CEBRASPE | PC AL · AGENTE E ESCRIVÃO

REVISÃO DE VÉSPERA · PARTE 1

Java e Python

Tecnologia da Informação: Linguagem de programação
Sábado, 28/11 · 15h30 · questões CEBRASPE

ROTEIRO DA AULA

01 **Java: plataforma e memória**
JVM e bytecode | Passagem por valor | Garbage Collector

02 **Java: estruturas e classes**
Controle de fluxo | Modificadores de acesso | Classes e objetos

03 **Java: herança e interfaces**
Herança simples | Interfaces | Polimorfismo

04 **Python: sintaxe**
Indentação

05 **Python: coleções**
Lista | Tupla | Conjunto | Dicionário

destrinchandoTI
Prof. Rogério Araújo
@destrinchandoti

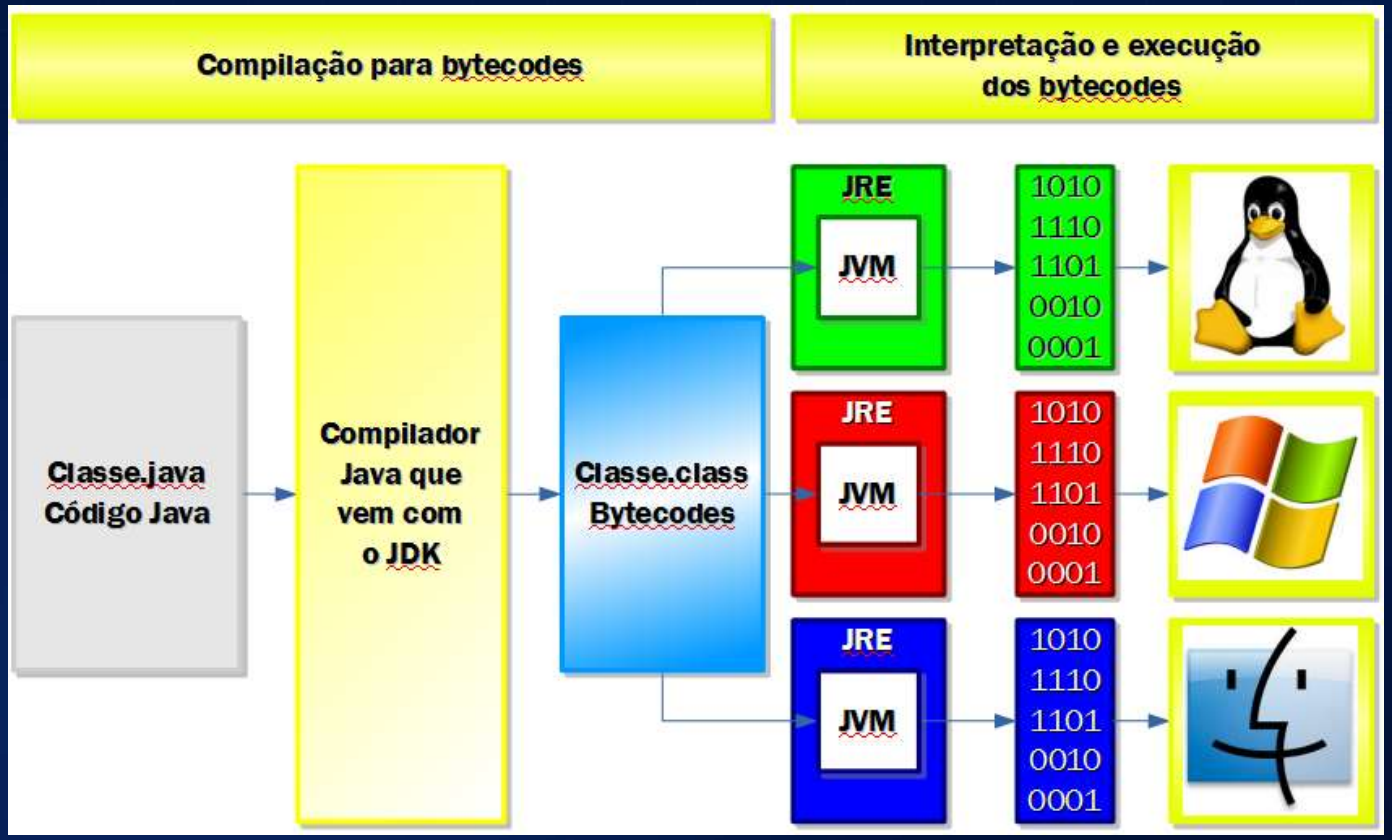
01

MÓDULO 01

Java

JVM e bytecode | Passagem por valor | Garbage Collector | Controle de fluxo | Acesso | Classes | Herança | Interfaces

Compilação, interpretação e execução



Compilação, interpretação e execução



Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Perito Criminal Federal – Área 3: Informática Forense] Julgue os próximos itens, a respeito das características e das aplicações básicas das linguagens Java e JavaScript.

- Devido à compilação do código-fonte em *bytecode*, que é executado pela máquina virtual Java (JVM), os programas Java podem ser executados em diversos sistemas operacionais, sem a necessidade de recompilação.

Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Perito Criminal Federal – Área 3: Informática Forense] Julgue os próximos itens, a respeito das características e das aplicações básicas das linguagens Java e JavaScript.

- Devido à compilação do código-fonte em *bytecode*, que é executado pela máquina virtual Java (JVM), os programas Java podem ser executados em diversos sistemas operacionais, sem a necessidade de recompilação.
  - Gabarito: CERTO.

Passagem de parâmetros em Java

