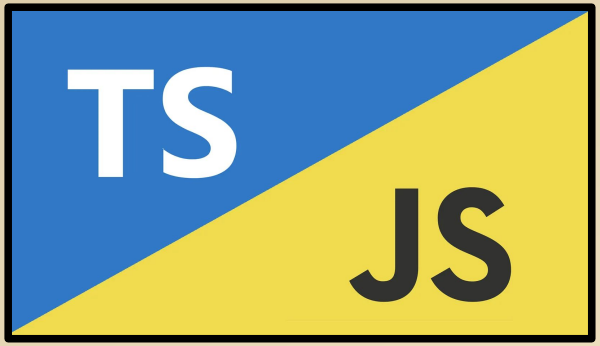
- Introdução
- Ambiente JS
- Linguagem
- Objetos Nativos



JavaScript - Parte 1

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

JavaScript

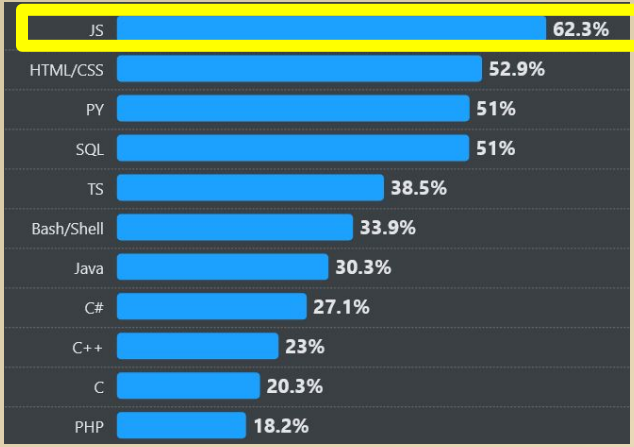
Introdução

- Criado em 4/dezembro/1995
- Inventada por Brendan Eich / Netscape
- É interpretada e sua execução ocorre em: navegadores, ou node.js ou WinJS
- Utilizada por milhões de páginas web para interatividade em documentos HTML
- Não há relação com Java
- Não é seguro para tipos (type-safe)
- Multi-paradigma: protótipos, orientado a objeto, imperativo e funcional

