para muitas tarefas
  - Controlar usinas nucleares;
  - Ajudar pilotos a controlar suas aeronaves;
  - Jogos;
  - Telefones celulares e aplicativos, etc.

Domínios de Programação

- Computadores são utilizados para muitas tarefas
  - Controlar usinas nucleares;
  - Ajudar pilotos a controlar suas aeronaves;
  - Jogos;
  - Telefones celulares e aplicativos, etc.
- Vejamos algumas aplicações e suas linguagens associadas.

Domínios de Programação

- Aplicações científicas
  - O foco está na utilização de estruturas de dados relativamente simples, porém com a exigência de diversos cálculos aritméticos de ponto flutuante.

Domínios de Programação

- Aplicações científicas
  - O foco está na utilização de estruturas de dados relativamente simples, porém com a exigência de diversos cálculos aritméticos de ponto flutuante.
  - As estruturas de dados mais comuns são vetores e matrizes, e as estruturas de controle mais comuns são os laços de contagem e as seleções.

Domínios de Programação

- Aplicações científicas
  - O foco está na utilização de estruturas de dados relativamente simples, porém com a exigência de diversos cálculos aritméticos de ponto flutuante.
  - As estruturas de dados mais comuns são vetores e matrizes, e as estruturas de controle mais comuns são os laços de contagem e as seleções.
  - A eficiência é uma preocupação primordial!

Domínios de Programação

- Aplicações científicas
  - O foco está na utilização de estruturas de dados relativamente simples, porém com a exigência de diversos cálculos aritméticos de ponto flutuante.
  - As estruturas de dados mais comuns são vetores e matrizes, e as estruturas de controle mais comuns são os laços de contagem e as seleções.
  - A eficiência é uma preocupação primordial!
  - **Fortran** foi a primeira linguagem para aplicações científicas, seguida do **ALGOL 60** e seus descendentes. Porém, nenhuma linguagem conseguiu ser mais eficiente que o Fortran (ainda hoje é usada).

Domínios de Programação

- Aplicações científicas
  - O foco está na utilização de estruturas de dados relativamente simples, porém com a exigência de diversos cálculos aritméticos de ponto flutuante.
  - As estruturas de dados mais comuns são vetores e matrizes, e as estruturas de controle mais comuns são os laços de contagem e as seleções.
  - A eficiência é uma preocupação primordial!
  - **Fortran** foi a primeira linguagem para aplicações científicas, seguida do **ALGOL 60** e seus descendentes. Porém, nenhuma linguagem conseguiu ser mais eficiente que o Fortran (ainda hoje é usada).
  - Outras linguagens de programação mais modernas são também amplamente utilizadas para computação científica, porém com menor foco em eficiência, pois a linguagem em si é de propósito geral: C, C++, Matlab, Julia.

Domínios de Programação

- Aplicações empresariais
  - Em 1960 surgiu a primeira linguagem de alto nível bem sucedida para negócios: COBOL.

Domínios de Programação

- Aplicações empresariais
  - Em 1960 surgiu a primeira linguagem de alto nível bem sucedida para negócios: COBOL.
  - Linguagens de negócios são caracterizadas por facilidades para a produção de relatórios elaborados, maneiras precisas de descrever e armazenar números decimais e caracteres, e a habilidade de especificar operações aritméticas decimais.

Domínios de Programação

- Aplicações empresariais
  - Em 1960 surgiu a primeira linguagem de alto nível bem sucedida para negócios: COBOL.
  - Linguagens de negócios são caracterizadas por facilidades para a produção de relatórios elaborados, maneiras precisas de descrever e armazenar números decimais e caracteres, e a habilidade de especificar operações aritméticas decimais.
  - Poucos avanços ocorreram nas linguagens para aplicações comerciais além do COBOL. Outras linguagens mais modernas, de propósito geral são utilizadas para esse propósito.

Domínios de Programação

- Inteligência Artificial
  - É uma ampla área de aplicações computacionais caracterizadas pelo uso de computações simbólicas em vez de numéricas (imagine a computação de fórmulas matemáticas contendo variados símbolos).

