



File Edit Selection View Go Run Terminal Help



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <link href="https://www.sc.senac.br/assets/css/estilo.css" rel="stylesheet" type="text/css" />
5     &display=sw
6   </head>
7   <body>
8     <div class="cabecalho">
9       <nav>
10         <a href="#">Home</a>
11         <ul>
12           <li><a href="#">Sobre o Desenvolvedor</a></li>
13           <li><a href="#">Contato</a></li>
14           <li><a href="#">Aulas</a></li>
15         </ul>
16       </nav>
17     </div>
```

The screenshot shows a web browser window with the URL <HTTPS://PORTAL.SC.SENAC.BR/>. The page features a large, stylized logo with blue and orange waves above the text "Desenvolvedor". Below the logo, the word "Senac" is written in large, bold, blue letters. A dark blue banner at the bottom contains the text "Transformando códigos em sites incríveis". A cursor is visible over the banner.



File Edit Selection View

Untitled -TextEdit

File Edit View Help

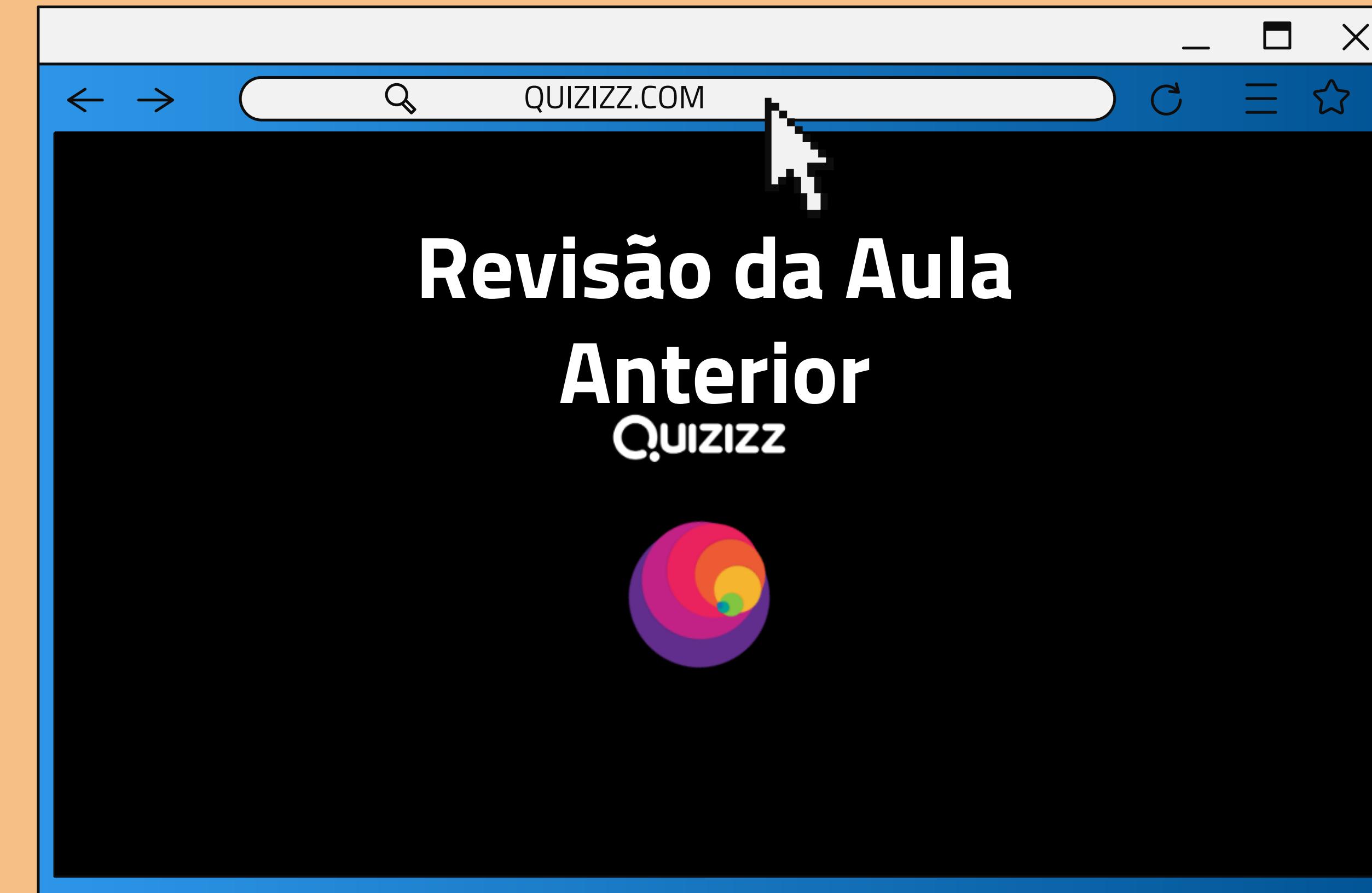
01	Revisão da Aula Anterior
02	Objetos
03	Array
04	Manipulação de Funções
05	Boas Práticas em Geral
06	Finalização da Aula

◀ ▶ //





File Edit Selection View Go Run Terminal Help





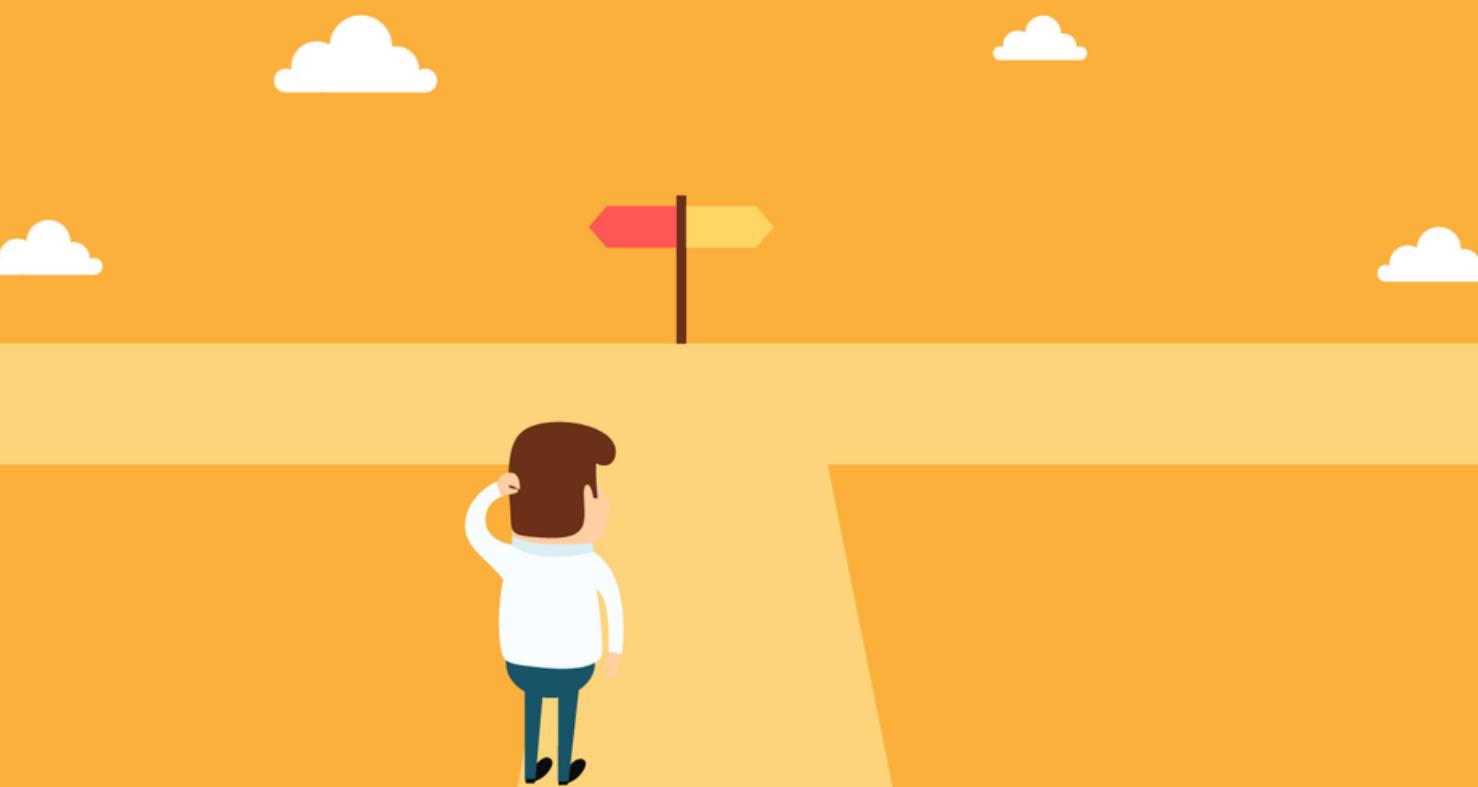
# ESTRUTURAS CONDICIONAIS

JAVASCRIPT

# Script Runner 2.0

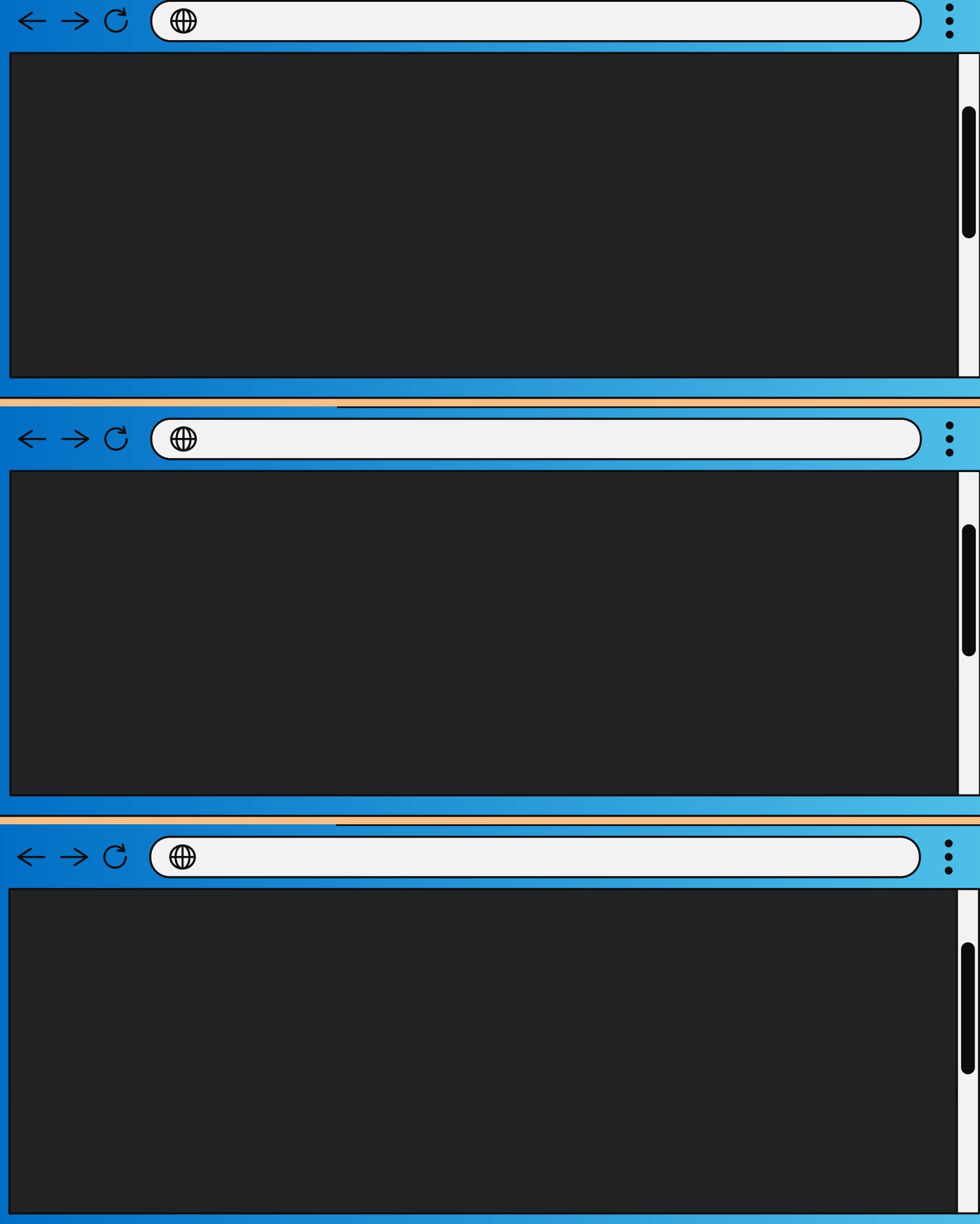
Script Language:

HyperType



Run Script

**if**



## Script Runner 2.0

Script Language: HyperType



Run Script

if

← → ⌂

