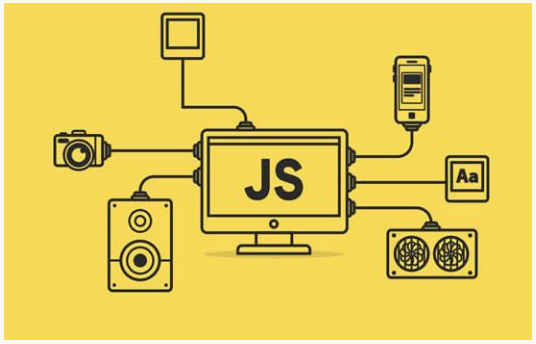


Programação WEB

Franciele Petry

UNOESC

Programação Web

Formas de uso:

Dentro próprio código HTML:

```
<a href="#" onclick="alert('alô mundo!')">Diga alô</a>
```

Separado em uma tag de script (preferencialmente dentro da tag <head></head>):

```
<script type="text/javascript">
        alert("alô mundo");
</script>
```

Mais separado ainda dentro de um arquivo “texto” com extensão .js sendo chamado por uma tag script:

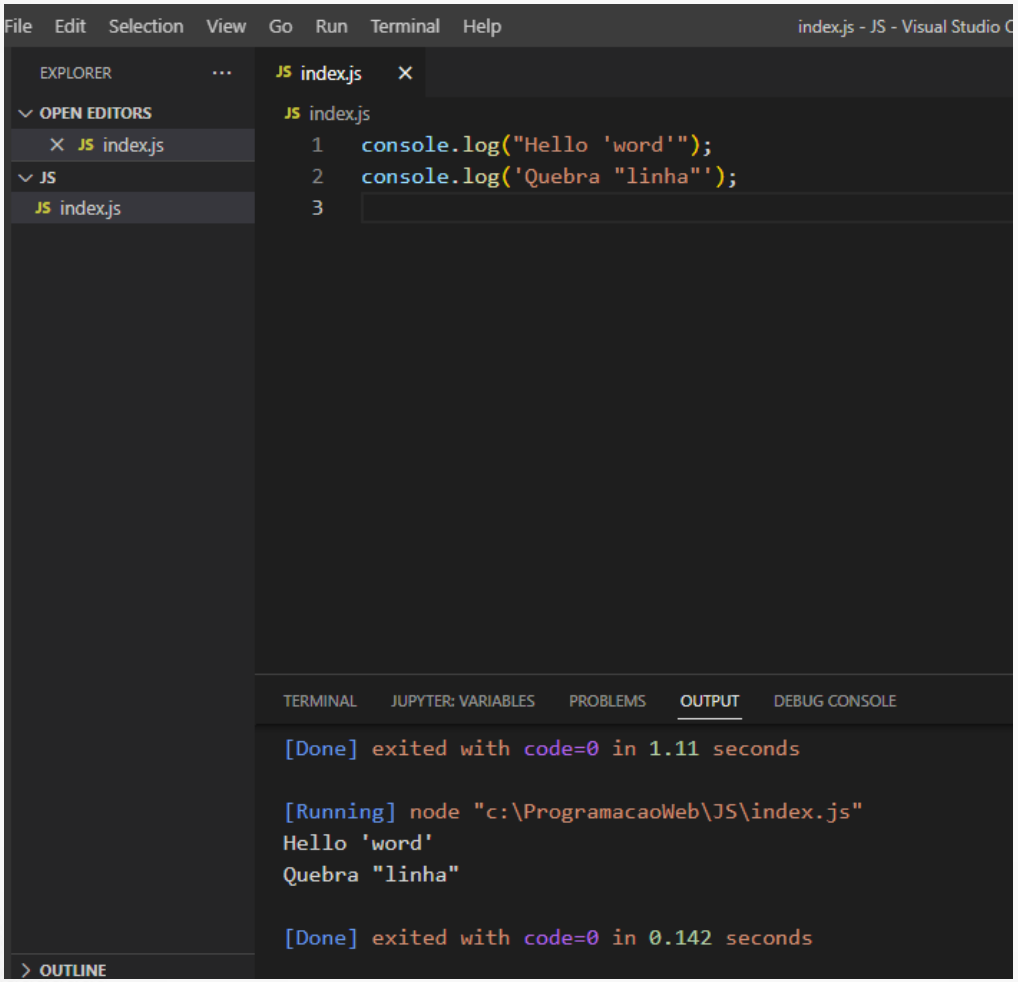
```
<script type="text/javascript" src="script.js"></script>
```

