

```
//LENDO ERROS NO CONSOLE
```

```
//O console é uma ferramenta poderosa do Javascript que pode ser usada para entendermos que erros estão acontecendo no nosso programa. Para isso o console utiliza 2 ferramentas especiais, que são: "Tipo de Erro" e o "StackTrace"
```

```
//TIPO DE ERRO:
```

```
//O tipo de erro consiste nos mais diferentes tipos de erro que podem acontecer enquanto o interpretador do javascript está fazendo a leitura na nossa aplicação. Entre os tipos de erro mais comuns temos:
```

```
/*
```

```
    - RangeError:
```

```
        Quando o código recebe um dado do tipo certo, porém não dentro do formato aceitável. Por exemplo, um processamento que só pode ser feito com números inteiros maiores ou iguais a zero, mas recebe -1.
```

```
    - ReferenceError:
```

```
        Normalmente ocorre quando o código tenta acessar algo que não existe, como uma variável que não foi definida; muitas vezes é causado por erros de digitação ou confusão nos nomes utilizados, mas também pode indicar um erro no programa.
```

```
    - SyntaxError:
```

```
        Na maior parte dos casos ocorre quando há erros no programa e o JavaScript não consegue executá-lo. Os erros podem ser métodos ou propriedades escritos ou utilizados de forma incorreta, por exemplo, operadores ou sinais gráficos com elementos a menos, como esquecer de fechar chaves ou colchetes.
```

```
    - TypeError:
```

```
        Indica que o código esperava receber um dado de um determinado tipo, tal qual uma string de texto, mas recebeu outro, como um número, booleano ou null.
```

```
*/
```

```
//Os tipos de erro já facilitam a nossa vida nos ajudando identificar qual o tipo de erro que está acontecendo, e se não bastasse isso o console ainda aponta para nós a linha onde o erro está acontecendo e até o caractere, como mostra o exemplo logo abaixo:
```

```
/*
```

```
    - Erro cometido:
```

```
"const constante;" declaramos uma constante que não foi inicializada...
```

```
- Resultado no console:
```

```
c:\Users\Dell\Documents\javascript\arquivos_das_aulas\163-LEND0_ERROS_NO_CONSOLE.js:27
```

```
const constante;  
^^^^^^^^^^
```

```
SyntaxError: Missing initializer in const declaration
```

Perceba que o erro foi um "SyntaError" ou seja, um erro de escrita onde a constante não foi inicializada, acima vemos no Path do arquivo que ele aponta exatamente em que linha o erro aconteceu, na linha "27". Em alguns casos, como encontramos no stacktrace regularmente ele não só aponta a linha mas também o caractere, dessa forma: "27:15", mostrando que foi na linha 27 no caractere de número "15".

```
*/
```

```
//STACKTRACE:
```

//O stacktrace, ou "rota a pilha", é ainda mais útil quando trabalhamos com muitas bibliotecas e módulos, pois ele mostra o caminho completo entre os mais diversos módulos dentro no Javascript e do Node onde podemos onde podemos percorrer até encontrar o local onde ocorreu o erro. Vejamos um exemplo de "stacktrace":

```
/*
```

```
at Object.compileFunction (node:vm:352:18)  
at wrapSafe (node:internal/modules/cjs/loader:1031:15)  
at Module._compile (node:internal/modules/cjs/loader:1065:27)  
at Object.Module._extensions..js (node:internal/modules/cjs/loader:1153:10)  
at Module.load (node:internal/modules/cjs/loader:981:32)  
at Function.Module._load (node:internal/modules/cjs/loader:822:12)  
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)  
at node:internal/main/run_main_module:17:47
```

```
*/
```

//Perceba no trecho de código acima que ele percorre por todos os módulos que foram utilizados para que o programa fosse interpretado, mostrando por onde passou cada linha de código e seus respectivos caracteres.

//Para mais detalhes sobre os tipos de erros que podem ser detectados pelo console, acesse a página:
https://nodejs.org/api/errors.html#errors_errors