## #Aula 1

### Conceitos básicos

##### Dissertativos

1. Qual analogia nós usamos na aula para representar a memória do computador, seus espaços, e conteúdos?

**A analogia do Estacionamento.**

1. O que entendemos como ‘bloco de comandos’?

**Um conjunto ou lote de instruções entre o início e o fim do programa. É executado em sequência lógica, um após o outro, até que o programa termine.**

1. Quais palavras reservadas (ou símbolos) usamos para delimitar esse bloco, em diferentes linguagens?

**Utilizamos as chaves: { e }**

1. O que significa para o programa a estrutura sequencial?

**Um programa que possua um início definido, o bloco de comandos e um fim.**

**Ou seja, sua estrutura executada de forma lógica, uma linha após a outra, de cima para baixo.**

1. Qual o símbolo utilizado para terminar instruções?

**O ponto e vírgula (;)**

1. Via programação, como exibir uma mensagem simples na tela?

**Através do comando: console.log(“Mensagem a ser exibida”);**

1. O que fazer quando ao executar um programa, perceber que o código que tinha sido digitado “não apareceu no console”? Qual elemento visual do editor ajuda a perceber isso?

**Utilizar a depuração para inspecionar o código e encontrar o possível erro. O elemento visual que possibilita perceber na execução do programa a entrada através da digitação é o prompt.**

1. Qual facilidade o *Coderunner* apresenta para executar apenas um trecho do código?

**A facilidade de navegar pelo código e inspecioná-lo através do Depurador.**

1. Faz diferença nas linguagens de programação próxima do C, como Java ou JavaScript, escrever instruções em maiúsculas ou minúsculas? Justifique.

**Sim, faz diferença. São linguagens chamadas de Case Sensitive e portanto fazem diferenciação entre maiúsculas e minúsculas. Por exemplo, a variável ‘nome’ é diferente da variável ‘Nome’.**

1. O que faz o comando console.log? Qual a diferença entre este e o document.write?

**O comando console.log exibe no Console uma mensagem. Já o comando document.write é uma função do JavaScript que é escrita no html da página, também para exibição.**

1. Como fazer uma quebra de linha no texto do console?

**Deve-se utilizar o comando “\n” para quebrar uma string (texto).**

1. Para que serve a instrução “use strict” no início do arquivo Js?

**Normalmente, devemos declarar uma variável antes de usá-la. Contudo, no passado, era tecnicamente possível usar uma variável atribuindo um valor a ela antes de declara-la. O comportamento padrão dos browsers aceitam isso hoje para manter a compatibilidade com scripts antigos. Contudo, está é uma má prática. Podemos forçar a declaração de uma variável antes do seu uso usando o “strict mode”. Para isso o “use strict” deve sempre ser declarada no início do documento JavaScript. Como a primeira linha do nosso arquivo. Caso contrário, não funciona**.

1. Escreva a tecla de atalho, ou botão, ou item de menu no VSCode para:
   1. Salvar todos os documentos abertos - **CTRL + S**
   2. Executar o programa (No Coderunner) - **CTRL + ALT + N**
   3. Duplicar uma linha (criar uma cópia da linha corrente) – **SHIFT + ALT + Seta para baixo**
   4. Abrir um cursor multilinha (digitação múltipla) – **SHIFT + ALT + I**
   5. Exibir o painel de comandos – **CTRL + SHIFT + P**
   6. Alternar o modo de exibição do texto, em linha contínua ou com quebra ao final da página – **ALT + Z**
   7. Adicionar ou remover comentário na linha corrente - **//**
   8. Exibir a lista dos atalhos. – **Arquivo -> Preferências -> Atalhos de Teclado ou CRTL K CTRL S**
2. Para quê serve a indentação? Ela influencia o programa?

**A indentação serve para ressaltar a estrutura do código, aumentando sua legibilidade.**

**Ela não tem influência sobre o programa, apenas sobre o programador que poderá ter um código mais legível para trabalhar.**

1. Existe diferença entre a programação embutida no HTML (utilizando a tag script) e o arquivo com código JavaScript separado?

**Em relação ao código JS a ser executado através da leitura do mesmo não há diferença, pode somente melhorar a organização dos arquivos.**