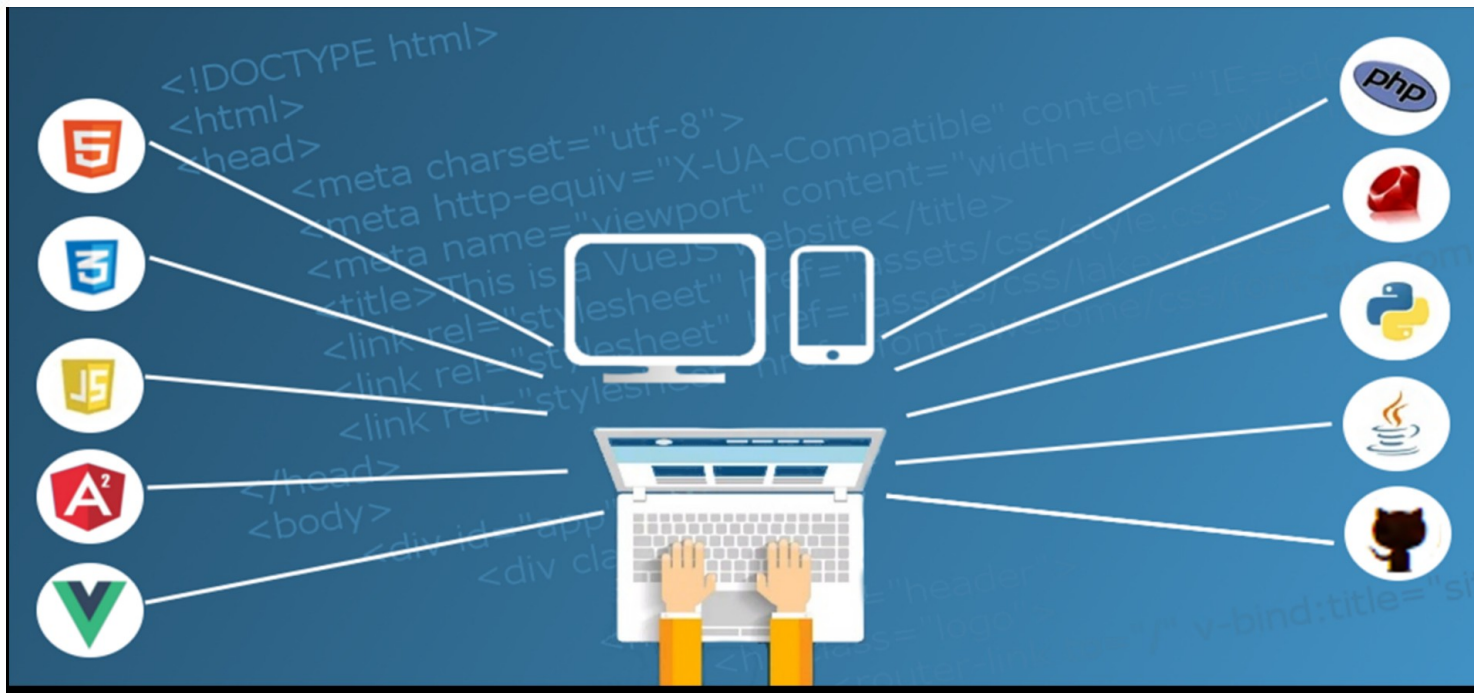


Programação Front-end





JavaScript

A linguagem - Função



```
function name([param,[, param,[..., param]]) {  
    [statements]  
}
```

- ✓ **Name:** O nome da função.
- ✓ **Param:** O nome de um argumento a ser passado para a função. Uma função pode ter até 255 argumentos.
- ✓ **Statements:** As instruções que compõem o corpo da função.





JavaScript

A linguagem - Funções




- ✓ Funções anônimas;
- ✓ Funções de primeira classe,
- ✓ Funções de alta ordem,
- ✓ Funções aninhadas
- ✓ Função recursivas e
- ✓ Fechamentos (closures).