Em Java, **toda passagem é por valor**. Com objetos, o que se copia é a **referência**, e não o objeto.

Sempre por valor

1 Tipo primitivo

int, double, boolean, char...

```
void dobrar(int x) {
    x = x * 2;
}
int a = 5;
dobrar(a);
// a continua 5
```

× Original não muda

O método recebe uma **cópia do valor**. Mexer na cópia não afeta a variável de quem chamou.

2 Objeto: altera o estado

Usa a cópia da referência

```
void depositar(Conta c) {
    c.saldo += 100;
}
Conta conta = new Conta();
depositar(conta);
// saldo agora é 100
```

✓ Objeto alterado

A cópia da referência aponta para o **mesmo objeto** no heap. Mudar seus atributos é visível fora do método.

3 Objeto: reatribui o parâmetro

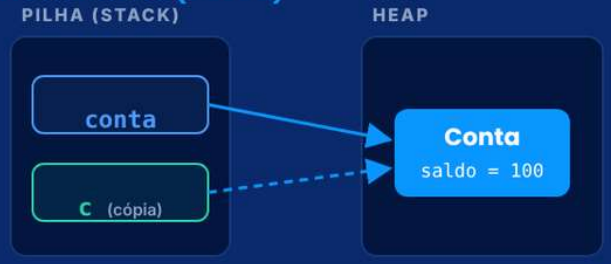
Aponta a cópia para outro objeto

```
void trocar(Conta c) {
    c = new Conta();
    c.saldo = 999;
}
trocar(conta);
// conta.saldo continua 100
```

× Original não muda

Só a **cópia** passou a apontar para um objeto novo. A variável de quem chamou continua no objeto antigo.

Na memória (caso 2)



Como a banca cobra

"Java passa objetos por referência." ERRADO
Só existe **passagem por valor**; com objetos, copia-se a referência.

"Um método pode alterar o estado de um objeto recebido." CERTO
A cópia da referência aponta para o mesmo objeto.

"Reatribuir o parâmetro muda a variável original." ERRADO
Só a cópia local passa a apontar para outro objeto.

"Concatenar uma String no método altera a original." ERRADO
String é **imutável**: s + "!" cria outro objeto. Vale também para Integer e demais wrappers.

Passagem de valor em Java

Tipos primitivos	Tipos por referência	Passagem de parâmetros
Cópia do valor	Cópia da referência	Sempre por valor

Questões de concursos

[CESPE/CEBRASPE 2026 Câmara de Ponta Porã/MS – Analista de Sistemas] Acerca de linguagens de programação, julgue o item que se segue.

