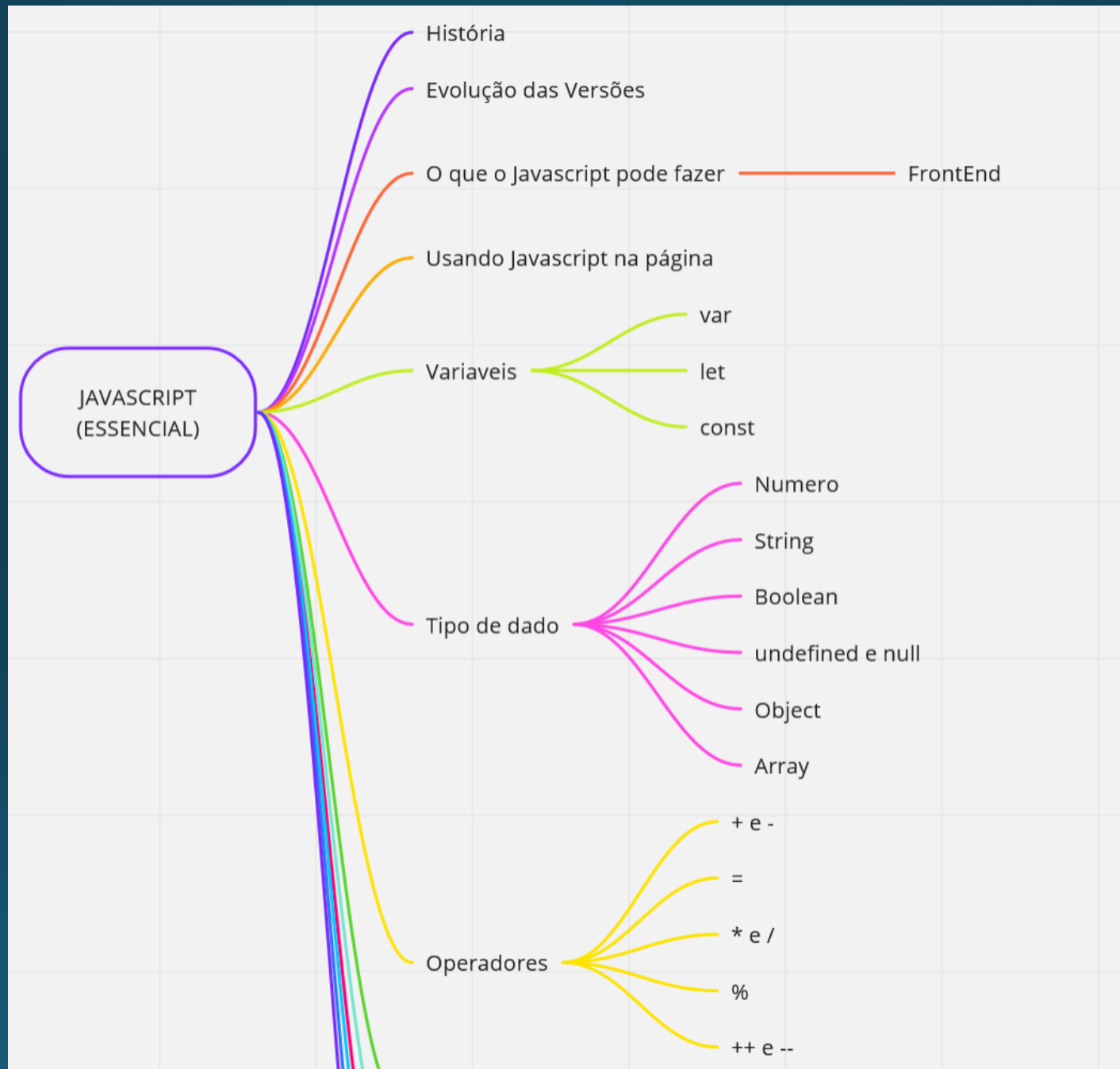
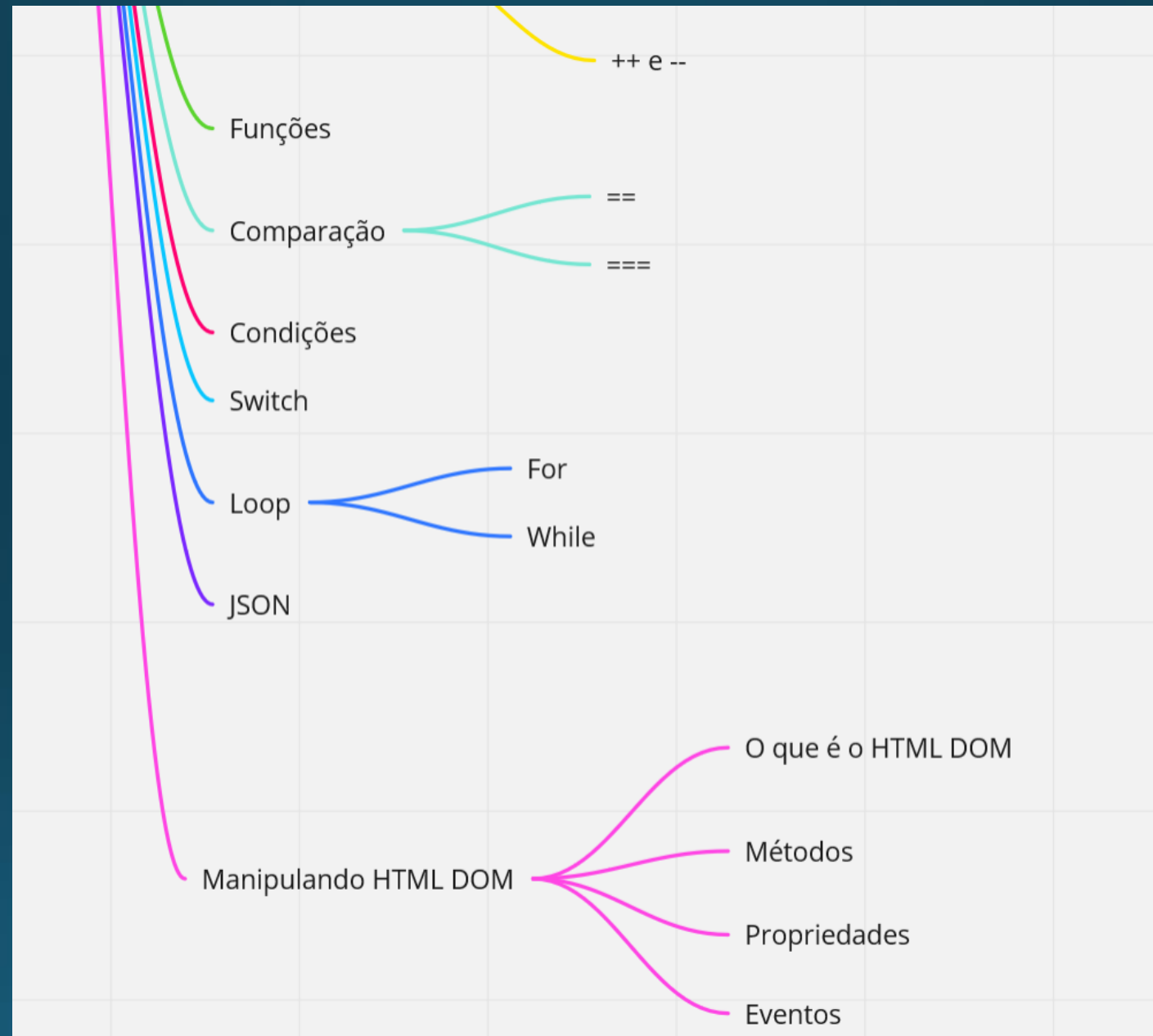


