

# Avaliação Programação Imperativa (2º Checkpoint)

Total de pontos 16/20 ?

Segundo Checkpoint - Turma 06 NT

E-mail \*

humbertoccmoraes@yahoo.com.br

0 de 0 pontos

Informe seu nome. \*

Humberto Cicero Cesar de Moraes

Informe o nome do seu professor. \*

Ana Teixeira

Questões

16 de 20 pontos

20 questões cada questão vale 1 ponto, responda com atenção.  
Questões dissertativas serão corrigidas após entrega.



- ✓ 1 - Considere o código abaixo e escolha a alternativa que mostra as saídas corretas. \*

1/1

```
function soma(a=1, b=3)
{return a+b;}

console.log(soma());
console.log(soma(3));
console.log(soma(1,2));
```

- ☐ 4 4 3
- ☒ 4 6 3
- ☐ 0 6 3
- ☐ 4 3 3
- ☐ 3 1 2



- ✓ 2 - Quais os operadores utilizados no Javascript para: concatenar uma string, comparar o conteúdo de uma variável e seu tipo, operador de negação e and lógico respectivamente: \*

1/1

- ☐ + == ! ||
- ☐ + == ! &
- ☐ ! === . &
- ☐ + == !! ||
- ☒ + === ! &&



✓ 3 - Dado o array: `let numeros= [10, 8, 7, 99, 0, 1, 6]`. Qual comando devemos executar para que a saída seja: `[ 99, 10, 8, 7, 6, 1, 0]` ? \*

1/1

- ☐ `numeros.sort(function(a,b){ return a-b;});`
- ☐ `numeros.sort(function(a,b){ return a*b;});`
- ☒ `numeros.sort(function(a,b){ return b-a;});`
- ☐ `numeros.sort(function(a,b){ return a,b;});`
- ☐ `numeros.sort(function(a,b){ a-b;});`



✓ 4 - Após a execução do código a seguir, assinale a alternativa correta para o valor de z. \*

1/1

```
var z=0;
for (var i=20; i<50; i+=10){
    z+=i;}
console.log(z);
```

- ☐ 50
- ☐ 20
- ☒ 90
- ☐ 60
- ☐ 100



## ✓ 5 - Com relação ao Javascript assinale a alternativa correta: \*

1/1

- ☐ O operador ++ incrementa 2, o operador -- decrementa 1, o operador % calcula a porcentagem, o operador + concatena e soma
- ☐ O operador ++ incrementa 1, o operador -- decrementa 1, o operador % calcula a porcentagem, o operador + concatena ou soma
- ☐ O operador ++ incrementa -1, o operador -- decrementa 1, o operador % calcula o resto, o operador + concatena ou soma
- ☐ O operador ++ decrementa 1, o operador -- incrementa 1, o operador % calcula o resto, o operador + concatena e soma
- ☒ O operador ++ incrementa 1, o operador -- decrementa 1, o operador % calcula o resto, o operador + concatena ou soma ✓

## ✗ 6 - Explique o que significa: instanciar um objeto. \*

.../1

é criar um novo objeto de uma classe antes definida, por exemplo o método new para criar um novo objeto

## ✗ 7 - Com base no script abaixo, explique cada uma das linhas que contenha código. \*

.../1

```
1 let frutasAmarelas=['Melão','Mamão','Limão siciliano'];
2 let frutasVermelhas=['Morango','Cereja','Maçã red'];
3 let frutasVerdes=['Limão','Kiwi','Maçã verde'];
4
5 frutasVermelhas1=frutasVermelhas.map(fruta => fruta.toUpperCase());
6 let todasFrutas=[...frutasAmarelas,...frutasVermelhas1,...
  frutasVerdes];
7 console.log(todasFrutas);
```

Linhas 1,2 e 3 cria arrays

Linha 5 - cria uma nova variável com base no array da linha 2, mas passando os elementos para letra maiúscula na nova variável

Linha 6 - cria um novo array usando o array da linha 1 o novo array criado em letra maiúscula e mais o array da linha 3



✗ 8 - Explique os principais métodos para o trabalho de LEITURA E ESCRITA .../1  
DE ARQUIVOS. \*

`readFileSync()` - método nativo do file system, serve para recuperar dados, importante adicionar o segundo parâmetro (utf-8) para declarar o tipo de encoding

`JSON.parse()` - passa os dados em formato json a um objeto literal para poder ser manipulado no js

`writeFileSync()` - permite escrever arquivos, com dois parâmetros, um o arquivo aonde queremos escrever e o segundo o conteúdo que queremos escrever.