Domínios de Programação

- Inteligência Artificial
  - É uma ampla área de aplicações computacionais caracterizadas pelo uso de computações simbólicas em vez de numéricas (imagine a computação de fórmulas matemáticas contendo variados símbolos).
  - Em 1959 surgiu a primeira linguagem para aplicações de IA: **Lisp**. No início da década de 70 surgiu o **Prolog**.
  - Mais recentemente **Python** se tornou a linguagem mais utilizada para aplicações tanto de IA quanto de Aprendizado de Máquina. Outras linguagens são também bastante utilizadas: C, C++, Java; porém nenhuma delas teve aplicações de IA como seu foco.

Domínios de Programação

- Software para a Web
  - A Internet é mantida por uma coleção eclética de linguagens.
  - Linguagens de marcação: **HTML** e **XML**.

Domínios de Programação

- Software para a Web
  - A Internet é mantida por uma coleção eclética de linguagens.
  - Linguagens de marcação: **HTML** e **XML**.
  - Linguagem de estilo: **CSS**.

Domínios de Programação

- Software para a Web
  - A Internet é mantida por uma coleção eclética de linguagens.
  - Linguagens de marcação: **HTML** e **XML**.
  - Linguagem de estilo: **CSS**.
  - Linguagens de *scripting*: **JavaScript** e **PHP**.

Domínios de Programação

- Software para a Web
  - A Internet é mantida por uma coleção eclética de linguagens.
  - Linguagens de marcação: **HTML** e **XML**.
  - Linguagem de estilo: **CSS**.
  - Linguagens de *scripting*: **JavaScript** e **PHP**.
  - Linguagens de propósito geral: **Java**, **Ruby**, **Python**.

03
Critérios de avaliação de linguagens

Critérios de avaliação de linguagens

- É um tanto difícil e controverso classificar as linguagens de programação.
- Tão difícil quanto são dois cientistas da computação concordarem 100% com o valor de certas características das linguagens em relação às outras.

Critérios de avaliação de linguagens

- É um tanto difícil e controverso classificar as linguagens de programação.
- Tão difícil quanto são dois cientistas da computação concordarem 100% com o valor de certas características das linguagens em relação às outras.
- Veremos quatro critérios importantes: **legibilidade**, **facilidade de escrita**, **confiabilidade** e **custo**.

Critérios de avaliação de linguagens

- Legibilidade
  - É a facilidade com que os programas podem ser lidos e entendidos.

Critérios de avaliação de linguagens

- Legibilidade
  - É a facilidade com que os programas podem ser lidos e entendidos.
  - De início o foco era sobre a eficiência, que tinha por consequência o projeto de linguagens mais do ponto de vista do computador.

Critérios de avaliação de linguagens

- Legibilidade
  - É a facilidade com que os programas podem ser lidos e entendidos.
  - De início o foco era sobre a eficiência, que tinha por consequência o projeto de linguagens mais do ponto de vista do computador.
  - Nos anos 1970 o conceito de ciclo de vida de software foi desenvolvido, e a manutenção do software foi reconhecida como parte importante do ciclo.

Critérios de avaliação de linguagens

- Legibilidade
  - É a facilidade com que os programas podem ser lidos e entendidos.
  - De início o foco era sobre a eficiência, que tinha por consequência o projeto de linguagens mais do ponto de vista do computador.
  - Nos anos 1970 o conceito de ciclo de vida de software foi desenvolvido, e a manutenção do software foi reconhecida como parte importante do ciclo.
  - **Simplicidade geral**: a multiplicidade de recursos, ao contrário do que parece, não melhora muito a legibilidade, principalmente quando há uma quantidade exagerada de recursos:

```
count = count + 1
count += 1
count++
++count
```

    - **Sobrecarga de operadores**
      $1 + 1$
      $[1\,2\,3\,4\,5] + [6\,7\,8\,9\,10]$

Critérios de avaliação de linguagens

- Legibilidade
  - **Ortogonalidade**: Um conjunto relativamente pequeno de construções primitivas pode ser combinado a um número relativamente pequeno de formas para construir as estruturas de controle e de dados da linguagem.
    - A palavra *ortogonal* é proveniente do conceito matemático de vetores ortogonais, os quais são independentes entre si.