DACOMP – MÊS WEB

JAVASCRIPT







Brendan Eich

HISTORIA



EVOLUÇÃO

ECMA - European Computer Manufacturers Association

Órgão responsável pelo desenvolvimento e manutenção

ECMAScript Versões

Ver	Official Name	Description
1	ECMAScript 1 (1997)	First Edition.
2	ECMAScript 2 (1998)	Editorial changes only.
3	ECMAScript 3 (1999)	Added Regular Expressions. Added try/catch.
4	ECMAScript 4	Never released.
5	ECMAScript 5 (2009) Read More: JS ES5	Added "strict mode". Added JSON support. Added String.trim(). Added Array.isArray(). Added Array Iteration Methods.
5.1	ECMAScript 5.1 (2011)	Editorial changes.
6	ECMAScript 2015 Read More: JS ES6	Added let and const. Added default parameter values. Added Array.find(). Added Array.findIndex().
7	ECMAScript 2016	Added exponential operator (**). Added Array.prototype.includes.
8	ECMAScript 2017	Added string padding. Added new Object properties. Added Async functions. Added Shared Memory.
9	ECMAScript 2018	Added rest / spread properties. Added Asynchronous iteration. Added Promise.finally(). Additions to RegExp.

https://www.w3schools.com/js/js_versions.asp

O que podemos fazer com JavaScript ?