TS JS

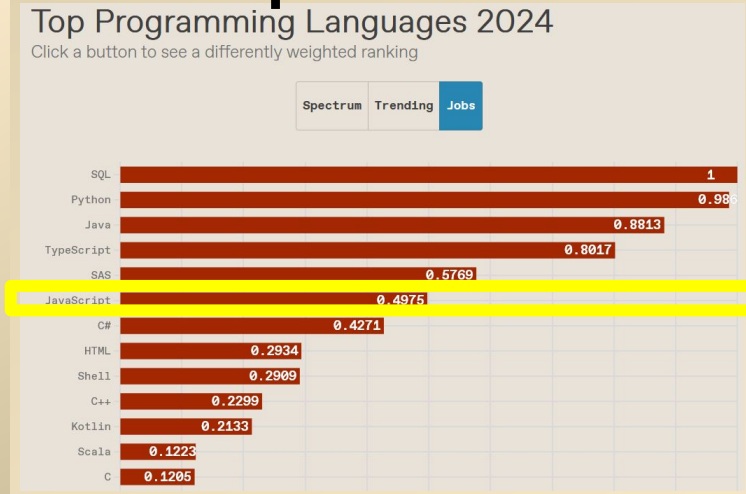
Introdução

✔Stack Overflow/2024



1ª

✔IEEE Spectrum/2024

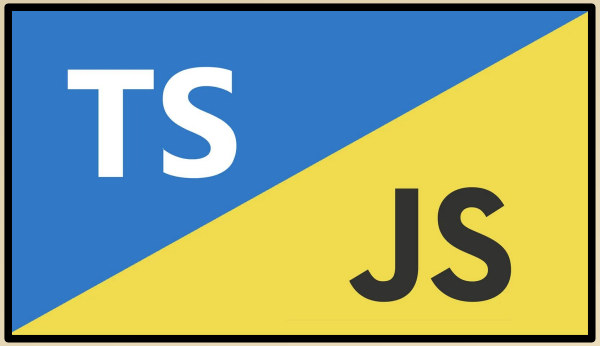


6ª

✔RedMonk/2024

1 JavaScript
2 Python
3 Java
4 PHP
5 C#
6ª TypeScript
7 CSS
7 C++
9 Ruby
10 C
11 Swift

- Introdução
- Ambiente JS
- Linguagem
- Objetos Nativos



JavaScript - Parte 1

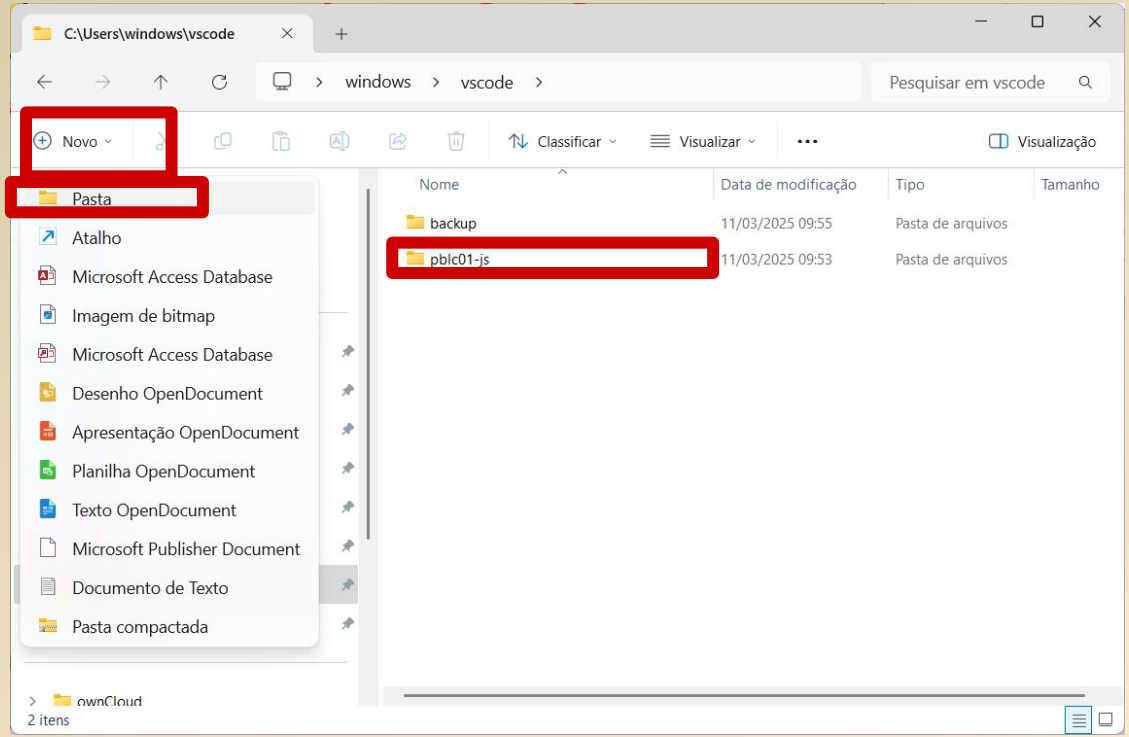
Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