Critérios de avaliação de linguagens

- Legibilidade
  - **Ortogonalidade**: Um conjunto relativamente pequeno de construções primitivas pode ser combinado a um número relativamente pequeno de formas para construir as estruturas de controle e de dados da linguagem.
    - A palavra *ortogonal* é proveniente do conceito matemático de vetores ortogonais, os quais são independentes entre si.
    - O siginificado de um recurso de linguagem ortogonal é independente do contexto de sua aparição em um programa.
    - A falta de ortogonalidade leva a exceções às regras da linguagem.

Critérios de avaliação de linguagens

- Legibilidade
  - **Ortogonalidade**: Um conjunto relativamente pequeno de construções primitivas pode ser combinado a um número relativamente pequeno de formas para construir as estruturas de controle e de dados da linguagem.
    - A palavra *ortogonal* é proveniente do conceito matemático de vetores ortogonais, os quais são independentes entre si.
    - O siginificado de um recurso de linguagem ortogonal é independente do contexto de sua aparição em um programa.
    - A falta de ortogonalidade leva a exceções às regras da linguagem.
    - Ex.:
      Em C:
      `int func(int a)` → Passagem por valor
      ≠
      `int func(int a[])` → Passagem por referência

Critérios de avaliação de linguagens

- Legibilidade
  - **Ortogonalidade**: Um conjunto relativamente pequeno de construções primitivas pode ser combinado a um número relativamente pequeno de formas para construir as estruturas de controle e de dados da linguagem.
    - A palavra *ortogonal* é proveniente do conceito matemático de vetores ortogonais, os quais são independentes entre si.
    - O siginificado de um recurso de linguagem ortogonal é independente do contexto de sua aparição em um programa.
    - A falta de ortogonalidade leva a exceções às regras da linguagem.
    - Ex.:
      Em C:
      `int func(int a)` $\rightarrow$ Passagem por valor
      $\neq$
      `int func(int a[])` $\rightarrow$ Passagem por referência
    - Ao mesmo tempo, ortogonalidade demais pode prejudicar a legibilidade, devido a construções extremamente complexas.

Critérios de avaliação de linguagens

- Legibilidade
  - Tipos de dados
    - A falta de tipos de dados pode prejudicar a leitura de um código.
    - Ex.:
      Uma linguagem que não possui o tipo **bool**:
      `var = 1`
      Uma linguagem que possui o tipo **bool**:
      `var = true`

Critérios de avaliação de linguagens

- Legibilidade
  - Tipos de dados

  - Projeto da sintaxe
    - Escolha das palavras especiais/reservadas e símbolos: **while**, **class**, **if**, @, **#**, **{** e **}** etc.
    - Palavras especiais/reservadas podem ser utilizadas como nome de variáveis?

Critérios de avaliação de linguagens

- Facilidade de escrita
  - É a medida do quão facilmente uma linguagem pode ser usada para criar programas para um domínio.

Critérios de avaliação de linguagens

- Facilidade de escrita
  - É a medida do quão facilmente uma linguagem pode ser usada para criar programas para um domínio.
  - Não é justo comparar a facilidade de escrita de duas linguagens no contexto de determinada aplicação quando uma delas foi projetada para tal aplicação e a outra não.

Critérios de avaliação de linguagens

- Facilidade de escrita
  - Simplicidade e ortogonalidade

Critérios de avaliação de linguagens

- Facilidade de escrita
  - Simplicidade e ortogonalidade
    - Uma grande quantidade de construções faz com que vários programadores não tenham ciência de sua existência ou seu correto uso.

Critérios de avaliação de linguagens

- Facilidade de escrita
  - Simplicidade e ortogonalidade
    - Uma grande quantidade de construções faz com que vários programadores não tenham ciência de sua existência ou seu correto uso.
    - É possível, inclusive, a utilização acidental de recursos desconhecidos, gerando resultados inesperados.

Critérios de avaliação de linguagens

- Facilidade de escrita
  - Simplicidade e ortogonalidade
    - Uma grande quantidade de construções faz com que vários programadores não tenham ciência de sua existência ou seu correto uso.
    - É possível, inclusive, a utilização acidental de recursos desconhecidos, gerando resultados inesperados.
    - É melhor ter um número menor de construções primitivas e um conjunto de regras consistente para combiná-las (ortogonalidade).