Se o arquivo informado não existir, esse método o cria, porém se o arquivo já existir ele apaga o conteúdo existente e adiciona os novos conteúdos.

`appendFileSync()` - praticamente idêntico ao `writeFileSync`, a única diferença é que se o arquivo informado já existir, ele adiciona os novos componentes sem apagar os já existentes

.....



✓ 9 - Qual o código correto para o cálculo do fatorial de um número? \* 1/1

O fatorial é representado pelo símbolo "!". Definimos como  $n!$  ( $n$  fatorial) a multiplicação de  $n$  por todos os seus antecessores até chegar em 1.  $n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$ .

```
let valor=5;
let fatorial=1;
for (let i = valor; i > 1; i--) {
  fatorial*=i;}
console.log("Fatorial = "+fatorial);
```

☒ Opção 1



```
let valor=5;
let fatorial=1;
for (let i = valor; i > 1; i--) {
  fatorial=i;}
console.log("Fatorial = "+fatorial);
```

☐ Opção 2

```
let valor=5;
let fatorial=1;
for (let i = 0; i <=5; i++) {
  fatorial*=i;}
console.log("Fatorial = "+fatorial);
```

☐ Opção 3

```
let valor=5;
let fatorial=1;
for (let i = valor; i <=5; i++) {
  fatorial*=i;}
console.log("Fatorial = "+fatorial);
```

☐ Opção 4

```
let valor=5;
let fatorial=1;
for (let i = valor; i <=5; i++) {
  fatorial+=i;}
console.log("Fatorial = "+fatorial);
```

☐ Opção 5



✓ 10 - Os métodos `.readFileSync()`, `.appendFileSync()`, `.parse()` e `.writeFileSync()` respectivamente são utilizados para: \*

1/1

- ☐ Ler um arquivo, adicionar conteúdo substituindo o já existente, transformar objetos literais em Json, adicionar conteúdo ao já existente.
- ☐ Transformar objetos literais em Json, adicionar conteúdo ao já existente, ler um arquivo, adicionar conteúdo substituindo o existente.
- ☐ Adicionar conteúdo substituindo o já existente, ler um arquivo, transformar dados Json em objetos literais, adicionar conteúdo ao já existente.
- ☒ Ler um arquivo, adicionar conteúdo ao já existente, transformar dados Json em objetos literais, adicionar conteúdo substituindo o já existente. ✓
- ☐ Transformar objetos Json em literais, adicionar conteúdo ao já existente, ler um arquivo, adicionar conteúdo substituindo o já existente.

✓ 11 - Em relação às CLOSURES (escopo ou escopo léxico) é INCORRETO afirmar: \*

1/1

- ☐ São funções dentro de outras funções
- ☐ São funções que ficam aninhadas
- ☐ São funções que se alojam em containers
- ☐ São funções onde temos a relação de pai e filho
- ☒ São funções representadas por arrow function ✓



- ✗ 12 - De acordo com o código abaixo, escolha a opção que mostre os valores das variáveis fruta1, fruta2, fruta3 respectivamente e o conceito utilizado no script. \* 0/1

```
let frutasAmarelas=['Melão','Mamão','Limão'];  
let [fruta1, fruta2, fruta3]= frutasAmarelas;  
  
console.log(fruta1);  
console.log(fruta2);  
console.log(fruta3);  
console.log(frutasAmarelas);
```

- ☐ Limão, Mamão e Melão, conceito: união
- ☐ Melão, Mamão e Limão, conceito: união
- ☐ Melão, Mamão e Limão, conceito: desestruturação
- ☐ Mamão, Melão e Limão, conceito: desestruturação
- ☒ Melão, Mamão e Limão, conceito: atribuição