- A linguagem Java utiliza passagem de parâmetro por valor, permitindo alteração do endereço de memória dos ponteiros originais das variáveis.

Questões de concursos

[CESPE/CEBRASPE 2026 Câmara de Ponta Porã/MS – Analista de Sistemas] Acerca de linguagens de programação, julgue o item que se segue.

- A linguagem Java utiliza exclusivamente passagem de parâmetro por valor, permitindo alteração do ~~endereço de memória dos ponteiros originais das variáveis~~ estado dos objetos referenciados, mas não a alteração da referência original da variável passada como argumento.
  - Gabarito: ERRADO.

Objeto elegível para coleta de lixo

É um objeto que não possui mais referências

Referências mantidas em uma variável são normalmente descartadas quando

Todas as referências a um objeto devem ser descartadas

A variável sai do escopo

Um valor null é atribuído

Antes que o objeto seja elegível para coleta de lixo

Em que ela se encontra

Explicitamente à variável-objeto

Garbage collector

É um programa que está dentro do JRE

Não é controlado pelo programador

Libera periodicamente a memória usada

A JVM decide quando executá-lo

Por objetos que não são mais referenciados

Garbage collector

Faz o seu trabalho automaticamente quando determina que é a hora certa

Não há um momento pré-determinado

Há a possibilidade do programador chamar o gabarge collector

Mesmo assim não há garantia de que este irá realmente ser executado no momento que é chamado

Garbage Collector em Java

A JVM libera **automaticamente** a memória do **heap** ocupada por objetos que nenhuma parte do programa consegue mais alcançar.

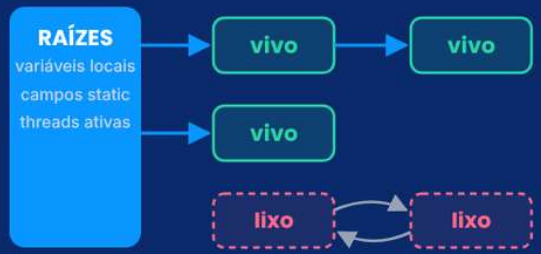
Gerência automática de memória

1 Quando um objeto vira lixo

```
Conta a = new Conta(); // obj 1
Conta b = new Conta(); // obj 2
a = null;    // obj 1: elegível
b = new Conta(); // obj 2: elegível
```

- Referência recebe **null**
- Referência é **reatribuída**
- Variável local sai de **escopo**
- **Ilha de isolamento**: objetos que só se referenciam entre si

2 Alcançável ou não?



O GC parte das **raízes** e segue as referências. O que não é alcançado é lixo, **mesmo com referência circular**.

3 Como ele trabalha

1. **Marcar** — a partir das raízes, marca os vivos
2. **Varrer** — libera os objetos não marcados
3. **Compactar** — junta os vivos e evita fragmentação

Fases **"stop-the-world"** pausam a aplicação.

O heap em gerações

GERAÇÃO JOVEM · Minor GC, frequente

Eden (objetos novos) | S0 | S1

sobreviveu a vários ciclos →

GERAÇÃO VELHA
objetos de vida longa
Major / Full GC, mais caro

Hipótese geracional: a maioria dos objetos morre jovem. Por isso o Eden é coletado com frequência e de forma barata. Coletores: Serial, Parallel, **G1 (padrão desde o Java 9)**, ZGC e Shenandoah.

Como a banca cobra

"O programador libera a memória com System.gc()." ERRADO
System.gc() é só uma **sugestão**; a JVM decide quando coletar.

"O GC atua sobre os objetos do heap." CERTO
Variáveis locais ficam na pilha e somem ao fim do método.

"Objetos com referência circular nunca são coletados." ERRADO
Se nenhuma raiz os alcança, são coletados.

"Com GC, não há vazamento de memória em Java." ERRADO
Referências esquecidas (ex.: em listas) seguram objetos e podem levar a **OutOfMemoryError**.

Exemplo

- Classe a = new Classe(“A”);
- Classe b = a;
- Classe c = new Classe(“C”);
- b = null;
- c = new Classe(“D”);



Questões de concursos

[CESPE/CEBRASPE 2026 VALEC – Analista – Especialidade: Analista de Sistemas] Julgue os itens a seguir, relativos a clean code e às linguagens de programação COBOL e Java.