Pasta Ambiente JS/TS

➔C:\Users\aluno\vscode\pblc01js

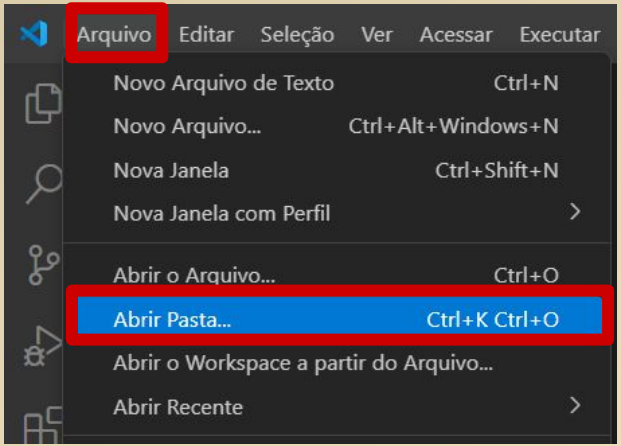
Ambiente

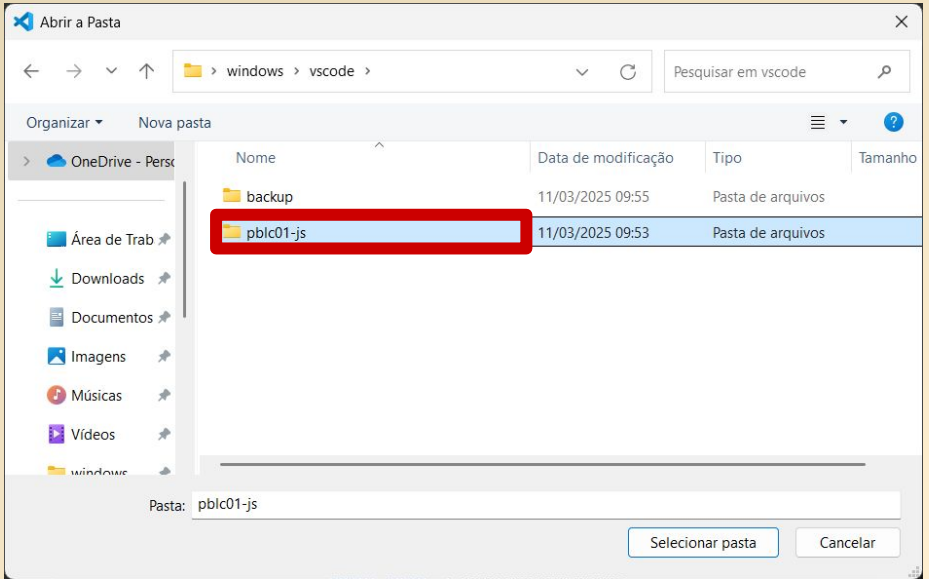


Abrir Pasta VSCode

Ambiente

➔ C:\Users\aluno\vscode\pblc01js





NPM-Node Package Manager

- Ferramenta do Node.js para o gerenciamento de pacotes:
- Cria ambiente isolado para aplicação
- Instala/Desinstala pacote

Ambiente

Isolamento de Ambiente