```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  };
  alert(mensagem);
```

← → ⌂

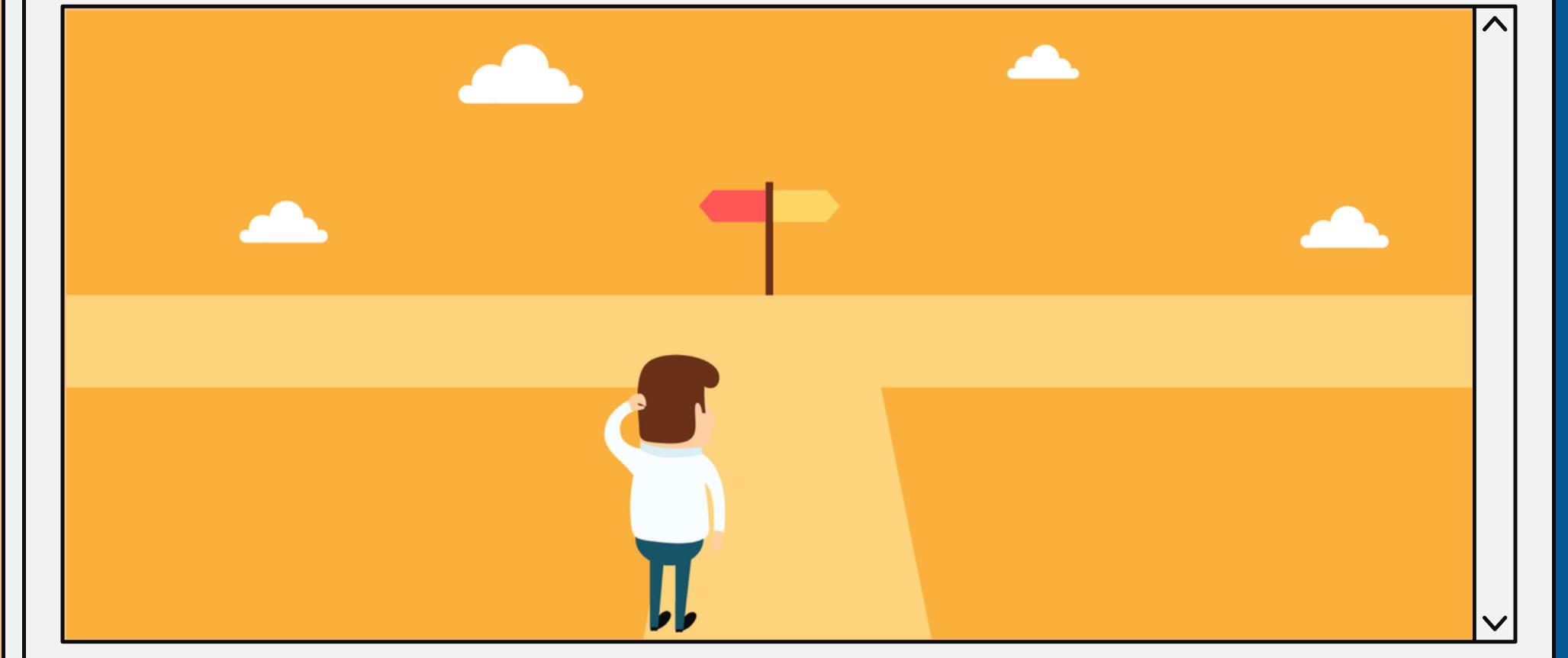
```
> 
```

← → ⌂

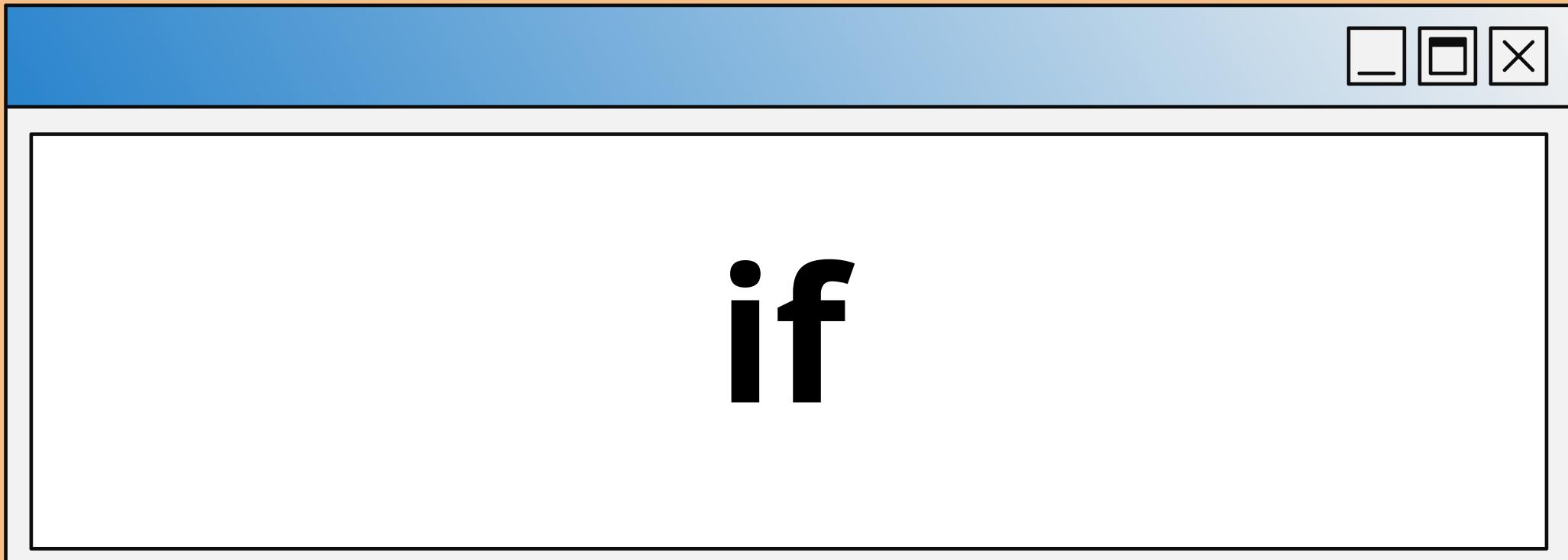
```
> 
```

## Script Runner 2.0

Script Language: HyperType



Run Script



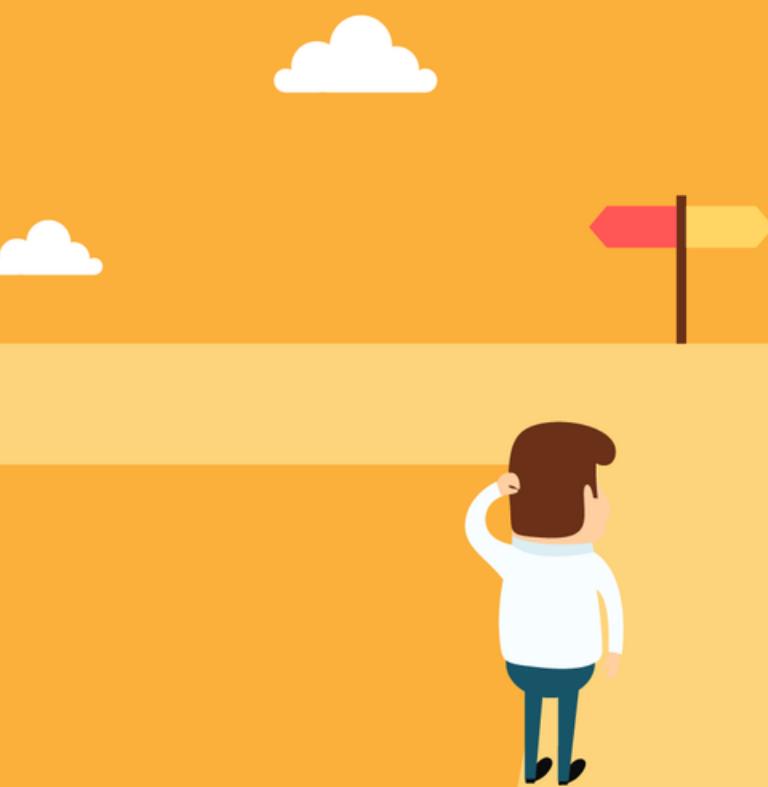
```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  };
  alert(mensagem);
```