- A linguagem Java utiliza ponteiros para o gerenciamento de endereços de memória, o que garante mais segurança para o uso do Garbage Collector.

Questões de concursos

[CESPE/CEBRASPE 2026 VALEC – Analista – Especialidade: Analista de Sistemas] Julgue os itens a seguir, relativos a clean code e às linguagens de programação COBOL e Java.

- A linguagem Java utiliza ponteiros para o gerenciamento de endereços de memória, o que garante mais segurança para o uso do Garbage Collector.
  - Gabarito: ERRADO.

Estruturas de Controle de Fluxo



De condição

- if-else
- switch



De repetição

- for
- while
- do-while



Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- A execução do código a seguir será bem-sucedida e imprimirá, em tela, 6 linhas.

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 2; j++) {
                System.out.println("Estudante "
+ i + ", Prova " + j);
            }
        }
    }
}
```

Comentários

- Código:
  - public class Main {
    - public static void main(String[] args) {
      - for (int i = 1; i <= 3; i++) {
        - for (int j = 1; j <= 2; j++) {
          - System.out.println("Estudante " + i + ", Prova " + j);
        - }
      - }
    - }
  - }

Comentários

- Resultado da execução:
  - Estudante 1, Prova 1
  - Estudante 1, Prova 2
  - Estudante 2, Prova 1
  - Estudante 2, Prova 2
  - Estudante 3, Prova 1
  - Estudante 3, Prova 2

Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- A execução do código a seguir será bem-sucedida e imprimirá, em tela, 6 linhas.
  - Gabarito: CERTO.

Modificadores de acesso

Modificador	Modificador	Classe	Pacote	Subclasse	Todos
public	Público	✓	✓	✓	✓
protected	Protegido	✓	✓	✓	✗
package-private	Pacote	✓	✓	✗	✗
private	Privado	✓	✗	✗	✗

Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- Considere a seguinte classe Estudante.

```
public class Estudante {
    private String nome;
    private int matricula;
    public Estudante(String nome, int
matricula) {
        this.nome = nome;
        this.matricula = matricula;
    }
}
```

Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- Considere, também, o seguinte trecho de código, que visa cadastrar e exibir os dados de um estudante.

```
public class Main {
    public static void main(String[] args) {
        Estudante e = new Estudante("Carlos",
12345);
        System.out.println("Nome: " + e.nome);
        System.out.println("Matricula:    "    +
e.matricula);
    }
}
```

Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- A execução dos trechos de código precedentes criará um objeto da classe *Estudante* com os dados informados e exibirá as informações na tela sem apresentar erro de compilação.

Comentários

- Código:
  - public class Estudante {
    - private String nome;
    - private int matricula;
    - public Estudante(String nome, int matricula) {
      - this.nome = nome;
      - this.matricula = matricula;
    - }
  - }

Comentários

- Código:
  - public class Main {
    - public static void main(String[] args) {
      - Estudante e = new Estudante("Carlos", 12345);
      - System.out.println("Nome: " + e.nome);
      - System.out.println("Matrícula: " + e.matricula);
    - }
  - }

Questões de concursos

[CESPE/CEBRASPE 2025 FUB – Técnico de Tecnologia da Informação] No que se refere à linguagem de programação Java, julgue o item subsecutivo.

- A execução dos trechos de código precedentes criará um objeto da classe *Estudante* com os dados informados e exibirá as informações na tela sem apresentar erro de compilação.
  - Gabarito: ERRADO.

Herança em Java

De **classes**, a herança é **simples**: só uma superclasse. De **interfaces**, é **múltipla**: vários contratos ao mesmo tempo.

extends 1 · implements N

1 Classes: herança simples

extends · só UMA superclasse




```
class ContaCorrente extends Conta { }
class X extends A, B { } // não compila
```

A subclasse herda **estado e implementação** de uma única classe. Sem **extends**, ela herda de **Object**.

2 Interfaces: herança múltipla

implements · várias interfaces




```
class ContaCorrente extends Conta
    implements Tributavel, Auditavel { }
interface Investimento
    extends Tributavel, Auditavel { }
```

A classe assume **vários contratos**. E uma interface pode **estender várias** interfaces.

3 E o problema do diamante?

por que não há herança múltipla de classes

Duas superclasses com o mesmo método e atributos gerariam **ambiguidade** sobre qual herdar. Java evita isso nas classes.

