**Raoni Martins Raposo. 🡪** Aluno do Fluency sem Limites.

**Exercícios Unidade 2.**

**Linguagens e Paradigmas.**

1. A única linguagem compreendida pelo hardware de computador é uma linguagem de máquina.
   1. de máquina.
   2. simbólica.
   3. de alto nível.
   4. nenhuma das anteriores.
2. Faça uma distinção entre linguagem de máquina e linguagem de alto nível.  
     
   Em resumo, a linguagem de máquina é a linguagem que o computador entende diretamente, enquanto a linguagem de alto nível é uma linguagem mais abstrata e amigável ao usuário que deve ser traduzida para a linguagem de máquina antes que o computador possa executá-la como o Java.
3. Descreva com suas palavras o que é o processo de compilação.  
     
   O processo de compilação é um processo fundamental na programação que transforma o código escrito em uma linguagem de programação de alto nível em uma linguagem que o computador pode entender e executar, que é a linguagem de máquina.
4. Descreva com suas palavras o que é o processo de interpretação.  
     
   O processo de interpretação é um aspecto fundamental da execução de programas de computador. Diferentemente da compilação, onde todo o código é convertido em linguagem de máquina antes da execução, a interpretação converte e executa o código linha por linha durante a execução do programa.

Os interpretadores são geralmente mais fáceis de usar, especialmente para a programação de iniciantes, porque fornecem mensagens de erro mais detalhadas e permitem a execução interativa de código. No entanto, os programas interpretados geralmente rodam mais devagar do que os programas compilados.

Alguns exemplos de linguagens interpretadas incluem Python, Ruby e JavaScript.

1. Javascript é uma linguagem interpretada.
   1. de máquina.
   2. interpretada. Como mencionado acima. 🙄
   3. compilada.
   4. nenhuma das anteriores.
2. Descreva o que é o paradigma de orientação a objetos.  
     
   O paradigma de orientação a objetos é um modelo de programação que organiza o software como uma coleção de objetos distintos, cada um representando uma instância de alguma classe.
3. Descreva o que é uma linguagem com tipagem estática.  
     
   Uma linguagem de programação com tipagem estática é aquela em que o tipo de cada variável é conhecido em tempo de compilação, ou seja, antes da execução do programa. Isso significa que você precisa declarar explicitamente o tipo de cada variável no código.

Como exemplos de linguagens de programação com tipagem estática incluem Java, C, C++, Rust, Go, entre outras.

1. Descreva o que é uma linguagem com tipagem fraca.  
     
   Uma linguagem de programação com tipagem fraca é aquela em que o sistema de tipos permite e, em muitos casos, converte automaticamente os tipos de dados. Isso significa que o programador pode, por exemplo, concatenar uma “string” com um número sem ter que converter explicitamente o número em uma string primeiro. Em uma linguagem de tipagem fraca, se você tentar usar um valor de um tipo onde um valor de outro tipo é esperado, o interpretador ou compilador tentará converter automaticamente o valor para o tipo esperado.

As linguagens de tipagem fraca tendem a ser mais flexíveis porque permitem operações que seriam proibidas em linguagens de tipagem forte. Isso pode tornar mais fácil escrever código rapidamente como em Python.

Essa flexibilidade vem com riscos. A conversão automática de tipos pode levar a resultados inesperados, o que pode causar BUGS sutis que são difíceis de detectar. 🤦🏻‍♂️

É importante notar que a distinção entre “tipagem fraca” e “tipagem forte” não é sempre clara, e diferentes pessoas podem usar esses termos de maneiras ligeiramente diferentes.

1. Descubra o que será apresentado no código JavaScript abaixo:

|  |
| --- |
| console.log(a);  var a = 4;  console.log(a);  console.log(typeof a)  a = a + '2'  console.log(a)  console.log(typeof a) |

* 1. console.log(a); -> Neste ponto, a variável a foi declarada, mas ainda não foi inicializada. Portanto, seu valor é “undefined”. Então, “undefined” será impresso no console.
  2. var a = 4; -> Aqui, a variável a é inicializada com o valor 4.
  3. console.log(a); -> Agora, “a” tem o valor 4, então 4 será impresso no console.
  4. console.log(typeof a) -> Neste ponto, “a” é um número, então number será impresso no console.
  5. a = a + '2' -> Aqui, “a” é somado com a string '2'. Em JavaScript, quando você tenta somar um número e uma “string”, o número é convertido em uma string e então concatenado. Portanto, a agora será a string '42'.
  6. console.log(a) -> Agora, a é a string '42', então '42' será impresso no console.
  7. console.log(typeof a) -> Neste ponto, “a” é uma string, então string será impresso no console.

Então, a saída do código será:

undefined  
4  
number  
42  
string  
undefined