JavaScript

A linguagem - Funções

- 
- ✓ **funções de primeira classe**,
quando funções podem ser tratadas com valores que podem ser passados, manipulados, retornados, este conceito é um atributo da função.

`int soma (int num1, int num2); // função de primeira classe em C`

`var resultado=soma(var num1,var num2); //JavaScript`

`return (num1, num2); // instrução final da função soma JS`



JavaScript

A linguagem - Funções



- ✓ **funções Anonimas,**
quando funções não contêm nomes e são executadas imediatamente, não podendo ser reutilizadas ...

```
function () {  
  // código  
  // código  
  // código  
  // código  
}
```

```
var nomeVAR = function () {  
  // código  
  // código  
  // código  
  // código  
  return ( valor)  
}
```





JavaScript

A linguagem - Funções



- ✓ **Função de alta ordem.**
é uma função que recebe ou retorna uma função. Só é possível definir uma função de alta ordem, se a linguagem tem funções de ***primeira classe***, pois é necessário que funções sejam "passáveis", conceito de atributo da função.

`source.addEventListener(eventName, eventHandler);`

`Botao.addEventListener('click',function(){}); //JavaScript`



JavaScript

A linguagem - Funções



Funções aninhadas,

Quando funções também podem ser criadas dentro de outras funções, criando vários níveis de “localidades”.

```
var landscape = function() {  
  var result = "";  
  var flat = function(size) {  
    for (var count = 0; count < size; count++)    result += "_";  
  };  
  var mountain = function(size) {  
    result += "/";  
    for (var count = 0; count < size; count++)    result += "°";  
    result += "\\°";  
  };  
  flat(3);  
  mountain(4);  
  flat(6);  
  return result;  
};
```

JavaScript

A linguagem - Funções



Funções aninhadas,

Quando funções também podem ser criadas dentro de outras funções, criando vários níveis de “localidades”.

```
alice = {  
  name: "Alice"  
}
```

```
eve = {  
  name: "Eve",  
  talk: function(greeting) {  
    console.log(greeting + ", my name is " + this.name);  
  }  
}
```

//saida

```
eve.talk("yo");  
eve.talk.apply(alice, ["hello"]);
```

```
yo, my name is eve  
hello, my name is alice
```


JavaScript

A linguagem - Funções



Funções recursivas,

Uma função recursiva é uma função que se refere a si própria.
função factorial:

```
function rFact(num)
{
    if (num == 0)
        { return 1; }
    else
        { return num * rFact( num - 1 ); }
}
```

```
function sFact(num)
{
    var rval=1;
    for (var i = 2; i <= num; i++)
        rval = rval * i;
    return rval;
}
```



JavaScript

A linguagem



Fechamento: (closures)

Closure é a forma de fazer com que as variáveis dentro de uma função sejam privadas e persistentes.

“Baseado nas variáveis e blocos de escopo definidos, definido em tempo de compilação.”

```
function ModuloMatematico() {  
  var x = 0;  
  function somaUm() {  
    x++;  
    console.log(x);  
  }  
  function subtraiUm() {  
    x--;  
    console.log(x);  
  }  
  return {  
    somaUm: somaUm,  
    subtraiUm: subtraiUm  
  };  
}  
  
var teste = ModuloMatematico();  
teste.somaUm(); // 1  
teste.somaUm(); // 2  
teste.somaUm(); // 3  
teste.subtraiUm(); // 2
```





JavaScript

<https://www.mundojs.com.br/2020/04/22/javascript-lidera-ranking-de-linguagens/#page-content>