✗

Resposta correta

- ☒ Melão, Mamão e Limão, conceito: desestruturação





- ✓ 13 - Observando o código abaixo, escolha a opção que exiba corretamente o valor das variáveis: \*

1/1

```
let x=5;  
let y = x++ + ++x;  
console.log("y= "+y);  
x=3;  
y=x*(x + 1)*x++;  
x=5;  
y=3;  
y*= x+1  
console.log("x= "+x);  
console.log("y= "+y);
```

- ☐ 11, 6, 17
- ☐ 7, 5, 17
- ☒ 12, 5 e 18
- ☐ 12, 10, 18
- ☐ 12, 13, 14



✓ 14 - Numa construção de um IF sem o ELSE, o que acontece se o teste da condição do IF retornar FALSE? \*

- ☐ Dá erro?
- ☐ Não acontece nada?
- ☐ O corpo do if é executado
- ☒ A instrução abaixo do fechamento do if é executada.
- ☐ Dá uma resposta false



✓ 15 - Após a execução do código abaixo, o que deve ser impresso: \*

1/1

```
for (var i=0; i<50; i+=10){  
    console.log(i);  
}  
console.log(i);
```

- ☐ 0 10 20 30 40
- ☒ 0 10 20 30 40 50
- ☐ 10 20 30 40 50
- ☐ 0 10 20 30 40 50 60
- ☐ 0 50



✓ 16 - Qual o nome de variável que está correto? \*

1/1

- ☐ 1celular
- ☐ @email
- ☒ nomePessoa
- ☐ #end\_comercial
- ☐ end residencial



✓ 17 - Uma estrutura de if ternário como vista abaixo, substitui qual estrutura? \*

1/1

```
let x=50;  
let resto=50%2;  
resposta = resto==0 ? 'Par': 'Impar';  
console.log(resposta);
```

- ☐ for
- ☐ if
- ☒ if else
- ☐ else
- ☐ switch case



- ✓ 18 - Observando o código abaixo, escolha a alternativa que mostra o que 1/1 cada função fez , ou seja, o que fez o `replace()`, `toLocaleUpperCase()`, `split()` e o `join()` respectivamente. Vale ressaltar que as saídas são: D-I-G-I-T-A-L-H-O-U-S-E e C-T-D \*

```
function soletrar(texto){  
    console.log(texto.replace('-', '').  
        toLocaleUpperCase().split('').join  
        ('-'));  
}  
soletrar("digital-house");  
soletrar("ctd");
```

- ☐ Substitui cada hífen por "", transforma o texto em letras minúsculas, transforma a string em array, junta o array com o separador hífen
- ☒ Substitui cada hífen por "", transforma o texto em letras maiúsculas, transforma a string em array, junta o array com o separador hífen ✓
- ☐ Junta o array com o separador hífen, substitui cada hífen por "", transforma o texto em letras minúsculas, transforma a string em array,
- ☐ Transforma o texto em letras maiúsculas,junta o array com o separador hífen,substitui cada hífen por "",transforma a string em array
- ☐ Transforma a string em array,junta o array com o separador hífen,substitui cada hífen por "",transforma o texto em letras maiúsculas



✓ 19 - Sobre o Node é correto afirmar: \*

1/1

- ☐ O Node é um compilador da linguagem Javascript que possibilita o trabalho com front-end.
- ☐ O Node é um compilador da linguagem Javascript.
- ☒ O Node é um interpretador da linguagem Javascript, de código aberto, que possibilita a execução de códigos da linguagem do lado do servidor. ✓
- ☐ O Node é um interpretador de código privado.
- ☐ O Node permite somente o trabalho com o back-end.

✓ 20 - A sigla NPM significa: \*

1/1

- ☐ Never Program More
- ☐ Node Program Manager
- ☐ Node Package Monitor
- ☒ Node Package Manager ✓
- ☐ Node Progress Manager

Este formulário foi criado em Digital House.

Google Formulários