Programação Web

Usaremos dois arquivos texto:

- **Um com HTML com extensão .html**
- **Outro com JavaScript com extensão .js**
- **Haverá ainda uma tag HTML que "unirá" os arquivos**

Programação Web



Programação Web

index.js

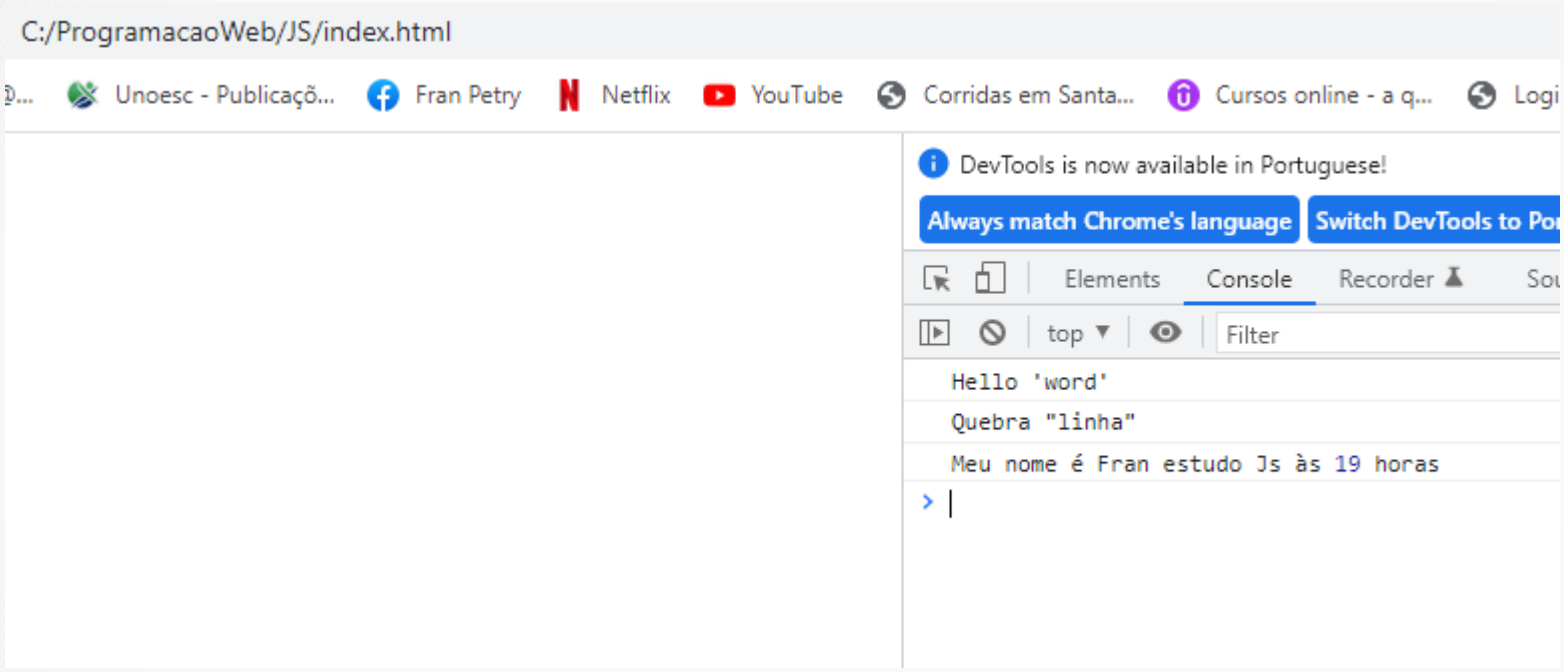
```
1  console.log("Hello 'word'");
2  console.log('Quebra "linha"');
3  //comentário de uma linha
4  console.log('Meu nome é Fran estudo Js às', 19, 'horas'); // posso concatenar valores
5
6  /* comentario de bloco */
7
```

