

Ex: . if (condição 1) {
instrução 1
else if (condição 2) {
instrução 2
else if (condição 3) {
instrução 3
else { ...
instrução N

simbologia 1

if () {
else if () {
else {
}

simbologia 2

4) Tipos de operadores:

OPERADORES ARITMÉTICOS: são os operadores básicos da Matemática, que aprendemos no ensino médio para operações simples. Ordem PEMDAS: ;

P E M D A S
↓ ↓ ↓ ↓ ↓
parênteses exponenciação multiplicação divisão adição
subtração
:> (), **, *, /, //, %, +, -

OPERADORES DE IMPORTAÇÃO: vem por blocos ou partes;

import, math, import math, from, float, Parse(), int.Parse(), toFixed()

OPERADORES CONDICIONAIS: são hipotéticos;

if, else if, else... e suas combinações!

OPERADORES RELACIONAIS: 6 tipos, dependentes ou reais entre si no caso de igualdades;

==, !=, >, <, >=, <=

EXPRESSÕES BOOLEANAS: true or false;

OPERADORES LÓGICOS: 3 tipos, são distintos entre si.

&&, ||, !

5) Conceito: uma estrutura condicional é baseada em uma condição que se for atendida o algoritmo toma uma decisão. Entender que a condição é verdadeira, deve-se exibir um console.log() ou "mensagem" de alerta que "o número A é maior que o número B", se esta condição não for atendida, ou seja, se ela for falsa, o algoritmo não irá tomar nenhuma ação ou exibir uma mensagem de "valor inválido". Pois a condição não é atendida pelo último caso.

TIPOS: if, else if, else if -> se, se não, senão

• É montado por parênteses e chaves. {> (); }

• Usagens de uso (estrutura): são aninhados ou o mesmo que várias condições em cascata, um IF dentro de outro IF. Exemplo para aninhamento, agora a mensagem aparece caso o valor seja igual a zero.

MÚLTIPLAS CONDICIONAIS: ilustrações das situações-problema.

(IF... ELSE IF... IF... ELSE IF... ELSE)

algar a transmissão de dados. Esses dispositivos podem ser o mouse, o teclado, o pen drive e as caixas de som, entre outros;

- PROCESSAMENTO: serve para organizar de forma adequada os dados de entrada, para depois serem convertidos à saída;

- SAÍDA: é caracterizada pelos equipamentos que mostram os dados como, por exemplo, o monitor de vídeo ou uma impressora.

3) Funções: conjunto nomeado de instruções sequenciadas para quebrar tarefas maiores em pequenos tarefas ou subtarefas. Melhoram a qualidade do seu código por meio da organização de códigos e ideias. Uma rotina ou método!

Por que funções?

- Facilidade de leitura, pois um grupo de instruções passa a ser nomeado, tem significado explícito;

- Redução de linhas de código, devido ao reaproveitamento invés de repetição;

- Facilidade de depuração, pelo isolamento das funcionalidades;

- Reutilização no mesmo arquivo ou em outros projetos;

- Agrupamento de várias funções de modo a chamá-las.

ESTRUTURA E ELEMENTOS:

- MODULARIZAÇÃO: quebra de problemas em um bloco de instruções com efeito para a resolução de problemas;

- CHAMADA DE FUNÇÕES: invocação e uso. Exs.: `Math()`, `Number()`, `prompt-pyne()`, ...

- ENCAPSULAMENTO: trata-se de um processo de encapsular, deixar uma área separada, e dar um nome para que você possa reutilizar futuramente a cápsula de funções, ou seja, um agrupamento de funções que tem um código que é um pouco mais complexo que você, não quer que ele fique no fluxo principal, se você encapsula uma função, nomeia ela, dar um nome adequado a esta função e depois a chama para encapsular determinado processo. Você delega uma função para uma tarefa, esperando um retorno;

- GENERALIZAÇÃO: é quando eu melhora ou aprimoro meu código deixando pe-dacos, eu posso generalizar, chamar ela (a função) passando valores diferentes que nos configurando o(s) pe-dacos finais do meu problema. Isto é, genérico, configurável e também independente.