```
chrome://new-tab-page diz
undefined
```

OK

## Script Runner 2.0

Script Language: HyperType



Run Script

if

← → ⌂



⋮

```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  };
  alert(mensagem);
```

← → ⌂



⋮

```
chrome://new-tab-page diz  
undefined
```

OK

← → ⌂



⋮

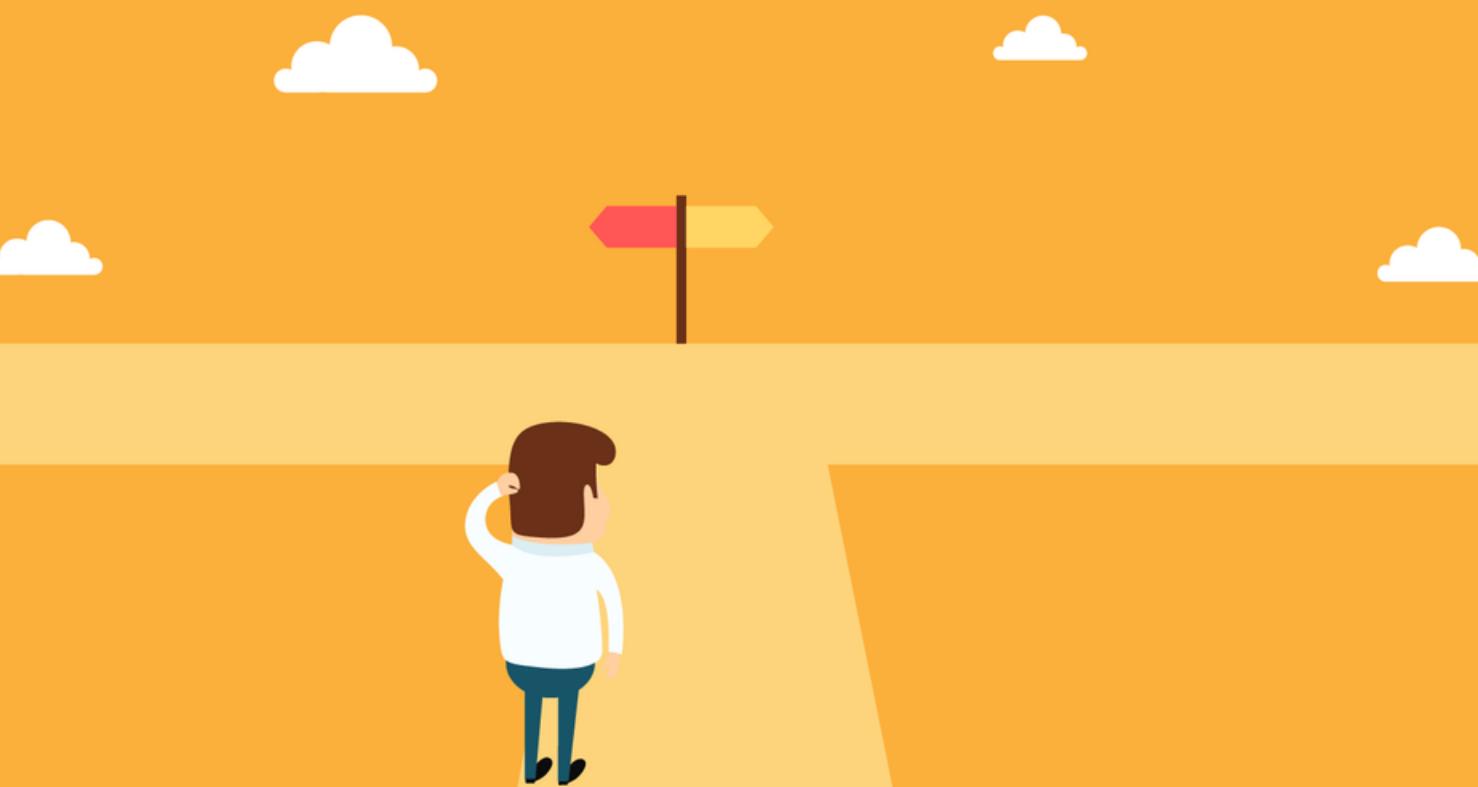
```
chrome://new-tab-page diz  
Maior de idade
```

OK

# Script Runner 2.0

Script Language:

HyperType

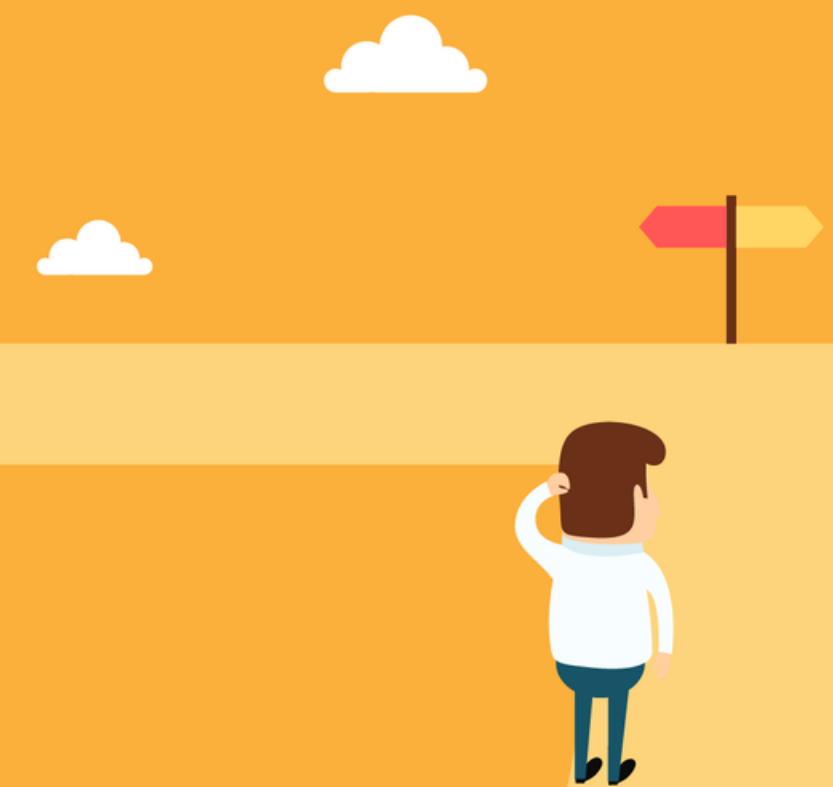


Run Script

**if else**

## Script Runner 2.0

Script Language: HyperType



Run Script

# if else

← → ⌂

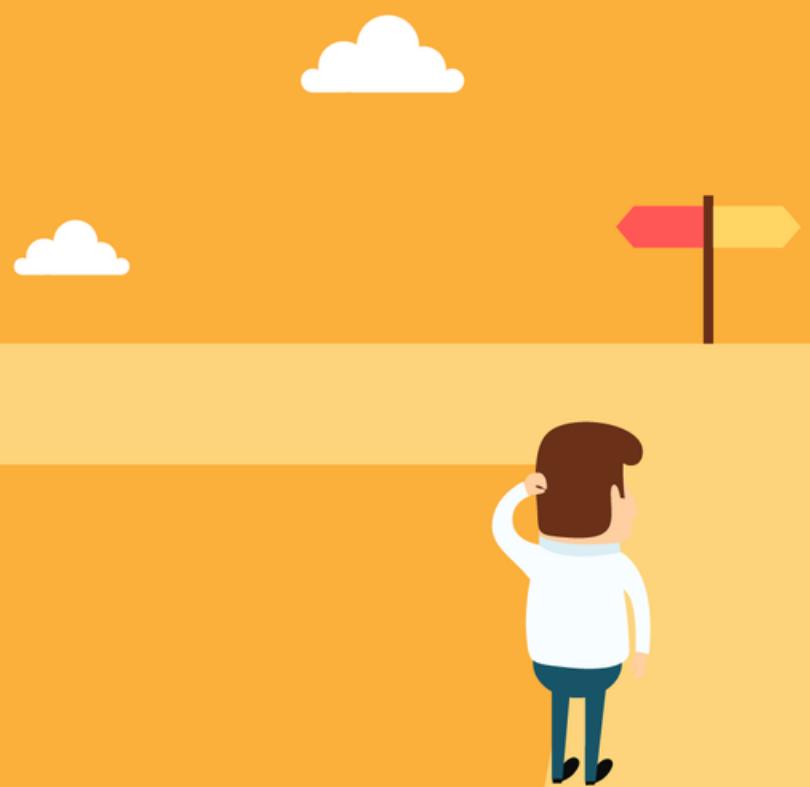
```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  } else {
    var mensagem = 'Menor de idade';
  };
  alert(mensagem);
```

← → ⌂

← → ⌂

## Script Runner 2.0

Script Language: HyperType



Run Script

# if else

← → ⌂