```
interface A { default void log() { … } }
interface B { default void log() { … } }
class C implements A, B {
  public void log() {
    A.super.log(); // escolhe
  }
}
```

Desde o **Java 8**, interfaces têm métodos **default**. Se dois colidem, a classe é **obrigada a sobrescrever**.

Classe × interface

	Classe	Interface
Palavra-chave	`extends`	`implements` (classe) / `extends` (interface)
Quantas?	Uma superclasse	Várias interfaces
Herda	Estado e implementação	Contrato (e métodos default)
Instanciar?	Sim, se não for abstract	Não: não se usa new

Como a banca cobra

- ERRADO — "Java permite herança múltipla de classes."
- CERTO — "Uma classe pode implementar várias interfaces."
- CERTO — "Uma interface pode estender várias interfaces."
- ERRADO — "Interface nunca tem método com corpo."
- CERTO — "Classe final não pode ser estendida."

Questões de concursos

[CESPE/CEBRASPE 2026 TELEBRAS – Especialista em Gestão de Telecomunicações – Analista de Tecnologia da Informação] A respeito de desenvolvimento de sistemas, julgue o item a seguir.

- A linguagem Java permite herança múltipla de classes, de modo que uma classe pode ser subclasse de outras duas classes diretamente.

Questões de concursos

[CESPE/CEBRASPE 2026 TELEBRAS – Especialista em Gestão de Telecomunicações – Analista de Tecnologia da Informação] A respeito de desenvolvimento de sistemas, julgue o item a seguir.

- A linguagem Java não permite herança múltipla de classes, de modo que uma classe não pode ser subclasse de outras duas classes diretamente.
  - Gabarito: ERRADO.

Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Perito Criminal Federal – Área 3: Informática Forense] Considerando os princípios da programação orientada a objetos, bem como os conceitos de linguagens de programação procedurais, julgue os itens subsequentes.

- Em linguagens de programação como Java e C#, a herança múltipla de classes é um recurso amplamente suportado e incentivado para maximizar a reutilização de código.

Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Perito Criminal Federal – Área 3: Informática Forense] Considerando os princípios da programação orientada a objetos, bem como os conceitos de linguagens de programação procedurais, julgue os itens subsequentes.

- Em linguagens de programação como Java e C#, a herança múltipla de classes não é um recurso amplamente suportado e incentivado para maximizar a reutilização de código.
  - Gabarito: ERRADO.

Interfaces em Java

Uma interface é um **contrato**: define os métodos que uma classe deve oferecer, sem dizer como.

Contrato + polimorfismo

1 Anatomia de uma interface

o contrato: diz O QUE fazer, não COMO

```
interface Tributavel {
  double ALIQUOTA = 0.1;          // constante  public static final
  double calcularTributo();       // abstrato   public abstract
  default String resumo() {…}     // default    Java 8
  static double teto() {…}        // static     Java 8
  private void log() {…}          // private    Java 9
}
```

Em constantes e métodos abstratos, os modificadores em cinza são **implícitos**. Interface **não tem construtor** nem atributos de instância.

2 Implementando o contrato

implements · uma classe, vários contratos

```
class ContaCorrente implements Tributavel {
  // obrigatório: public
  public double calcularTributo() {
    return saldo * ALIQUOTA;
  }
}
Tributavel t = new ContaCorrente();
t.calcularTributo(); // polimorfismo
```

- Implementa **todos** os métodos abstratos, ou a classe vira **abstract**
- O método implementado é **public**: não dá para reduzir a visibilidade
- A interface serve de **tipo** para a variável

3 Interface funcional

base das expressões lambda (Java 8)

Tem **exatamente um método abstrato**. Pode ter quantos **default** e **static** quiser.

