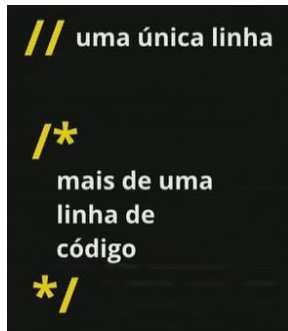


## AULA 5 – VARIÁVEIS E TIPOS PRIMITIVOS

### COMENTANDO O CÓDIGO

- É de suma importância colocar comentários no código para saber o que está sendo colocado ali



```
<script>
  window.alert('Minha primeira mensagem!')
  window.confirm('Está gostando de JS?') // Janela com o botão OK e Cancela
  window.prompt('Qual é o seu nome?') // Vai perguntar o nome
</script>
```

### VARIÁVEIS

- No JavaScript, além de utilizar a palavra **var**, também podemos utilizar a palavra **let**

- No JavaScript, eu posso atribuir uma string utilizando três tipos de aspas (dupla, apóstrofo e crase)

Exemplo:

identificador

```
var s1 = "JavaScript"
var s2 = 'Curso em Vídeo'
var s3 = `Guanabara`
```

- O nome de cada variável chamamos de **identificador**

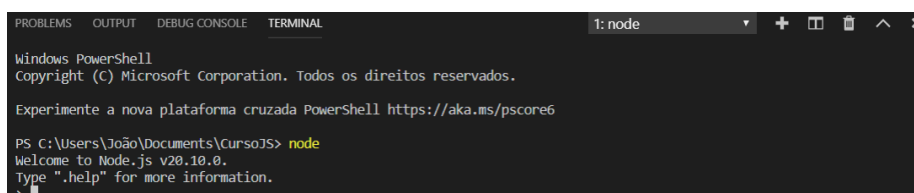
#### Regras para identificadores:

- Podem começar com **letra, \$ ou \_**
- Não podem começar com **números**
- É possível usar **letras** ou **números**
- É possível usar **acentos e símbolos**
- Não podem conter **espaços**
- Não podem ser **palavras reservadas**

### UTILIZANDO O NODE.JS

- Posso abrir ele tanto no menu iniciar como no Visual Studio

No Visual Studio: Terminal → New Terminal → Digitar node



OBS:

**.exit** → Fecha o Node.JS

**exit** → Fecha o terminal

**ctrl + L** → Limpa a tela do Node.JS

## DICAS PARA NOMEAR OS IDENTIFICADORES

- **Maiúsculas e minúsculas** fazem diferença
- Tente escolher **nomes coerentes** para as variáveis
- Evite se tornar um **'programador alfabeto'** ou um **'programador contador'**

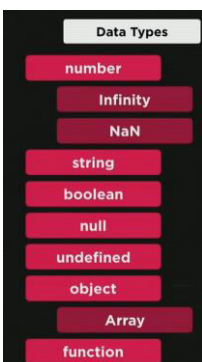
## TIPOS DE TIPOS PRIMITIVOS ESSENCIAIS



O JavaScript não diferencia números inteiros e reais!



OBS: No JavaScript, existem muitos outros tipos primitivos, como:




- Para eu saber esses tipos todos, tem um comando valioso no JavaScript: **typeof (tipo de)**

Exemplo:

```
> var n = 200
undefined
> n
200
> typeof n
'number'
```

OBS: Quando eu declaro uma variável, eu não digo o tipo primitivo dela! Exemplo:

```
> var n = 200  
undefined  
> n  
200  
> typeof n  
'number'  
> n = 'Google'  
'Google'  
> typeof n  
'string'
```



Nota-se que minha última atribuição à variável n foi 'Google', então, mesmo antes ela sendo uma variável **number**, ela é uma **string** devido à sua última atribuição