Critérios de avaliação de linguagens

- Facilidade de escrita
  - Simplicidade e ortogonalidade
  - Expressividade
    - Em geral, uma linguagem expressiva especifica computações de uma forma conveniente. Ex.: `count++` é mais conveniente do que `count = count + 1`
      `idade > 18` **`and`** `altura > 1.5` mais conveniente que `idade > 18` **`&&`** `altura > 1.5`

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
    - É a execução de testes para detectar erros de tipos em programa, em tempo de compilação e em tempo de execução.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
    - É a execução de testes para detectar erros de tipos em programa, em tempo de compilação e em tempo de execução.
    - É uma tarefa mais fácil em linguagens de **tipagem estática**, como Java, e mais difícil em linguagens de **tipagem dinâmica**, como Python.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
    - É a habilidade de um programa interceptar erros em tempo de execução (além de outras condições não usuais detectáveis pelo programa), tomar medidas corretivas e continuar sua execução.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
    - São utilizados quando é possível ter um ou mais nomes em um programa para acessar a mesma célula de memória.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
    - São utilizados quando é possível ter um ou mais nomes em um programa para acessar a mesma célula de memória.
    - Em outras palavras: duas ou mais “variáveis” que na verdade **apontam** para o mesmo endereço de memória.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
    - São utilizados quando é possível ter um ou mais nomes em um programa para acessar a mesma célula de memória.
    - Em outras palavras: duas ou mais “variáveis” que na verdade **apontam** para o mesmo endereço de memória.
    - Essa possibilidade, e como o programador pode utilizá-la, depende da linguagem.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
    - São utilizados quando é possível ter um ou mais nomes em um programa para acessar a mesma célula de memória.
    - Em outras palavras: duas ou mais “variáveis” que na verdade **apontam** para o mesmo endereço de memória.
    - Essa possibilidade, e como o programador pode utilizá-la, depende da linguagem.
    - Por causa disso, é recomendável sempre ter o conhecimento de como isso funciona na linguagem que está sendo utilizada, pois se o valor de uma das variáveis é modificada, a outra também é. A falta desse conhecimento pode gerar erros difíceis de detectar.

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
  - Legibilidade e facilidade de escrita
    - Programas difíceis de ler são também difíceis de escrever e modificar (manutenção).

Critérios de avaliação de linguagens

- Confiabilidade
  - Um programa é confiável quando está de acordo com suas especificações em todas as condições.
  - Verificação de tipos
  - Tratamento de exceções
  - Apelidos
  - Legibilidade e facilidade de escrita
    - Programas difíceis de ler são também difíceis de escrever e modificar (manutenção).
    - Um programa escrito em uma linguagem que não suporta maneiras naturais de expressar algoritmos exigidos, necessariamente usará estratégias artificias.
    - É menos provável que estratégias artificias estejam corretar para todas as situações possíveis.
    - Quanto mais fácil é escrever um programa, maior a probabilidade dele estar correto.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem
    - Python: é fácil de aprender e de escrever, porém é bastante custoso de executar.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem
    - Python: é fácil de aprender e de escrever, porém é bastante custoso de executar.
    - Se uma aplicação exige bom desempenho, os programadores têm de fazer parte do programa em C ou C++.

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem

  - Custo do sistema de implementação da linguagem. Quando custa para desenvolver ou instalar uma aplicação feita em uma determinada linguagem?

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem

  - Custo do sistema de implementação da linguagem. Quando custa para desenvolver ou instalar uma aplicação feita em uma determinada linguagem?
    - Matlab é uma linguagem e sistema proprietário (pago).

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem

  - Custo do sistema de implementação da linguagem. Quando custa para desenvolver ou instalar uma aplicação feita em uma determinada linguagem?
    - Matlab é uma linguagem e sistema proprietário (pago).
    - Quais os detalhes da licença para a utilização de alguma linguagem? Existe alguma condição que resulte em cobrança?

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem

  - Custo do sistema de implementação da linguagem. Quando custa para desenvolver ou instalar uma aplicação feita em uma determinada linguagem?

  - Custo da baixa confiabilidade: e se o programa falhar em uma atividade crítica?