Ambiente

- Arquivo package.json com relação pacotes do ambiente.
- Para criar o package.json abra o terminal e digite:

npm init -y

- Para abrir o terminal clique no menu:

  Ver | Terminal ou [Ctrl]+[’]

TS JS

Instalando Pacote prompt-sync

Ambiente

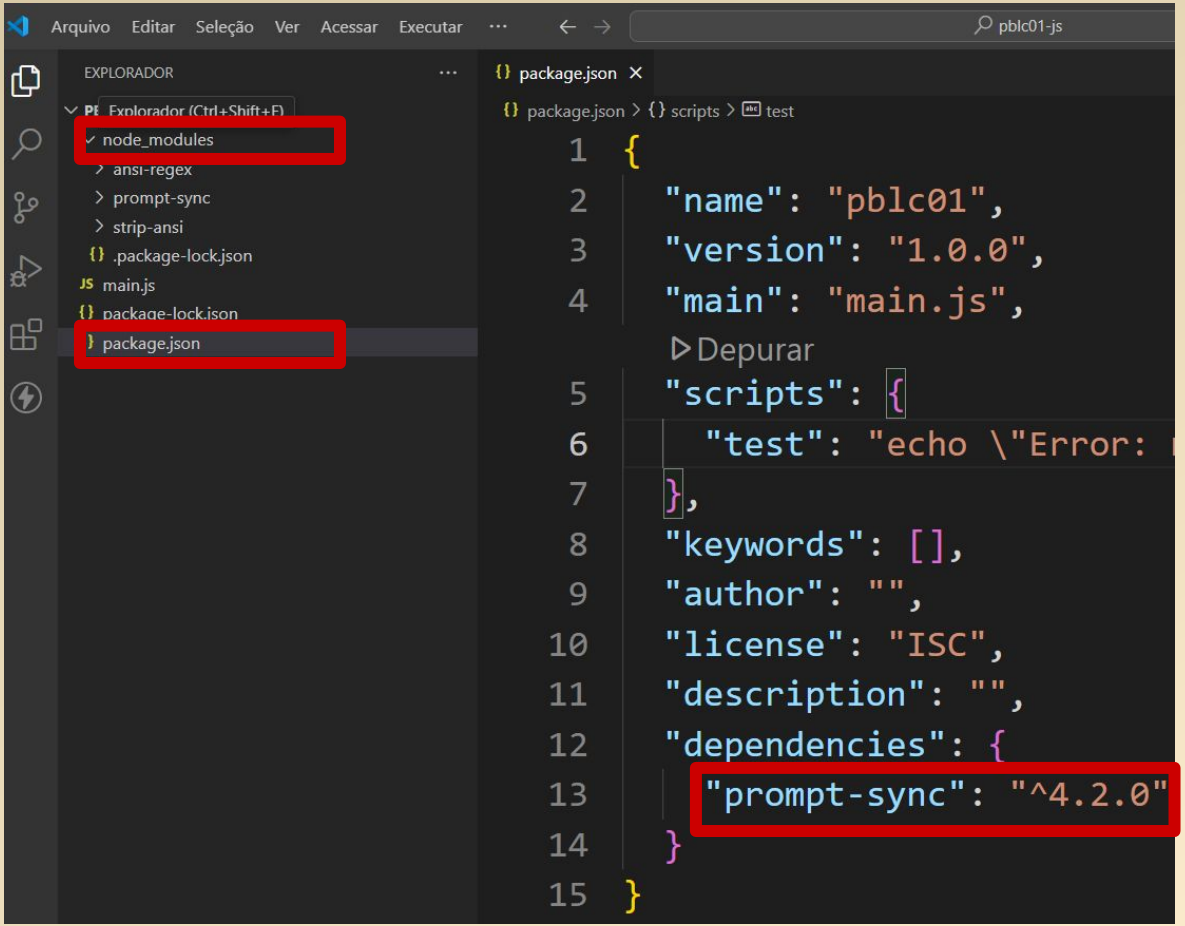
- Para instalar pacotes de dependências o pacote prompt-sync abra o terminal e digite:

  npm install prompt-sync

- Os pacotes de dependências são guardados no diretório node_modules

Instalando Pacote prompt-sync

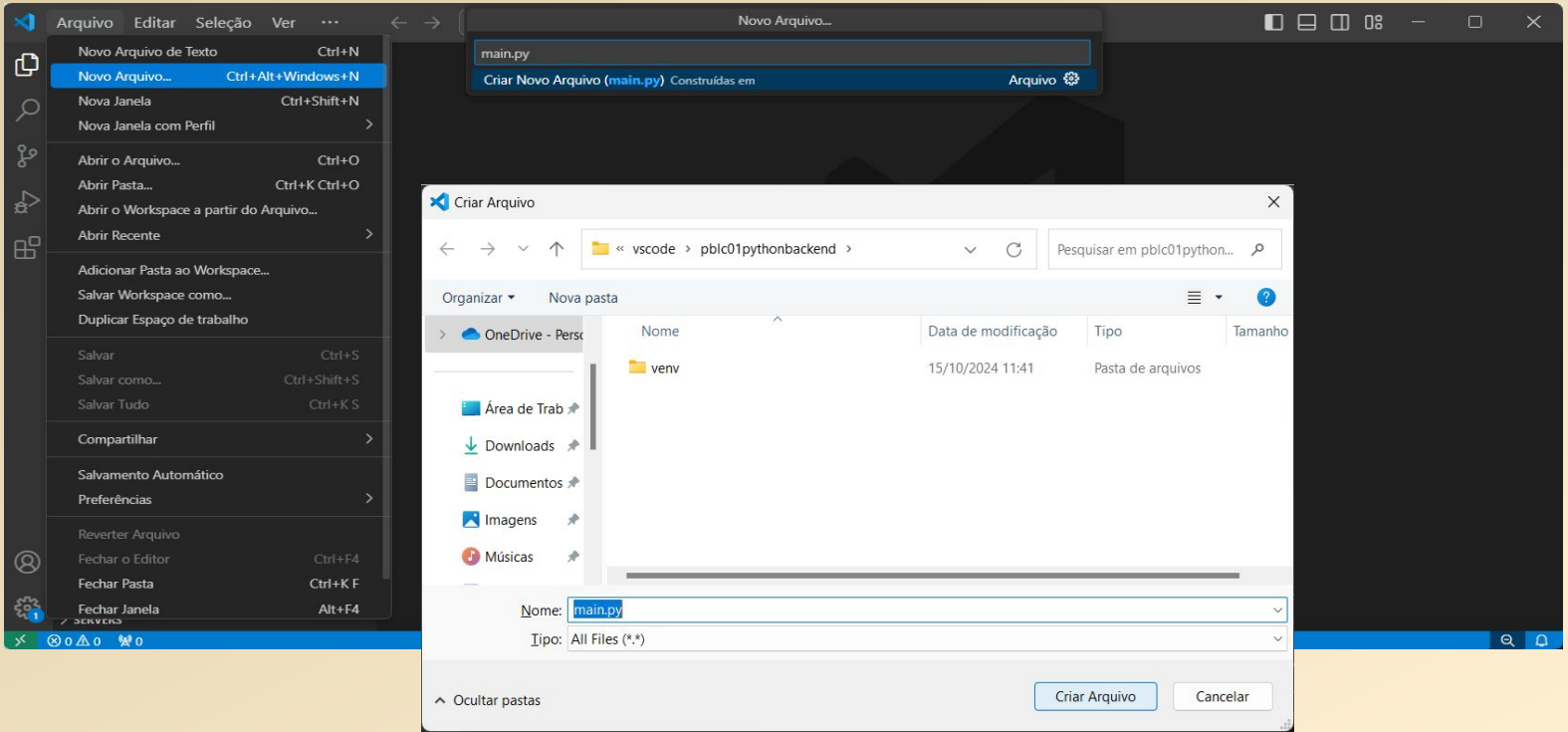
Ambiente



main.py

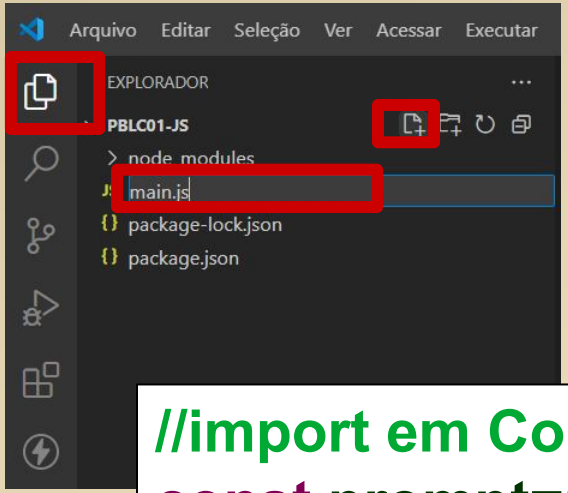
- Arquivo | Novo Arquivo
- main.py

Ambiente



Criando arquivo main.js

Ambiente




```
//import em CommonsJS
const prompt=require("prompt-sync")();

const nome=prompt("nome: ");
console.log(nome)
```

Executar

→ Executar o javaScript

node .\main.js

Ambiente

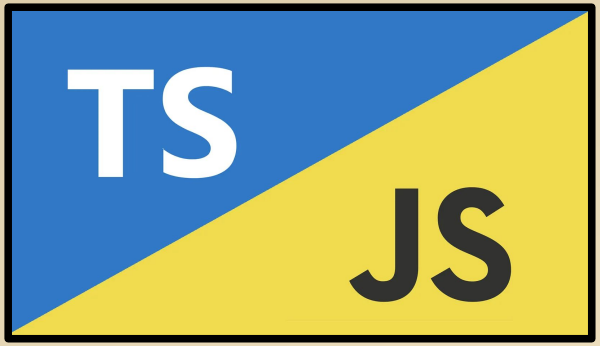
TS JS

Proteção

Ambiente

- Interpretação expõe codigo fonte
- Técnicas para “proteger” o código:
  - Obfuscação
    - obfuscator.io
    - github.com/javascript-obfuscator
  - Compressão
    - dean.edwards.name/packer
- Técnicas para “desproteger” código
  - Embelezador
    - beautifier.io

- Introdução
- Ambiente JS
- Linguagem
- Objetos Nativos



JavaScript - Parte 1

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

JavaScript

Linguagem

- Inicialmente sintaxe similar a C
- Diferencia maiúsculas e minúsculas (*case sensitive*)
- Toda instrução é finalizada com ";"
- Trata eventos
- Comentário → // (única linha)
- /* ... */ (multiplas linhas)
- Pode adicionar dinamicidade ao HTML

TS JS

Variáveis

Linguagem

- **Variáveis locais (var)** estarão disponíveis dentro da função de onde são declaradas
- **Variáveis globais (var)** estarão disponíveis em qualquer código no documento atual e são declarada de fora da função
- **Variável de escolo (let)** estarão disponíveis no escopo do bloco atual
- É opcional nas Variáveis iniciar com valor
- **Constante (cons)** estarão disponíveis na função ou global, o valor é atribuído na declaração e não pode ser mudado