index.js | index.html

index.html

```
1  <html>
2      <head>
3      <meta  charset="UTF-8">
4      <title> Página em HTML </title>
5      </head>
6
7  <body>
8
9  <script src="index.js"></script>
10 </body>
11 </html>
```

Programação Web



Programação Web

```
let nome; //declarou a variável
nome = "Franciele";
console.log(nome);
//não posso começar variáveis com números
//usar variáveis significativas
//não pode conter espaços e traços
//de preferencia começa com letra minúscula nomeSobrenome
//Js é case sensitive
// Não utilize Var, utilize let
```

```
const sobrenome =  "Petry";
console.log(sobrenome);
//não posso começar constantes com números
//usar variáveis significativas
//não pode conter espaços e traços
//utilizamos camelcase
// Não utilize Var, utilize const
```

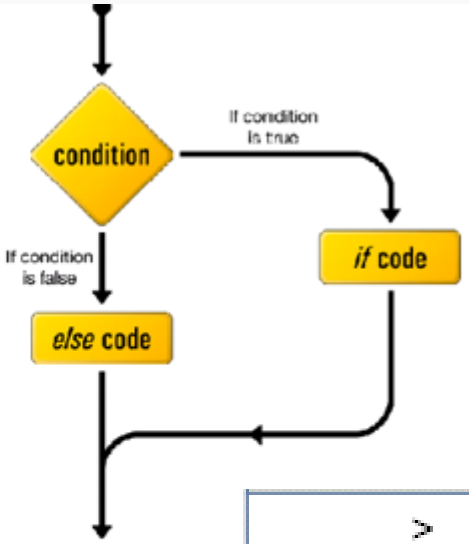
Programação Web

```
const idade = 29;
const peso = 70;
const altura = 1.63; // constante não muda de valor
let indiceMassaCorporal; // variável muda de valor ao decorrer do tempo

indiceMassaCorporal = peso/(altura * altura)

console.log("IMC = ", indiceMassaCorporal)
```

Programação Web – If Else

Estrutura de Decisão

```
if (condição) {
    código da condição verdadeira;
}
else {
    código da condição falsa;
}
```



{ simboliza um início/begin

} representa um fim/end

&& : and
|| : or
! : not

>	A > B
>=	A >= B
<	A < B
<=	A <= B
==	A == B
!=	A != B

Programação Web

Estrutura de Decisão

```
const idade = 16;
if (idade >= 16 && idade < 18) {
    console.log("voto facultativo");
}
```

Operador	Descrição
. [] ()	Acesso a propriedades, indexação, chamadas a funções e sub-expressões
++ — - ~ ! new delete typeof	Operadores unários e criação de objetos
* / %	Multiplicação, divisão, módulo
+ -	Adição, subtração, concatenação de strings
<< >> >>>	Deslocamento de bit
< <= > >= instanceof	Menor, menor ou igual, maior, maior ou igual, instanceof
== != === !==	Igualdade, desigualdade, igualdade estrita, e desigualdade estrita
&	AND bit a bit
^	XOR bit a bit
|	OR bit a bit
&&	AND lógico
||	OR lógico
? :	Operador condicional (ternário)