```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  } else {
    var mensagem = 'Menor de idade';
  };
  alert(mensagem);
```

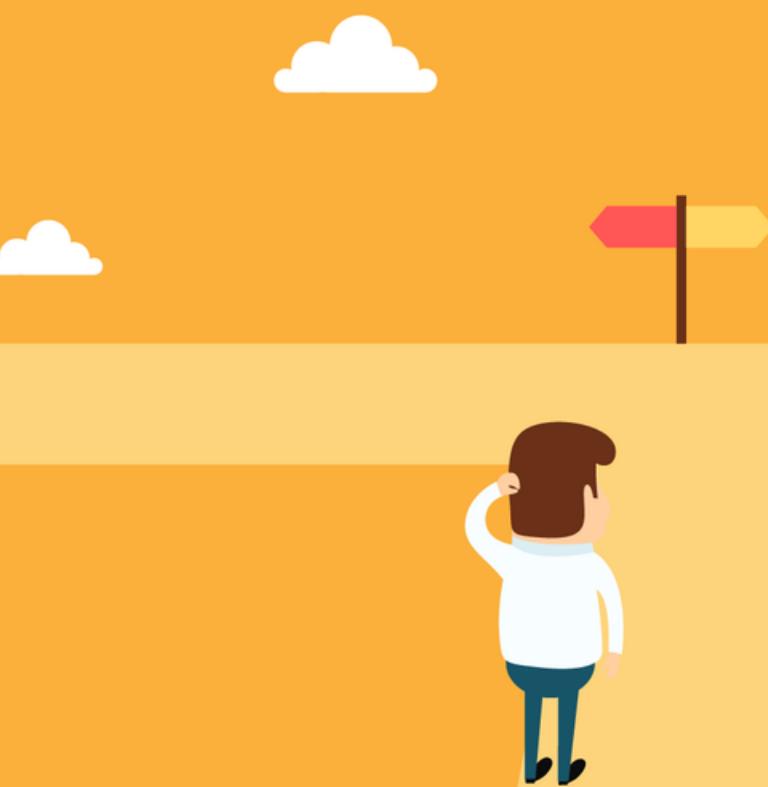
← → ⌂

```
> var idade = prompt('Insira uma idade');
  if ( idade > 18 ) {
    var mensagem = 'Adulto';
  } else if ( idade < 13 ) {
    var mensagem = 'Criança';
  } else {
    var mensagem = 'Adolescente';
  };
  alert(mensagem);
```

← → ⌂

## Script Runner 2.0

Script Language: HyperType



Run Script

# if else

← → ⌂

```
> var idade = prompt('Insira uma idade');
  if (idade > 18 ) {
    var mensagem = 'Maior de idade ';
  } else {
    var mensagem = 'Menor de idade';
  };
  alert(mensagem);
```

← → ⌂

```
> var idade = prompt('Insira uma idade');
  if ( idade > 18 ) {
    var mensagem = 'Adulto';
  } else if ( idade < 13 ) {
    var mensagem = 'Criança';
  } else {
    var mensagem = 'Adolescente';
  }
  alert(mensagem);
```

← → ⌂

```
> var idade = prompt('Insira uma idade');
  if ( idade >= 18 ) {
    var mensagem = 'Adulto';
  } else if ( idade < 11 ) {
    var mensagem = 'Criança';
  } else if ( idade >= 11 ) {
    var mensagem = 'Pré-adolescente';
  } else {
    var mensagem = 'Adolescente';
  }
  alert(mensagem);|
```

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = 'Maior de idade'
} else {
    var mensagem = 'Menor de idade'
};
alert(mensagem);
```

```
var idade = prompt('Insira uma idade');

if ( idade > 18 ) {
    var mensagem = 'Maior de idade'
} else {
    var mensagem = 'Menor de idade'
};

alert(mensagem);
```

variável

declarando a variável

```
var idade = prompt('Insira uma idade');

if ( idade >= 18 ) {
    var mensagem = 'Maior de idade'
} else {
    var mensagem = 'Menor de idade'
};

alert(mensagem);
```

variável

vai receber um dado

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    condição
    se
} else {
    var mensagem = 'Menor de idade'
};
alert(mensagem);
```

condição  
se

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = 'Maior de idade'
} else {
    var mensagem = 'Menor de idade'
};
alert(mensagem);
```

condição

dentro de parêntesis (...)

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {  
}  
}  
|
```

condição  
começo dos parâmetros

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
```

```
}
```

**condição**  
**final dos parâmetros**

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = 'Maior de idade'
}
}
```

**Parâmetros**  
o que a condicional deve fazer

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = "Maior de idade"
} else {
    |       senão
    |       se for diferente
};
```

senão

se for diferente

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = "Maior de idade"
} else {
    var mensagem = "Menor de idade"
};
```

senão  
faça isso

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = "Maior de idade"
} else {
    var mensagem = "Menor de idade"
};
alert(mensagem);
```

resultado

me mostre o resultado



Script Runner 2.0

Script Language: HyperType

```
var idade = prompt('Insira uma idade');
if ( idade >= 18 ) {
    var mensagem = 'Maior de idade'
} else {
    var mensagem = 'Menor de idade'
};
alert(mensagem);
```

Run Script

Awesome Web Browser

```
var idade = prompt('Insira uma idade')
var mensagem = ( idade >= 18 ?
    "Maior de idade" :
    "Menor de idade");
alert(mensagem);
```

← → C



⋮

```
var idade = prompt('Insira uma idade');
var mensagem = ( idade >= 18 ?
    "Maior de idade" :
    "Menor de idade");
alert(mensagem);
```

← → C



⋮

```
var idade = prompt('Insira uma idade');
var mensagem = ( idade >= 18 ?
    "Maior de idade" :
    "Menor de idade");
alert(mensagem);
```

Condição à variável  
vai acontecer tal se tal

← → C



⋮

```
var idade = prompt('Insira uma idade');
var mensagem = ( idade >= 18 ?
    "Maior de idade" :
    "Menor de idade");
alert(mensagem);
```

**Condição à variável  
vai acontecer tal se tal**

Awesome Web Browser



← → C



⋮

```
var idade = prompt('Insira uma idade');
var mensagem = ( idade >= 18 ?
    "Maior de idade" :
    "Menor de idade");
alert(mensagem);
```

?