feito para uma unidade de código que gera uma ordem para o computador.

b) EXPRESSÃO BOOLEANA: é uma expressão composta de valores, variáveis e operadores relacionais (6 deles: $=$, \neq , $>$, $<$, $>=$, $<=$) e lógicos (3 deles: $\&\&$, $\|\$ e $!$) que retornam somente 2 valores: true ou false.

c) VARIÁVEIS: originam-se pelo termo "var", que significa vaga ou espaço de memória armazenados. Dentro de uma variável pode existir "var", "let" ou "const" para funcionar como dicas de estrutura e organização para não deixar confuso os programas. Exemplo local ou global.

VALORES: é uma atribuição que recebe argumentos quando digita-se uma variável. Sinal de $(=)$ recebe os valores.

TIPOS: é a classificação dos tipos primitivos trabalhados em Java Script:

number { "Infinity"
 "NaN" (e NaN)
string
boolean
null
undefined
object { "Array"
function

d) LINGUAGENS FORMAIS: criados pelos pensos, aplicações específicas, têm regras de sintaxe e regras que governam a estrutura de declarações.

Exs.: Os matemáticos usam, é uma linguagem formal trabalhar números e símbolos;
Os químicos usam, é uma linguagem formal para representar a estrutura química de moléculas;

As linguagens de programação são idiomas formais criados para expor operações de computação, exs.: C, C++, Python, Java Script, etc.

e) O "fluxo de dados" é o caminho que as informações percorrem em um computador, passando pelos etapas de entrada, de processamento e de saída. E isso só é possível graças ao sistema operacional, que é um conjunto de programas que têm a função de gerenciar todos os recursos e as tarefas da máquina. Por meio desse sistema, as etapas do fluxo de dados funcionam da seguinte forma:

- ENTRADA: serve para os dispositivos que podemos conectar ao computador para re-

Instituto Federal do Piauí - IFPI
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)
Algoritmos e Programação, Prof.º Rogério Silva, ADS I - 2021.1
Aluno: Vinícius Gomes Araújo Costa, matrícula: 20191ADS0274

Atividade Individual - "Meio de Disciplina" - PARTE 1

1) Também chamada de raciocínio computacional, é a lógica que nós, desenvolvedores de software, adquirimos na tentativa de solucionar problemas e questões pertinentes no dia a dia, na universidade, na cidade ou até mesmo no país. Uma sequência para a posse de tarefas com um fim específico: a resolução de problemas de caráter totalmente inteligente, aprimorando a cognição relacionadas com a construção de algoritmos. Sistematizar e organizar modelos ou subrotinas de código como por exemplo: aplicativos, navegadores e programas fazem com que o computador realize atividades, isto é, atividades de programação exercidas na comunicação homem-máquina através de uma linguagem de programação.

3 pilares da Computação:

ABSTRAÇÃO: quebra de grandes tarefas em pequenos tarefas de modo a solucionar problemas usando algoritmos, é preciso depurar as informações por meio de uma linguagem em que se encontra separando ou decompondo as partes do problema;

AUTOMACÃO: usar um computador para automatizar tarefas além de ser transformador e progressor, capacita um projeto através de uma linguagem de programação específica. Outrossim, os desenvolvedores aperfeiçoam/iteram código e utilizam normas para diversas atividades como filtragem de dados, um gráfico por análise e o outro por satisfação de resultado;

ANÁLISE: Capacidade de um homem e uma máquina solucionar o problema de forma clara, eficiente e objetiva. Utiliza argumentos críticos sobre a questão - situação com os pilares 1 e 2 não sejam atendidos. Com isso, é feita uma negociação pelo usuário para um posicionamento final do problema.

2) a) INSTRUÇÃO: é tudo que eu possa colocar como valores, variáveis, operadores e chamados de funções. A prática de exercícios identifica padrão/parâmetros, e dar para o mundo real uma situação-problema e trazê-la na forma de um algoritmo representado por uma linguagem de programação.

EXPRESSÃO: uma unidade de código que tem valor quando mediada com