TS JS

Variáveis

Linguagem

- São espaços de memória onde podemos guardar uma informação, como um número ou uma cadeia de caracteres

```
var a=1;
var b=5;
var c=6;
let valor=1.20;
const razao="unifei";
var cidade="Itajuba";
```

TS JS

Variáveis podem assumir

Linguagem

- Não exige a declaração de tipos de variáveis
- String → quase qualquer valor entre aspas simples ou aspas duplas;
- Numérico → números;
- Booleano → true ou false
- Array → []
- Objeto → new ou factory
- Função → usado também para simular orientação objetos

TS JS

Variáveis

Linguagem

- null ausência intencional de valor
- undefined nunca houve valor atribuído

Entrada e Saída de Dados

Linguagem

- Impressão de valor

```
console.log("erro de lógica");
```

- Entrada de dados pelo teclado

```
const prompt=require('prompt-sync')();
var nome = prompt("digite nome: ");
```

TS JS

if ... else

Linguagem

```
var n1 = 5, n2 = 7;
if (n1 == n2) {
  console.log("n1 igual a n2");
} else if (n1 > n2) {
  console.log("n1 maior que n2");
} else {
  console.log("n2 maior ou igual a n1");
}
```

TS JS

switch … case … default

```
const prompt=require("prompt-sync")();
var op = prompt("Escola [1-impar/2-par] ?");
switch(op){
  case "1" :
    console.log("impar");
    break;
  case "2" :
    console.log("par");
    break;
  default:
    console.log("opção inválida!");
    break;
}
```

Linguagem

TS JS

while

Linguagem

```
const prompt=require("prompt-sync")();

var num = parseInt(prompt("Numero:"));
var divisor = 1;
console.log("divisores de " + num);
while (divisor <= num) {
  if (num % divisor == 0) {
    console.log(divisor);
  }
  divisor++;
}
```

TS JS

for

Linguagem

```
const prompt=require("prompt-sync")();

var base = prompt("base:");
var expo = prompt("expoente:");
var i, res=1;
if(expo!=0){
  res=base;
  for(i=1;i<expo;i++){
    res*=base;
  }
}
console.log(base +" ^ "+ expo + "=" + res);
```

TS JS

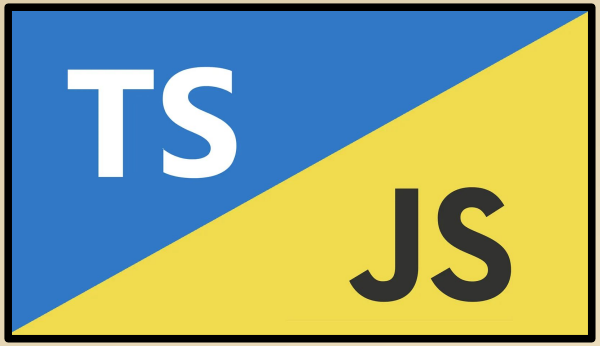
forEach

Linguagem

```
const prompt=require("prompt-sync")();
let qtd = prompt("Quantos números? "));
let numeros = [];
for (let i = 0; i < qtd; i++) {
  let num = parseFloat(prompt(
      "Digite o número ${i + 1}: "));
  numeros.push(num);
}
console.log("Número | Quadrado");

numeros.forEach((numero) => {
  console.log(`${numero}  |  ${numero ** 2}`);
});
```

- Introdução
- Ambiente JS
- Linguagem
- Objetos Nativos



JavaScript - Parte 1

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

Objetos Nativos ou Objetos Globais

Objetos Nativos

- String
- Array
- Date
- Math
- Console

TS JS

Objetos Nativos

String

- .length → tamanho do vetor
- .charAt() → caracter a partir do índice
- .concat() → combina duas ou mais strings e retorna nova string
- .includes() → se uma string está dentro de outra
- .indexOf() → índice da primeira ocorrência
- .split() → divide String em um array de strings.
- .substr() → caracteres de início e tamanho
- .substring() → caracteres de faixa de indice
- .toUpperCase() → converte em maiúscula.
- .toLowerCase() → converte em minúscula.