Programação Web - **switch**

```
switch (expressão) {
  case valor 1:
            //código a ser executado se a expressão = valor 1;
     break;
  case valor 2:
     //código a ser executado se a expressão = valor 2;
            break;
  ...
  case valor n:
            //código a ser executado se a expressão = valor n;
      break;
  default:
    //executado caso a expressão não seja nenhum dos valores;
}
```

Programação Web

```
switch (idade) {
    case (29):
    console.log("Você está no auge.");
    break;
    case (40) :
    console.log("A vida começa aqui.");
    break;
    case (60) :
    console.log("Iniciando a melhor idade.");
    break;
    default:
        console.log("A vida merece ser vivida, não importa a idade.");
    break;
}
```

Programação Web - For

Executa um trecho de código por uma quantidade específica de vezes
Sintaxe:

```
for (inicio; condicao; incremento/decremento) {
                    // código a ser executado.
}
```

```
let numeros = [1, 2, 3, 4, 5];
    for (var i = 0; i < numeros.length; i++) {
            numeros[i] = numeros[i]* 2;
            console.log(numeros[i]);
    }
```

Programação Web - While

Executa um trecho de código enquanto uma condição for verdadeira

Sintaxe:

```
while (condicao) {
        //código a ser executado
}
```

```
let numero = 1;
while (numero <= 5) {
    console.log("O número atual é: " + numero);
    numero = numero + 1;
}
```

Programação Web – do While

Executa um trecho de código enquanto uma condição for verdadeira
Mesmo que a condição seja falsa, o código é executado pelo menos uma vez
Sintaxe:

```
do {
        // código a ser executado.
} while (numero <= 5) ;
```

```
let number = 1;
do {
        console.log("O número atual é: " + number);
        number = number + 1;
} while (number <= 5) ;
```

Programação Web – Try Catch

Erros comuns devem ser tratados/evitados;

Testar para verificar possíveis erros comuns;

Exceções podem ser tratadas com: try, catch e finally;

```
/* Considerando a função do exemplo anterior já declarada */

try {
    lerarr(); //função que pode lançar exceção
} catch (e) {
    // comandos a serem executados em caso de exceção
    window.alert("Oops! No soup for you!");
} finally {
    document.write("ocorreu um erro");
}
```

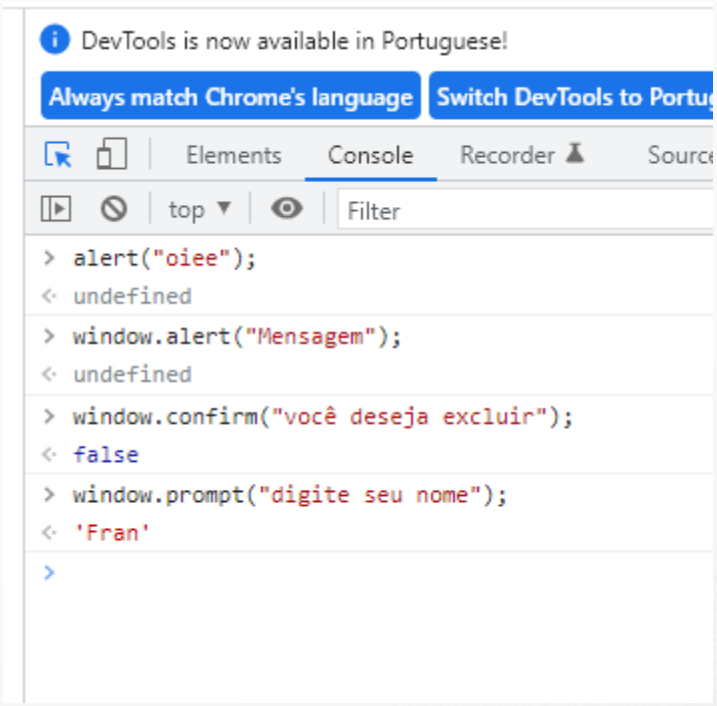
Programação Web – Inputs

inputs.html

```html
<html>
    <head>
    <meta  charset="UTF-8">
    <title> Página em HTML </title>
    </head>

<body>

<script src="inputs.js"></script>
</body>
</html>
```

inputs.js

```js
alert("oi");
```



Programação Web – Funções

Funções são blocos de código reutilizáveis;