- Mudar a estrutura da página
- Mudar atributos de elementos
- Mudar o estilo dos elementos
- Criar animações

Usando JavaScript no FrontEnd

```
<> index.html x
1  <html>
2
3  <head>
4      <title>Minicurso de Web</title>
5      <link rel="stylesheet" href="./estilo.css">
6
7      <script src="./codigo.js"></script>
8
9      <script>
10
11          alert('Olá JavaScript');
12
13      </script>
14
15  </head>
```

Importa um arquivo externo

Dentro tag <SCRIPT>

VARIÁVEIS

- `var`
- `let`
- `const`

TIPOS DE DADOS

- `number`
- `string`
- `boolean`
- `undefined` e `null`
- `object`
- `array`

OPERADORES

- =
- + e -
- * e /
- %
- ++ e --

FUNÇÕES

COMPARAÇÃO

- ==

- >

- >=

- <

- <=

- !=

CONDIÇÃO

```
if (condição)
```

SWITCH

LOOP

- `for`
- `ARRAY.forEach()`
- `while`

JSON

JavaScript Object Notation

JSON - JavaScript Object Notation

JavaScript Object

<=>

JSON

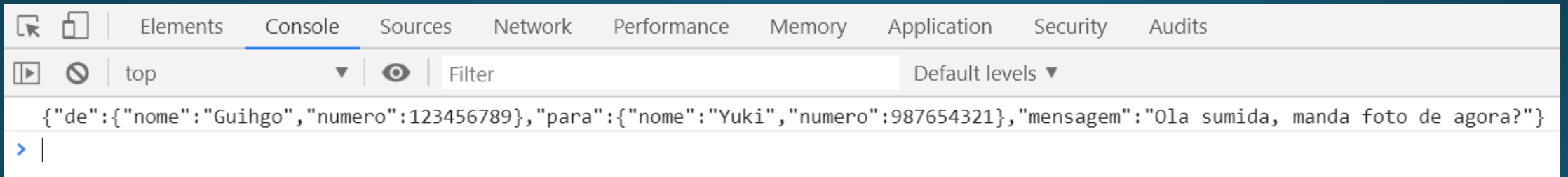
```
var sms = {  
  de: {  
    nome: "Guihgo",  
    numero: 123456789,  
  },  
  para: {  
    nome: "Yukeee",  
    numero: 987654321,  
  },  
  mensagem: 'Ola sumida, manda foto de agora?'  
};
```

```
{  
  "de": {  
    "nome": "Guihgo",  
    "numero": 123456789  
  },  
  "para": {  
    "nome": "Yuki",  
    "numero": 987654321  
  },  
  "mensagem": "Ola sumida, manda foto de agora?"  
}
```

JSON - JavaScript Object Notation

JavaScript Object => JSON

```
13   var smsEmJson = JSON.stringify(sms);  
14  
15   console.log(smsEmJson);
```



JSON - JavaScript Object Notation

JavaScript Object <= JSON

```
var smsEmJson = '{"de":{"nome":"Guihgo",' +  
  '"numero":123456789},"para":{"nome":"Yuki","numero":987654321},' +  
  '"mensagem":"Ola sumida, manda foto de agora?"}';  
  
var smsEmObj = JSON.parse(smsEmJson);  
  
console.log(smsEmObj);
```

