

Javascript

Prof. Romerito Campos

Plano de Aula

Objetivo: conhecer a estrutura Lexical da linguagem.

Conteúdo:

- ↗ Estrutura Lexical
- ↗ Tipos
- ↗ Variáveis
- ↗ Estruturas de controle: laços, condicionais

JavaScript

↗ JavaScript é

↗ Alto-nível

↗ Dinâmica

↗ Interpretada

↗ Suporta os estilos Orientado a objetos e Funcional

↗ JavaScript não é Java

JavaScript

- ~ O core da linguagem dá suporte a:
 - ~ tipos números e textuais, arrays, sets e maps;
 - ~ não inclui funcionalidade padrão para **input/output**;
- ~ O ambiente original da linguagem é **Browser**:
 - ~ O input é obtido de mouse, teclado e *requisições HTTP*;
- ~ Desde 2010, após o surgimento do [Node](#), O Javascript pode acessar recursos do sistema operacional (Ler e Escrever arquivos, usar a rede etc).

Estrutura Lexical

Estrutura Lexical - Case sensitive

- ↗ Case sensitive, espaços e quebras de linha (line breaks)
- ↗ Comentários
- ↗ Literais
- ↗ Identificadores e palavras reservadas
- ↗ Unicode
- ↗ Ponto-Vírgula opcionais (*Optional semicolons*)

Estrutura Lexical

~ Variáveis, palavras reservadas e identificadores devem ser escritos de maneira consistente.

```
//definindo variável (explicação a seguir)
let a = 10;
//vai ocorrer um erro!!!
if (A === 10) {
    /// Uncaught ReferenceError: A is not defined
}
```

Fonte: própria.

~ Erro: Uncaught ReferenceError: A is not defined

Estrutura Lexical

↗ Variáveis, palavras reservadas e identificadores devem ser escritos de maneira consistente.

```
//uso do IF ao invés de if  
IF (10 == 10)  
    console.log(10)
```

↗ Erro: Uncaught ReferenceError: IF is not defined

Estrutura Lexical

👉 Comentários: há três formas de comentar o código.

```
// comentário  
  
/*comentário*/  
  
/*  
 * comentário multi-linha  
 *  
 */
```

Estrutura Lexical

↗ Literais: valores que podem ser aplicados diretamente no código

```
12  
1.2  
"Hello World"  
'Hello'  
true  
false  
null
```

Estrutura Lexical

~ Identificadores e palavras reservadas

```
//identificadores
function teste () {
    ///
}
let variavel = 10;
const nome = "romerito";
```

~ teste , variavel e nome são identificadores

~ podemos acessar o conteúdo da variável e da constante pelo seus identificadores

~ podemos executar a função pelo seu identificador: teste()

Estrutura Lexical

~ Um **identificador** é utilizado para nomear constantes, variáveis, funções, propriedades de objetos, classes e etc.

Estrutura Lexical

- ~ Unicode: os programas em JavaScript são escritos utilizando o conjunto de caracteres Unicode.
- ~ Isso permite utilizar símbolos e palavras com acentos na definição de identificadores.

```
//exemplos de uso de Unicode  
let café = "moca";  
caf\u00e9 = "duplo moca";
```

- ~ Abra o console do navegador e digite: `console.log ("\u{1F600}")`

Estrutura Lexical

~ Unicode

~ evite utilizar símbolos especiais ou acentos

~ Javascript não normaliza as definições de identificadores para evitar conflitos

```
//ambas as palavras com o espace \ são café  
const caf\u{e9} = 10;  
const cafe\u{301} = 10;
```

Estrutura Lexical

~> Ponto-Vírgula Opcional (Optional Semicolons)

~> ; é utilizado para separar declarações

~> quebras de linha pode ser tratadas como ;

```
//utilização explícita de ;  
let a = 3;  
let b = 3;
```

```
// JS trata a quebra de linha como;  
let a = 3  
let b = 3
```

Estrutura Lexical

~ Ponto-Vírgula Opcional (Optional Semicolons)

~ ; é utilizado para separar declarações

~ quebras de linha podem ser tratadas como ;

```
let a
a
=
3
console.log(a);
```

~ Este exemplo funciona apenas rodando um script como o código `node script.js` .

~ JavaScript compreende como `let a; a=3; console.log(3)` .

Estrutura Lexical

⚡ Ponto-Vírgula Opcional (Optional Semicolons)

```
function f (a) {  
    return Math.sqrt(a)  
}  
  
let y = 'Resultado: ' + f  
(a+b).toString()
```

⚡ A declaração acima pode indicar duas linhas de código, mas acaba sendo interpretada como uma única linha de código.

⚡ O (aparece como sendo parte da execução de f() . Javascript enxerga f como função: `let y = x + f(a+b).toString()`

Referências

↗ FLANAGAN, D. **JavaScript - The Definitive Guide**. 7. ed. Sebastopol, CA, USA: O'Reilly Media, 2020.