**se for verdadeiro**

Awesome Web Browser



← → C

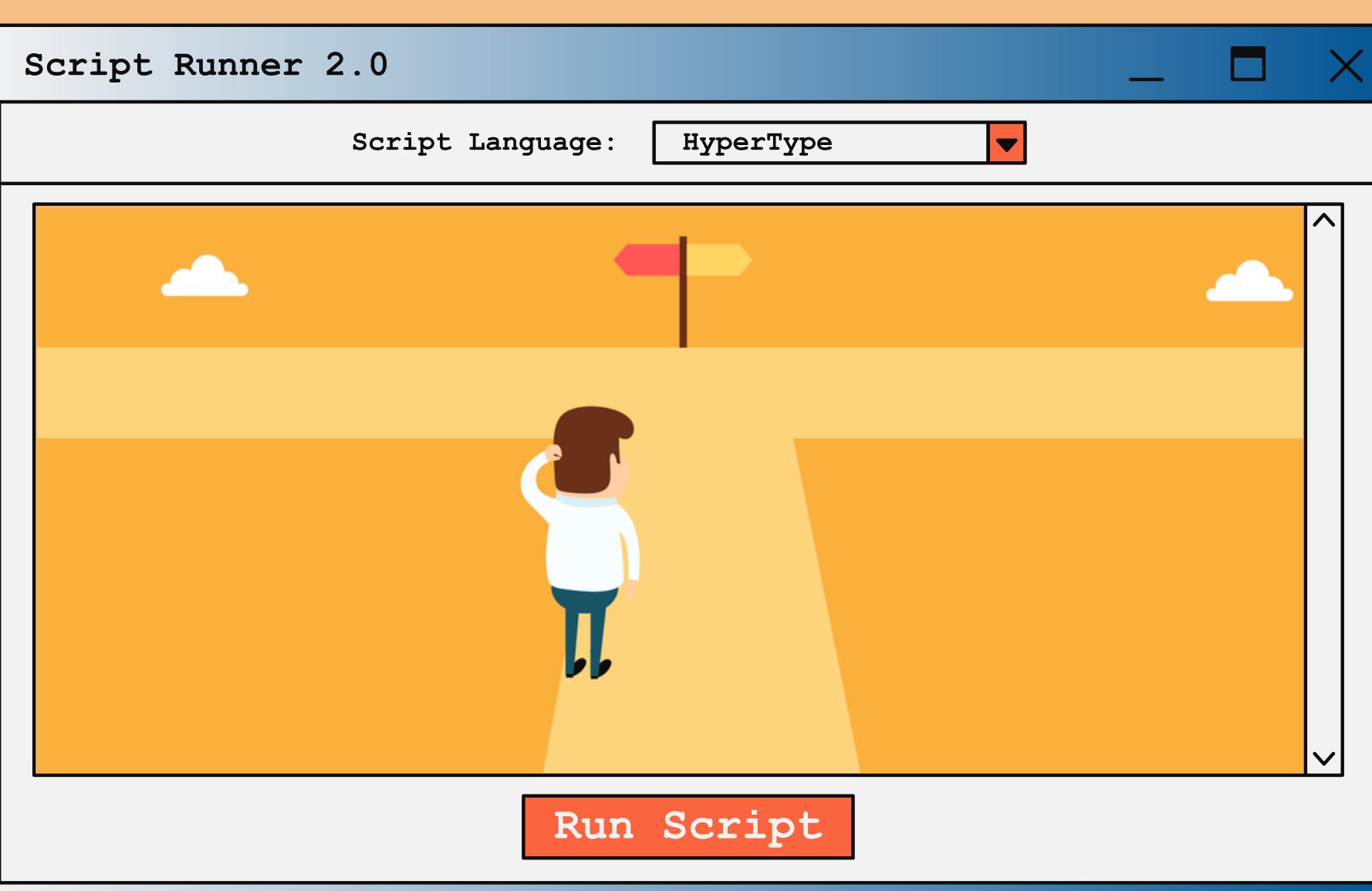


⋮

```
var idade = prompt('Insira uma idade');
var mensagem = ( idade >= 18 ?
    "Maior de idade"
    "Menor de idade");
alert(mensagem);
```

se for falso

# Booleano



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

## Verdadeiro ou Falso

Os dados booleanos só podem assumir dois valores: **true** ou **false**.

Com a condicional **if**, podemos usar livremente esses dados, pois queremos verificar se os dados são os que queremos ou não, e aí fazer algo com eles, de acordo com a resposta.

Page 1 Words: 88

# Switch - case



## Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font

12

B

I

U

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

☰ ☰ ☱

# **switch para (signo)**

```
let signo = prompt('Insira seu signo');
switch (signo) {
    case 'Aries':
        console.log("De 21 março a 20 abril");
        break;
    case 'Touro':
        console.log("de 21 abril a 20 maio");
        break;
    case 'Gêmeos':
        console.log("de 21 maio a 20 junho");
        break;
    case 'Câncer':
        console.log("de 21 junho a 22 julho");
        break;
    case 'Leão':
        console.log("de 23 julho a 22 agosto");
        break;
    case 'Virgem':
        console.log("de 23 agosto a 22 setembro");
        break;
    default:
        console.log("Signo não registrado");
        break;
}
```

**case**  
**caso ' ... '**

```
let signo = prompt('Insira seu signo');
switch (signo) {
    case 'Áries':
        console.log("De 21 março a 20 abril");
        break;
    case 'Touro':
        console.log("de 21 abril a 20 maio");
        break;
    case 'Gêmeos':
        console.log("de 21 maio a 20 junho");
        break;
    case 'Câncer':
        console.log("de 21 junho a 22 julho");
        break;
    case 'Leão':
        console.log("de 23 julho a 22 agosto");
        break;
    case 'Virgem':
        console.log("de 23 agosto a 22 setembro");
        break;
    default:
        console.log("Signo não registrado");
        break;
}
```

•  
**faça**

```
let signo = prompt('Insira seu signo');
switch (signo) {
    case 'Áries' :
        console.log("De 21 março a 20 abril");
        break;
    case 'Touro' :
        console.log("de 21 abril a 20 maio");
        break;
    case 'Gêmeos' :
        console.log("de 21 maio a 20 junho");
        break;
    case 'Câncer' :
        console.log("de 21 junho a 22 julho");
        break;
    case 'Leão' :
        console.log("de 23 julho a 22 agosto");
        break;
    case 'Virgem' :
        console.log("de 23 agosto a 22 setembro");
        break;
    default:
        console.log("Signo não registrado");
        break;
}
```

**break**  
**fim da ação**

```
let signo = prompt('Insira seu signo');
switch (signo) {
  case 'Áries':
    console.log("De 21 março a 20 abril");
    break;
  case 'Touro':
    console.log("de 21 abril a 20 maio");
    break;
  case 'Gêmeos':
    console.log("de 21 maio a 20 junho");
    break;
  case 'Câncer':
    console.log("de 21 junho a 22 julho");
    break;
  case 'Leão':
    console.log("de 23 julho a 22 agosto");
    break;
  case 'Virgem':
    console.log("de 23 agosto a 22 setembro");
    break;
  default:
    console.log("Signo não registrado");
    break;
}
```

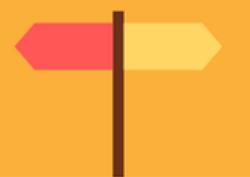
**default**  
se não for nenhum

```
let signo = prompt('Insira seu signo');
switch (signo) {
    case 'Áries':
        console.log("De 21 março a 20 abril");
        break;
    case 'Touro':
        console.log("de 21 abril a 20 maio");
        break;
    case 'Gêmeos':
        console.log("de 21 maio a 20 junho");
        break;
    case 'Câncer':
        console.log("de 21 junho a 22 julho");
        break;
    case 'Leão':
        console.log("de 23 julho a 22 agosto");
        break;
    case 'Virgem':
        console.log("de 23 agosto a 22 setembro");
        break;
    default:
        console.log("Signo não registrado");
        break;
}
```