```
@FunctionalInterface
interface Regra {
  boolean aprova(Conta c);
}
Regra saldoPositivo = c -> c.saldo > 0;
```

Exemplos da API: **Runnable**, **Comparator**, **Predicate**, **Function**. A anotação é opcional, mas faz o compilador conferir a regra.

Classe abstrata × interface

	Classe abstrata	Interface
Instanciar com new	Não	Não
Construtor	Sim	Não
Atributos	De instância, qualquer tipo	Só constantes
Herança	`extends` uma só	`implements` várias
Métodos com corpo	Sim	default, static, private

Como a banca cobra

- ERRADO — "Interface pode ser instanciada com new."
- CERTO — "Atributos de interface são constantes."
- ERRADO — "O método implementado pode ser protected."
- ERRADO — "Interface pode ter construtor."
- CERTO — "Interface funcional tem um único método abstrato."

Questões de concursos

[CESPE/CEBRASPE 2025 TRT 10ª Região – Analista Judiciário – Área: Apoio Especializado – Especialidade: Tecnologia da Informação] Acerca de desenvolvimento de sistemas, julgue o próximo item.

- Em Java, uma das maneiras de se implementar o polimorfismo é a partir de interfaces que permitem o tratamento de objetos de classes diferentes de forma uniforme.

Questões de concursos

[CESPE/CEBRASPE 2025 TRT 10ª Região – Analista Judiciário – Área: Apoio Especializado – Especialidade: Tecnologia da Informação] Acerca de desenvolvimento de sistemas, julgue o próximo item.

- Em Java, uma das maneiras de se implementar o polimorfismo é a partir de interfaces que permitem o tratamento de objetos de classes diferentes de forma uniforme.
  - Gabarito: CERTO.

02

MÓDULO 02

Python

Indentação | Lista | Tupla | Conjunto | Dicionário

Indentação em Python

Em Python, o **recuo do código define os blocos**. Não há chaves: a indentação faz parte da sintaxe.

Recuo = bloco

1 O recuo é a sintaxe

cada nível = 4 espaços (PEP 8)

```
def sacar(conta, valor):                    nível 0
    if valor <= conta.saldo:                nível 1 · corpo da função
        conta.saldo -= valor                nível 2 · corpo do if
        return True
    else:                                   volta ao nível 1
        print("Saldo insuficiente")         corpo do else
        return False
print("fim")                                fora da função
```

`:` abre um bloco · mesmo recuo = mesmo bloco · recuar menos = fecha o bloco

2 Os erros de indentação

o código nem chega a rodar

Faltou recuar

```
if x > 0:
print(x)
IndentationError: expected an indented block
```

Recuo inesperado

```
x = 1
    y = 2
IndentationError: unexpected indent
```

Tab misturado com espaço

```
if ok:
    a()
⇥   b()
TabError: inconsistent use of tabs and spaces
```

3 Java × Python

chaves e ponto e vírgula × dois-pontos e recuo

JAVA · bloco por **chaves**; recuo é só estética

```
if (valor <= saldo) {
    saldo -= valor;
} else {
    aviso();
}
```

PYTHON · bloco pelo **recuo**; sem chaves nem ;