TS JS

Objetos Nativos

String - Exemplo

```
const nome = prompt("Nome completo:");

nomeCompleto = nome.toLowerCase().trim();

const partes = nomeCompleto.split(" ");
const primeiroNome = partes[0];
const ultimoNome = partes[partes.length - 1];

const email = primeiroNome.concat(".",
  ultimoNome, "@empresa.com");

console.log("Email gerado:", email);
```

TS JS

Vetores ou Matrizes

Objetos Nativos

- Vetores → conjunto unidimensionais

```
var notas = [10], aprovados = [];
```

- Valores são acessados por índices

```
nota[5] = 8;
```

- Matrizes → conjunto multidimensionais

```
var pixels = [640];
for(i=0;i<640;i++){
  pixels[i]=[480];
  for(j=0;j<480;j++){
    pixels[i][j]=0;
    console.log(pixels[i][j])
  } }
```

TS JS

Vetores ou Matrizes

Objetos Nativos

- .length → tamanho do vetor
- .pop() → remove no final elemento e o retorna
- .push() → insere no final um ou mais elementos
- .shift() → remove no início primeiro elemento e retorna. Muda o tamanho do array
- .unshift() → insere no início um ou mais elementos do vetor. Muda o tamanho do array
- .sort() → ordena o vetor e o retorna

TS JS

Objetos Nativos

```
const prompt=require("prompt-sync")();

const unid=["zero","um","dois","três","quatro",
 "cinco","seis","sete","oito","nove","dez",
 "onze","doze","treze","quatorze","quinze",
 "dezesseis","dezesete","dezoito","dezenove"];
const dezena=["zero","dez","vinte","trinta",
 "quarenta","cinquenta","sessenta","setenta",
 "oitenta","noventa"];
var num = prompt("número:");
if(num<=20)
  return console.log(unid[num]);
else{
  var d=Math.trunc(num/10);
  var u=num%10;
  if(u>0)console.log(dezena[d]+" e "+unid[u]);
  else console.log(dezena[d]);
}
```

TS JS

Date

Objetos Nativos

- Date.now() → retorna data atual
- Date.parse() → transforma string em data
- Gets e Sets em Date
- getDate(), getDay(), getFullYear(), getHours(), getMilliseconds(), getMinutes(), getMonth(), getSeconds(), getTime(), getYear()
- .toString() → converte string
- .valueOf() → converte número
- .toUTCString() → converte string UTC

TS JS

Objetos Nativos

Date - Exemplo

```
const dias=["Domingo","Segunda","Terça",
   "Quarta","Quinta","Sexta","Sábado"];
var hoje=new Date();
console.log( dias[hoje.getDay()] );
```

TS JS

Math

Objetos Nativos

- Funções de ângulos
  Math.cos(),Math.acos(),Math.sin(),Math.asin(),
  Math.tan(),Math.atan()
- Funções de arredondamento
  Math.floor(), Math.round, Math.trunc,Math.abs()
- Função mínimas e máximas
- Math.min() → mínimo entre números
- Math.max() → máximo entre números
- Math.pow() → potência
- Math.random() → randômico entre 0 e 1
- Math.sqrt() → raiz quadrada

Math - Exemplo

```
const participantes = ["Ana", "Bruno",
  "Carlos", "Daniela", "Eduardo"];
const vencedor = participantes[
  Math.trunc(Math.random() *
  participantes.length)];

console.log("Vencedor foi:", vencedor);
```

TS JS

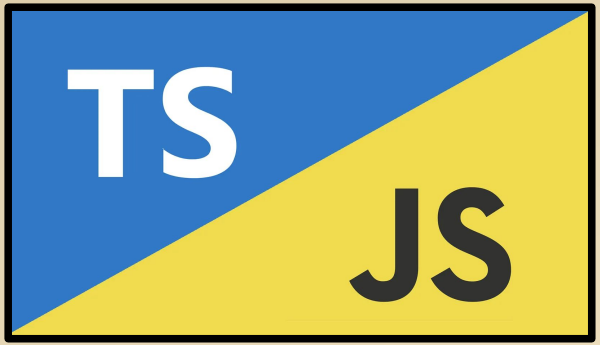
Objetos Nativos

console

- .log → imprime na mensagem no console
- .clear() → limpa o console, se o ambiente permitir.

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano



JavaScript - Parte 1