# Booleano

Script Runner 2.0

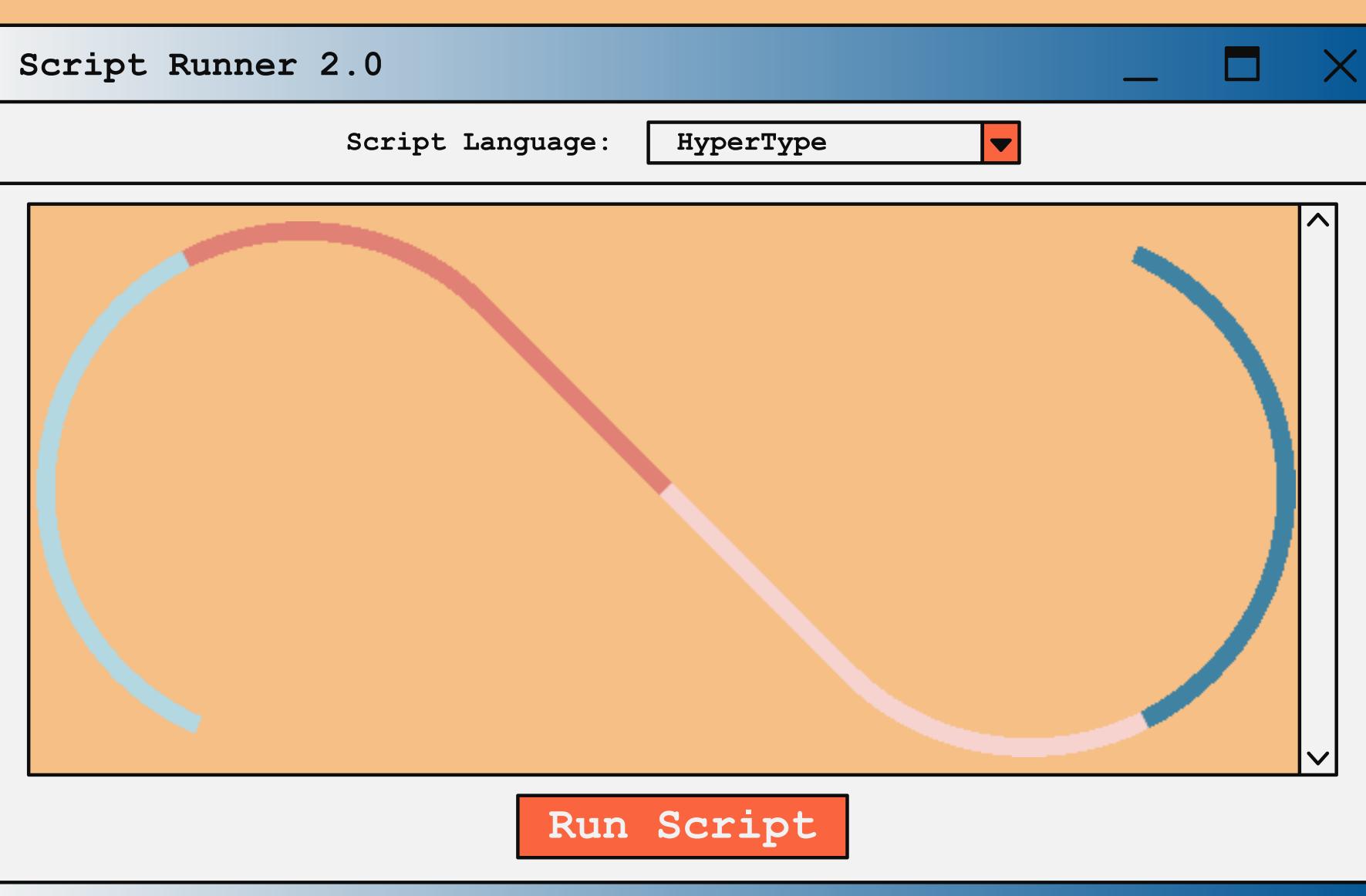
Script Language: HyperType



Run Script

```
let signo = prompt('Insira seu signo');
switch (signo) {
  case 'Áries':
    console.log("De 21 março a 20 abril");
    break;
  case 'Touro':
    console.log("de 21 abril a 20 maio");
    break;
  case 'Gêmeos':
    console.log("de 21 maio a 20 junho");
    break;
  case 'Câncer':
    console.log("de 21 junho a 22 julho");
    break;
  case 'Leão':
    console.log("de 23 julho a 22 agosto");
    break;
  case 'Virgem':
    console.log("de 23 agosto a 22 setembro");
    break;
  default:
    console.log("Signo não registrado");
    break;
}
```