```
if valor <= saldo:
    saldo -= valor
else:
    aviso()
```

Regras rápidas

: no fim da linha	def, if, elif, else, for, while, try, except, with, class
Bloco vazio	use `pass`: o bloco não pode ficar sem nada
Dentro de ( ) [ ] { }	a linha pode continuar sem regra de recuo
Uma linha só	if x: print(x) vale, mas a PEP 8 desaconselha

Como a banca cobra

- ERRADO — "Em Python, a indentação é apenas estética."
- ERRADO — "Python usa chaves para delimitar blocos."
- CERTO — "A PEP 8 recomenda 4 espaços por nível."
- CERTO — "Misturar tabs e espaços pode gerar TabError."
- CERTO — "Um bloco vazio pode usar a instrução pass."

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA Técnico – Área: Suprimento, Manutenção e Serviços – Subárea: Novas Tecnologias] Acerca de programação em Python, julgue o item subsequente.

- Os blocos de código em Python são delimitados pelo uso de chaves, que abrem e fecham cada bloco.

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA Técnico – Área: Suprimento, Manutenção e Serviços – Subárea: Novas Tecnologias] Acerca de programação em Python, julgue o item subsequente.

- Os blocos de código em Python são delimitados pelo uso de ~~chaves~~ indentação, que ~~abrem e fecham~~ delimita cada bloco.
  - Gabarito: ERRADO.

Questões de concursos

[CESPE/CEBRASPE 2024 CAU/BR – Analista de Sistema de Informação] Em relação à linguagem de programação Python, julgue o item a seguir.

- Nessa linguagem de programação, a indentação de código é considerada uma boa prática, pois não afeta a execução de um programa.

Questões de concursos

[CESPE/CEBRASPE 2024 CAU/BR – Analista de Sistema de Informação] Em relação à linguagem de programação Python, julgue o item a seguir.

- Nessa linguagem de programação, a indentação de código não é considerada apenas uma boa prática, pois ~~não~~ afeta a execução de um programa.
  - Gabarito: ERRADO.

Questões de concursos

[CESPE/CEBRASPE 2023 CNMP – Analista do CNMP – Àrea: Tecnologia da Informação e Comunicação – Especialidade: Suporte e Infraestrutura] Julgue o item subsecutivo, que se referem a conceitos de programação e banco de dados.

- Na linguagem Python, os blocos de código são separados por chaves.

Questões de concursos

[CESPE/CEBRASPE 2023 CNMP – Analista do CNMP – Àrea: Tecnologia da Informação e Comunicação – Especialidade: Suporte e Infraestrutura] Julgue o item subsecutivo, que se referem a conceitos de programação e banco de dados.

- Na linguagem Python, os blocos de código não são separados por chaves, mas usa-se indentação.
  - Gabarito: ERRADO.

Coleções em Python

Quatro estruturas nativas para agrupar dados. O que muda entre elas: **ordem**, **mutabilidade** e **repetição**.

list · tuple · set · dict

[ ] Lista
`list`

```
saldos = [100, 250, 100]
saldos.append(80)
saldos[0]      # 100
saldos[1:3]    # [250, 100]
```

- ✓ Ordenada
- ✓ Mutável
- ✓ Aceita repetidos

Métodos: append, insert, remove, pop, sort

⚠ Pode misturar tipos: [1, "a", 2.5]

( ) Tupla
`tuple`

```
ponto = (10, 20)
x, y = ponto   # desempacota
ponto[0] = 5   # TypeError
um = (5,)      # tupla de 1
```

- ✓ Ordenada
- × Mutável
- ✓ Aceita repetidos

Métodos: count, index

⚠ (5) é só o número 5; a tupla precisa da vírgula: (5,)

{ } Conjunto
`set`

```
ids = {3, 1, 3, 2}
# {1, 2, 3}: sem repetidos
ids.add(4)
a | b   a & b   a - b
```

- × Ordenado
- ✓ Mutável
- × Aceita repetidos

Métodos: add, remove, discard, union

⚠ Conjunto vazio é set(). {} cria um dicionário vazio!

{k:v} Dicionário
`dict`

```
conta = {"ag": "001",
         "saldo": 100}
conta["saldo"]      # 100
conta.get("cpf", 0)  # 0
```

- ✓ Ordem de inserção
- ✓ Mutável
- × Chaves repetidas

Métodos: get, keys, values, items, pop

⚠ Chave tem de ser imutável (hashable): lista não pode ser chave

Comparativo

COLEÇÃO	LITERAL	ORDENADA	MUTÁVEL	REPETIDOS	ACESSO
Lista	`[1, 2]`	Sim	Sim	Sim	Índice
Tupla	`(1, 2)`	Sim	Não	Sim	Índice
Conjunto	`{1, 2}`	Não	Sim	Não	Não há
Dicionário	`{"a": 1}`	Sim*	Sim	Chaves não	Chave

* O dicionário mantém a ordem de inserção desde o Python 3.7.

Como a banca cobra

- ERRADO — "{} cria um conjunto vazio."
- CERTO — "Tuplas são imutáveis."
- ERRADO — "(5) é uma tupla de um elemento."
- ERRADO — "Conjuntos permitem acesso por índice."
- ERRADO — "Uma lista pode ser chave de dicionário."

Exemplos

Lista []	• linguagens = ["Python", "Java", "PHP", "C"]
Tupla ()	• linguagens = ("Python", "Java", "PHP", "C")
Conjunto {}	• linguagens = {"Python", "Java", "PHP", "C"}
Dicionário {}	• optimus = {"id": 1, "nome": "Optimus Prime"}

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA – Analista – Área: Gestão da Informação – Subárea: Engenharia de Infraestrutura e Tecnologia da Informação] Julgue o seguinte item, referente à programação e estruturas de dados.

- O código a seguir, em Python, retornará Agropecuária como resultado da sua execução.