Elas não são executadas até que sejam chamadas;

Podem ter parâmetros de entrada e de saída;

Podem ter vários parâmetros de entrada separados por vírgulas;

Podemos retornar um valor através da instrução return.

```
function nomeDaFuncao() {
                        //códigos referentes à função.
                        ...
}
function nomeDaFuncao(p1, p2, p3, ...) {
                        //códigos referentes à função.
                        ...
}
function nomeDaFuncao(p1, p2, p3, ...) {
                        ...
                        return p1+p2-p3;
                        ...
}
```

Programação Web – Funções

```
function saudacao () {
    console.log("bom dia!")
}

saudacao();
```

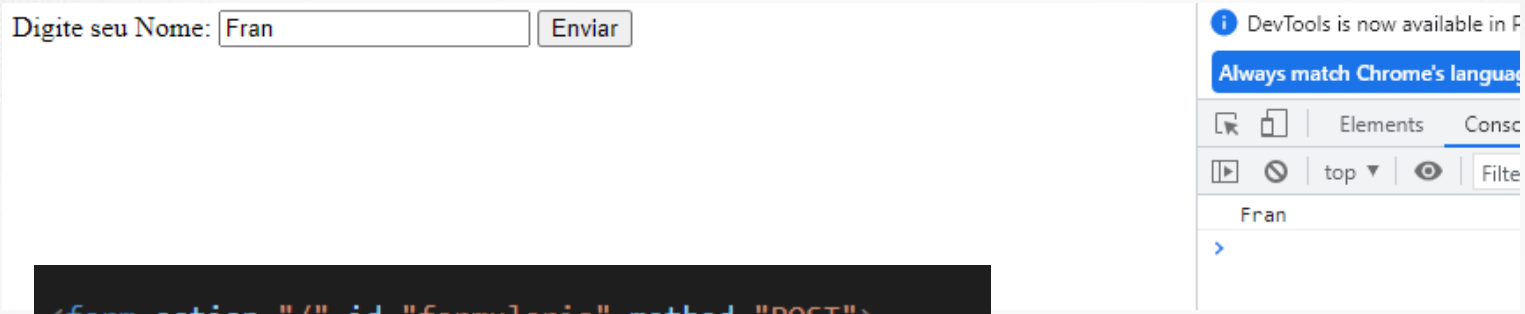
```
/*O uso de crase(acento grave) indica que esta sendo declarado uma
 Template String, mas você pode declarar tanto com aspas,
  quanto aspas duplas ou o acento grave */

function saudacao (nome) {
    console.log(`bom dia ${nome}!`) // use crases
}

saudacao("Fran");
```

Programação Web – Funções

```
function saudacao (nome) {
    console.log(`bom dia ${nome}!`) // use crases
    return 123456;
}

const variavel = saudacao("Fran");
console.log(variavel)
```

```
function soma(x,y){
    const resultado = x+y;
    return resultado;
}

const resultado = soma(1,10);
console.log(resultado);
```

Programação Web – Passando Parâmetros




```html
<form action="/" id="formulario" method="POST">
    <label for="nome">Digite seu Nome:</label>
    <input type="text" id="nome" name="nome">
    <button type="submit">Enviar</button>
</form>

<div id="resultado"></div>
```

```javascript
const form = document.querySelector('#formulario'); //antigamente usava-se getElementById por exemplo

form.addEventListener('submit', function (e) { //adiciona o metodo de submit
    e.preventDefault();

    const nome = e.target.querySelector('#nome');

   console.log(nome.value)
})
```