Critérios de avaliação de linguagens

- Custo
  - O curso total de uma linguagem é uma função de muitas de suas características.
  - O custo de treinar programadores em uma linguagem está associado à sua simplicidade e ortogonalidade.
  - O custo de escrever programas em uma linguagem está associado à sua facilidade de escrita.
  - Há ainda o custo de compilar programas em uma determinada linguagem.
  - Há também o custo de executar programas escritos em uma linguagem

  - Custo do sistema de implementação da linguagem. Quando custa para desenvolver ou instalar uma aplicação feita em uma determinada linguagem?

  - Custo da baixa confiabilidade: e se o programa falhar em uma atividade crítica?
  - Custo de manutenção.

Critérios de avaliação de linguagens

- Custo
  - Os três custos mais importantes: desenvolvimento de programas, manutenção e confiabilidade.

04
Categorias de linguagens

Categorias de linguagens

- **Imperativa**
  - **Procedural**
  - **Orientada a Objetos**
- **Declarativa**
  - **Funcional**
  - **Lógica**

Categorias de linguagens

- **Imperativa**
  - Descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa.
  - Um algoritmo é especificado em muitos detalhes, e deve ser incluída a ordem de execução específica das instruções.
  - **Procedural**
  - **Orientada a Objetos**
- **Declarativa**
  - **Funcional**
  - **Lógica**

Categorias de linguagens

- **Imperativa**
  - **Procedural**
    - O programa é construído a partir de um ou mais **procedimentos** (também chamados de *subrotinas* ou funções). Grande foco sobre a modularidade.
  - **Orientada a Objetos**
    - Paradigma baseado no conceito de objetos (classes), que podem ser dados, código, ou uma mistura de ambos.
    - Funções fazem parte de um objeto e passam a ser chamadas de métodos.
- **Declarativa**
  - **Funcional**
  - **Lógica**

Categorias de linguagens

- **Imperativa**
  - **Procedural**
  - **Orientada a Objetos**
- **Declarativa**
  - Inclui linguagens para a construção da estrutura e de elementos de um programa de forma a expressar sua lógica de computação, porém sem descrever o controle de fluxo.
  - **Funcional**
  - **Lógica**

Categorias de linguagens

- **Imperativa**
  - **Procedural**
  - **Orientada a Objetos**
- **Declarativa**
  - **Funcional**
    - Programas são construídos a partir da aplicação e composição de funções. As definições das funções são árvores de expressões que mapeia uns valores para outros valores (em vez de uma sequência de instruções imperativas).
  - **Lógica**
    - Conjunto de sentenças em forma lógica, expressando fatos e regras sobre algum domínio de problema.

Categorias de linguagens

- Exemplo de programa funcional (Elixir)

```
iex> [1, a] = [1, 2]
iex> a
2

iex> {:ok, [hello: a]} = {:ok, [hello: "world"]}
iex> a
"world"
```

Categorias de linguagens

- Exemplo de programa lógico (Prolog)

```
mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

sibling(X, Y) :- parent_child(Z, X), parent_child(Z, Y).

parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).


?- sibling(sally, erica).
Yes
```

Categorias de linguagens

- Multiparadigma
  - As linguagens mais modernas são multiparadigmas, ou seja, suportam vários dos diferentes paradigmas.
  - Ex.:
    - **Kotlin**: orientado a objetos, funcional, imperativo, bloco estruturado, declarativo, genérico, reflexivo, concorrente.
    - **PHP**: imperativo, orientado a objetos, procedural, reflexivo.
    - **Python**: orientado a objetos, procedural, funcional, estruturado, reflexivo.
    - **Go**: concorrente, imperativo, orientado a objetos.
    - **Rust**: concorrente, funcional, genérico, imperativo, estruturado.
    - **Scala**: concorrente, funcional, imperativo, orientado a objetos.
    - **Java**: genérico, orientado a objetos, funcional, imperativo, reflexivo, concorrente.
    - **Julia**: despacho múltiplo, funcional, procedural, metaprogramação, reflexivo, multiestágio, orientado a objetos.

05
Métodos de implementação

Métodos de implementação

- São quatro os tipos de implementação:
  1. Compilação;
  2. Interpretação;
  3. Híbrido.
  4. Pré-processador
     - Programa que processa outro imediatamente antes dele ser compilado.

Terminamos por hoje!

Aula baseada no livro:
SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. 11a Ed., Capítulo 1, Porto Alegre: Bookkman, 2018.