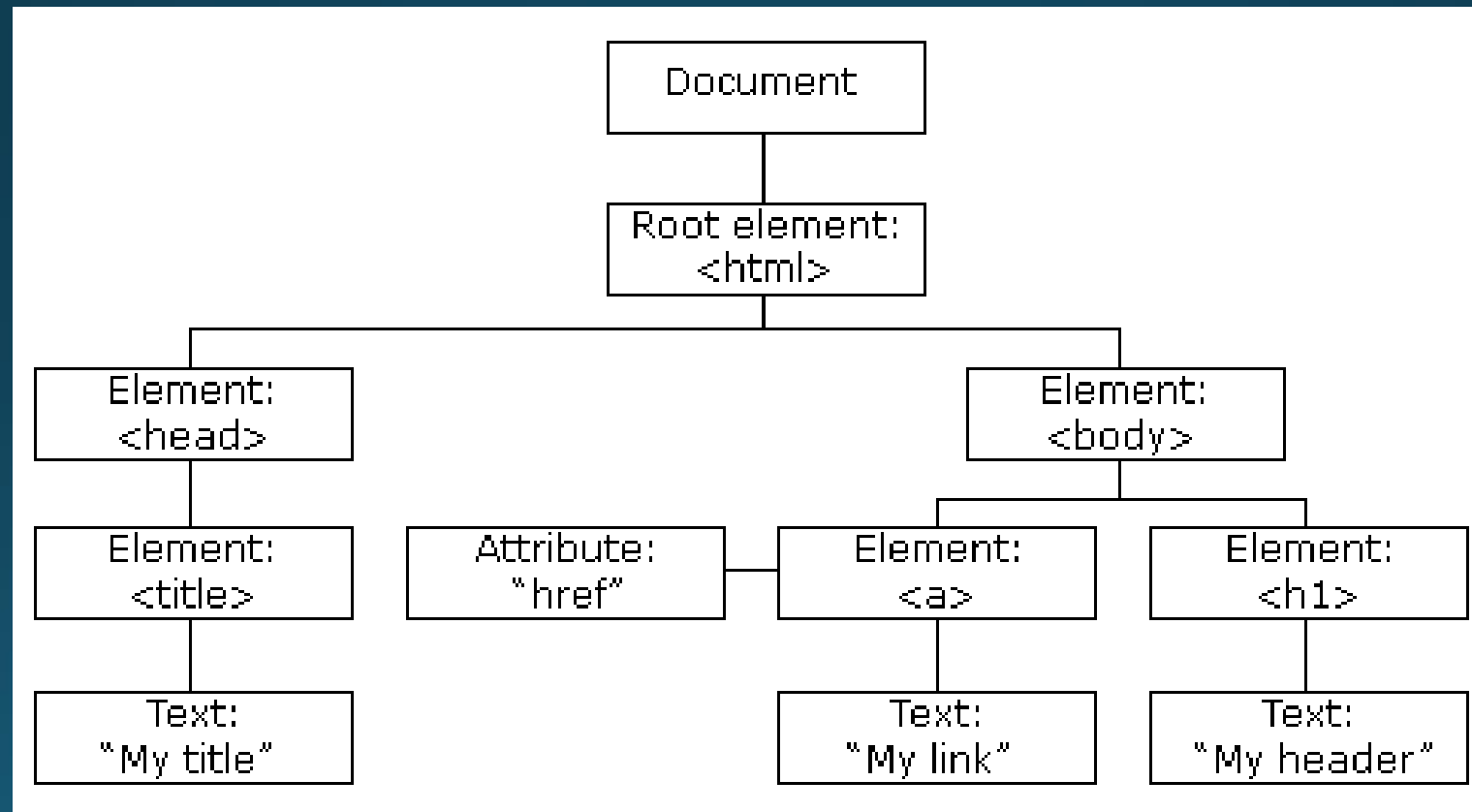
```
▼ {de: {...}, para: {...}, mensagem: "Ola sumida, manda foto de agora?"} ⓘ  
  ► de: {nome: "Guihgo", numero: 123456789}  
    mensagem: "Ola sumida, manda foto de agora?"  
  ► para: {nome: "Yuki", numero: 987654321}
```

HTML DOM

HTML Document Object Model

HTML DOM – O que é

Árvore de Objetos



HTML DOM – Métodos

```
1  document.getElementById(id)
2
3  document.getElementsByTagName(name)
4
5  document.getElementsByClassName(name)
6
7  document.querySelectorAll("<CSS SELECTOR>")
8
9  document.createElement(element)
10
11 document.appendChild(element)
12
13 element.setAttribute(attribute, value)
14
15 element.addEventListener(event, function(){})
```


HTML DOM – Propriedades

```
1    element.innerHTML  
2  
3    element.attribute  
4  
5    element.style.property
```

HTML DOM – Eventos

```
1 element.onclick
```



[Github.com/Guihgo](https://github.com/Guihgo)



guioli.1998@alunos.utfpr.edu.br