JavaScript

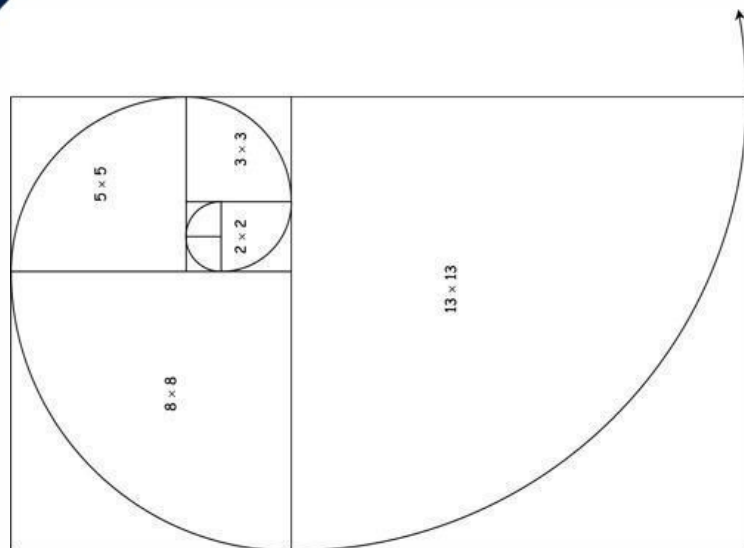


Exercicio – Programa que Calcula a media de 4 notas (box text) e mostra mensagem se o aluno foi aprovado ou reprovado.
(fazer em versão com Alert e versão de nova pagina – Document.write)



JavaScript

Exercicio – fazer um programa que calcula a série de Fibonacci até um numero digitado com termino



“A sequencia ou sucessão de Fibonacci, segundo a matemática, é a ordem de números inteiros que parte, geralmente, de 0 e 1 e que cada número subsequente representa a soma dos dois anteriores. Enigmaticamente, essa sequencia está presente em diversos fenômenos da natureza.”

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584 ...

$$F_n = F_{n-1} + F_{n-2}$$

e valores iniciais correspondentes a: $F_1 = 1$; $F_2 = 1$.



JavaScript

Exercício – calculando o consumo de combustível

Efetue o cálculo da quantidade de litros de combustível gastos em uma viagem, utilizando um automóvel que faz 12 km por litro.

Para obter o cálculo, o usuário deve fornecer o tempo gasto (**tempo**) e a velocidade média (**velocidade**) durante a viagem.

Desta fórmula será possível obter a distância percorrida com a fórmula (**distância = tempo * velocidade**).

Possuindo o valor da distância, basta calcular a quantidade de litros de combustível utilizada na viagem utilizando a fórmula (**litros_usados = distância/12**).

Ao final, o algoritmo deve apresentar os valores da velocidade média (velocidade), tempo gasto na viagem (tempo), a distância percorrida (distância) e a quantidade de litros (litros usados) utilizados na viagem.





JavaScript

Exercicio – Número de azulejos

Faça um código que leia as seguintes informações:

- ✓ Altura e Largura de uma parede;
- ✓ Altura e Largura de azulejos;

Calcule e informe a quantidade de azulejos necessários para cobrir a parede.

Segunda parte: se o faltante de pedaço de azulejo for menor que a metade – altura ou largura – o mesmo azulejo pode ser utilizado duas vezes.



JavaScript

Exercicio – Evento click

Implemente um código que, leia valores para a, b e c, e calcule x' e x'' usando a fórmula de Bhaskara:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Exemplo: no conjunto de teste ($x^2 - 2x + 1 = 0$), os valores serão a = 1, b = -2 e c = 1.

Conjunto de Testes:

$$\begin{aligned}x^2 - 2x + 1 &= 0 \\2x^2 - 4x + 2 &= 0 \\3x^2 + 5x - 9 &= 0 \\2x^2 + x - 3 &= 0 \\x^2 + x + 7 &= 0\end{aligned}$$

Respostas:

$$\begin{aligned}x' &= 1; x'' = 1 \\x' &= 1; x'' = 1 \\x' &= 1.08; x'' = -2.75 \\x' &= 1; x'' = -1.5 \\&\text{Sem raiz}\end{aligned}$$