```
x = ["Brasileira", "de", "Agropecuária"]
x[1] = "Empresa"
y = x.pop(2)
x.append("EMBRAPA")
x.remove("Brasileira")
print (x[1])
```

Comentários

- Código:
  - x = ["Brasileira", "de", "Agropecuária"]
  - x[1] = "Empresa"
  - print(x)
  - y = x.pop(2)
  - x.append("EMBRAPA")
  - x.remove("Brasileira")
  - print(y)
  - print(x)
  - print(x[1])

Comentários

- Resultado da execução:
  - ['Brasileira', 'Empresa', 'Agropecuária']
  - Agropecuária
  - ['Empresa', 'EMBRAPA']
  - EMBRAPA

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA – Analista – Área: Gestão da Informação – Subárea: Engenharia de Infraestrutura e Tecnologia da Informação] Julgue o seguinte item, referente à programação e estruturas de dados.

- O código a seguir, em Python, retornará ~~Agropecuária~~ EMBRAPA como resultado da sua execução.
  - Gabarito: ERRADO.

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA – Técnico – Área: Suprimento, Manutenção E Serviços – Subárea: Novas Tecnologias] Acerca de programação em Python, julgue o item subsequente.

- Em Python, as listas são coleções heterogêneas e mutáveis de objetos de qualquer tipo, inclusive de outras listas.

Questões de concursos

[CESPE/CEBRASPE 2025 EMBRAPA – Técnico – Área: Suprimento, Manutenção E Serviços – Subárea: Novas Tecnologias] Acerca de programação em Python, julgue o item subsequente.

- Em Python, as listas são coleções heterogêneas e mutáveis de objetos de qualquer tipo, inclusive de outras listas.
  - Gabarito: CERTO.

Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Agente de Polícia Federal] Julgue o próximo item, relativos aos modos de transmissão de dados, ao modelo OSI, à linguagem Python e aos conceitos de falso positivo (FPIR) e falso negativo (FNIR) dos testes de acurácia do NIST.GOV.

- Em Python, listas são estruturas de dados imutáveis, o que as torna ideais para serem usadas como chaves de dicionários (dict).

Questões de concursos

[CESPE/CEBRASPE 2025 Polícia Federal – Agente de Polícia Federal] Julgue o próximo item, relativos aos modos de transmissão de dados, ao modelo OSI, à linguagem Python e aos conceitos de falso positivo (FPIR) e falso negativo (FNIR) dos testes de acurácia do NIST.GOV.

- Em Python, ~~listas~~ tuplas são estruturas de dados imutáveis, o que as torna ideais para serem usadas como chaves de dicionários (dict).
  - Gabarito: ERRADO.

CEBRASPE | PC AL · AGENTE E ESCRIVÃO

JAVA E PYTHON

Obrigado!

Bons estudos e
bora para a aprovação.

ANTES DA PROVA, REVISE

1. Java compila para **bytecode**; a **JVM** executa em qualquer SO.
2. Em Java, a passagem é **sempre por valor**; o GC é **automático**.
3. Herança **simples** de classes, **múltipla** de interfaces.
4. Em Python, a **indentação** delimita os blocos.
5. Lista **mutável**, tupla **imutável**, set **sem repetidos**, dict **chave: valor**.

destrinchandoTI

Prof. Rogério Araújo
@destrinchandoti

@destrinchandoti | @profrogeraoaraujo | rogeraoaraujo.com.br