# Loop



Document2 - Macrostuff Board

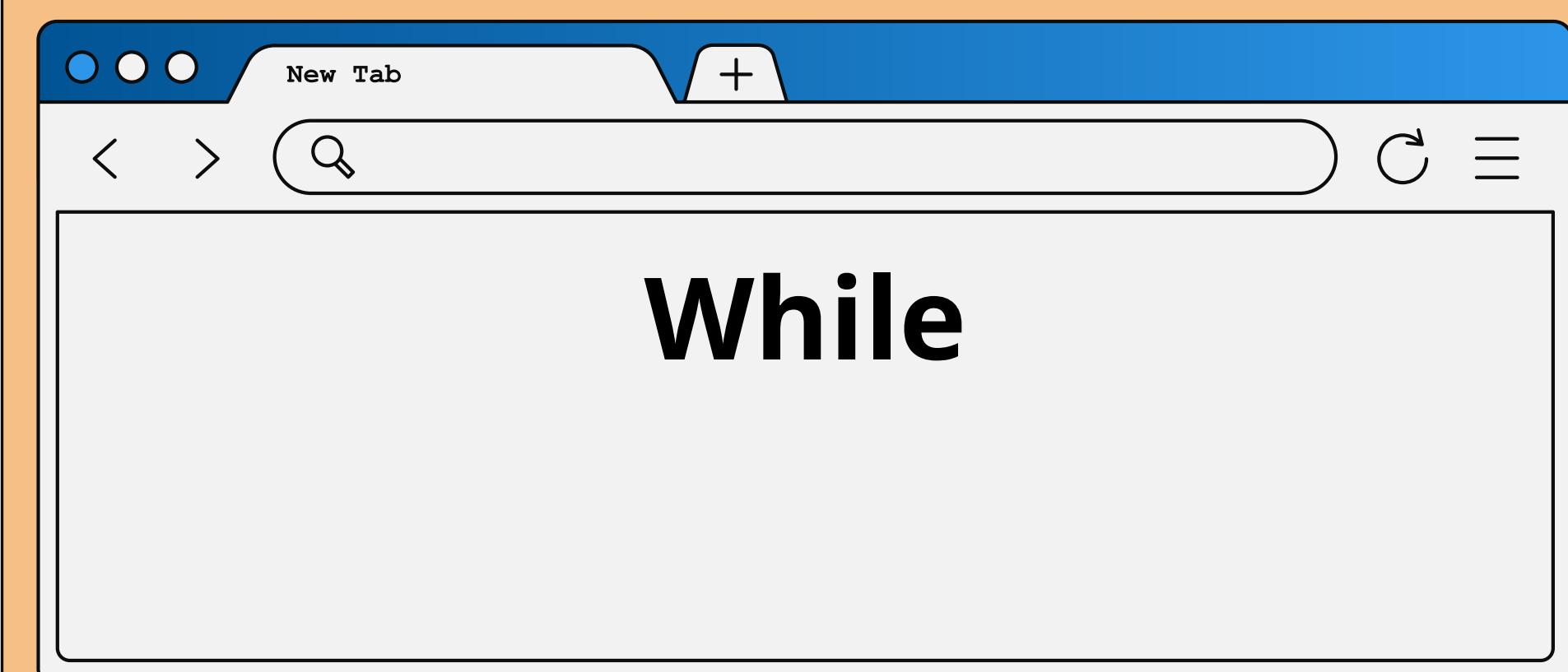
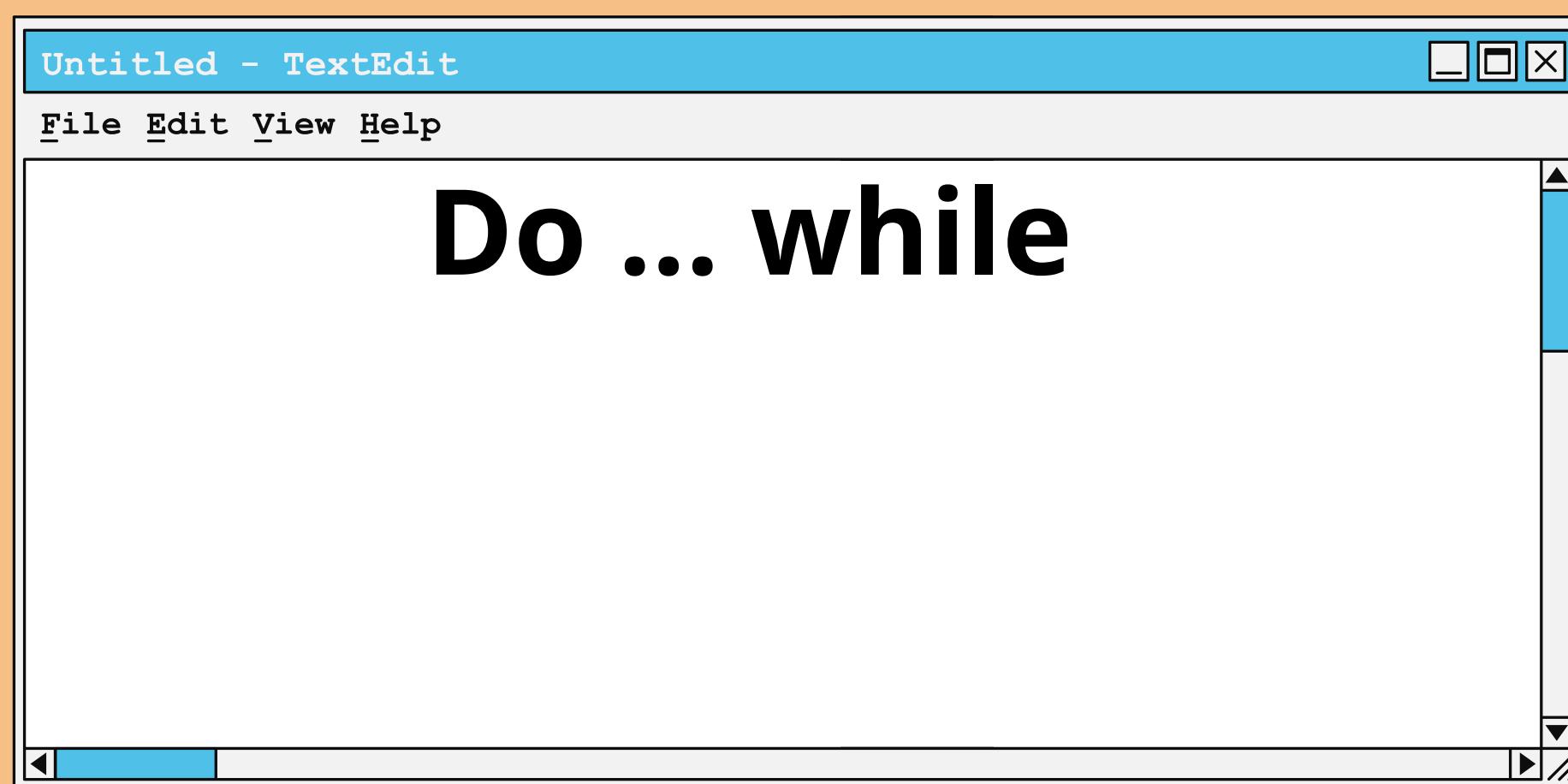
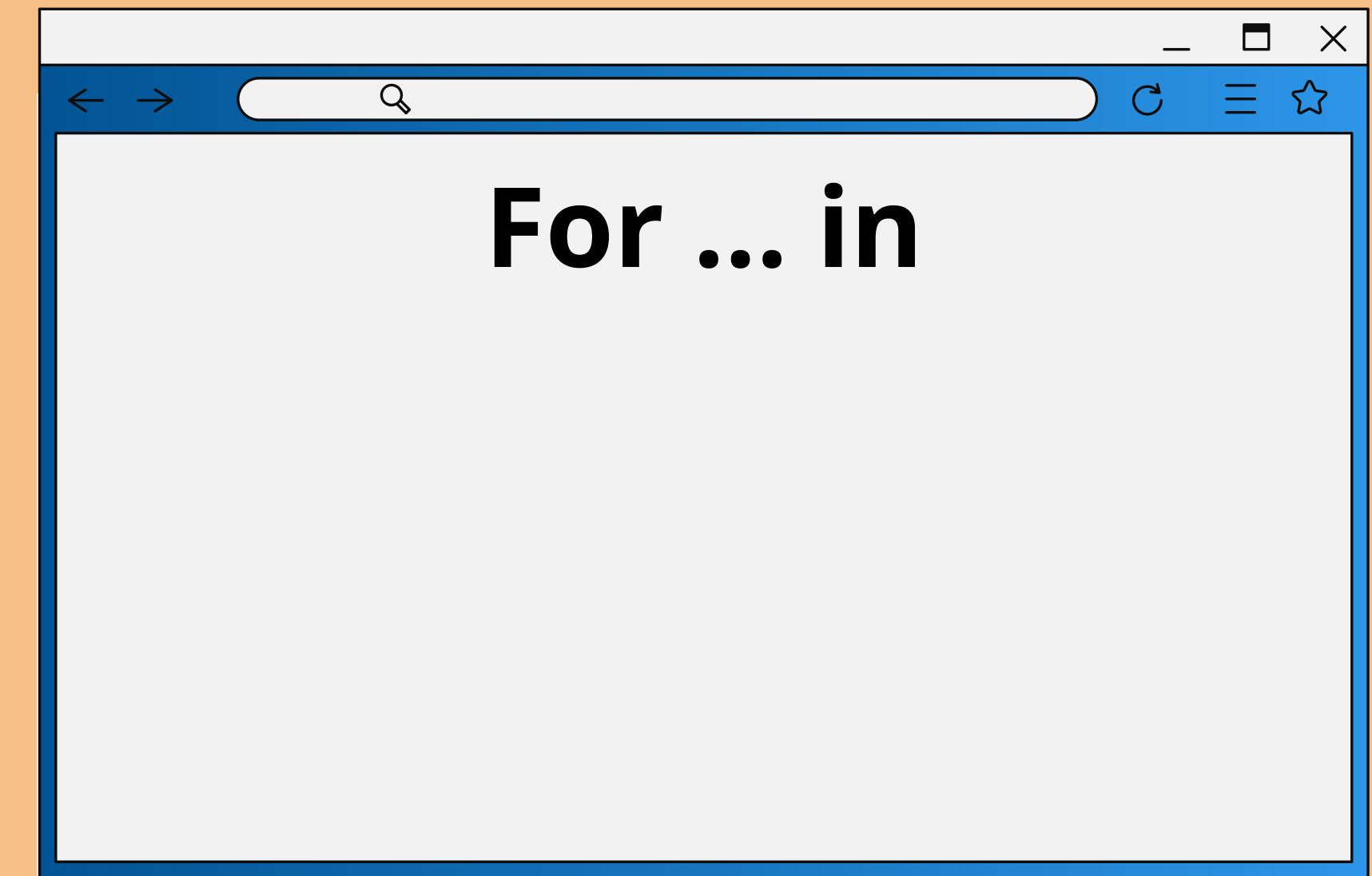
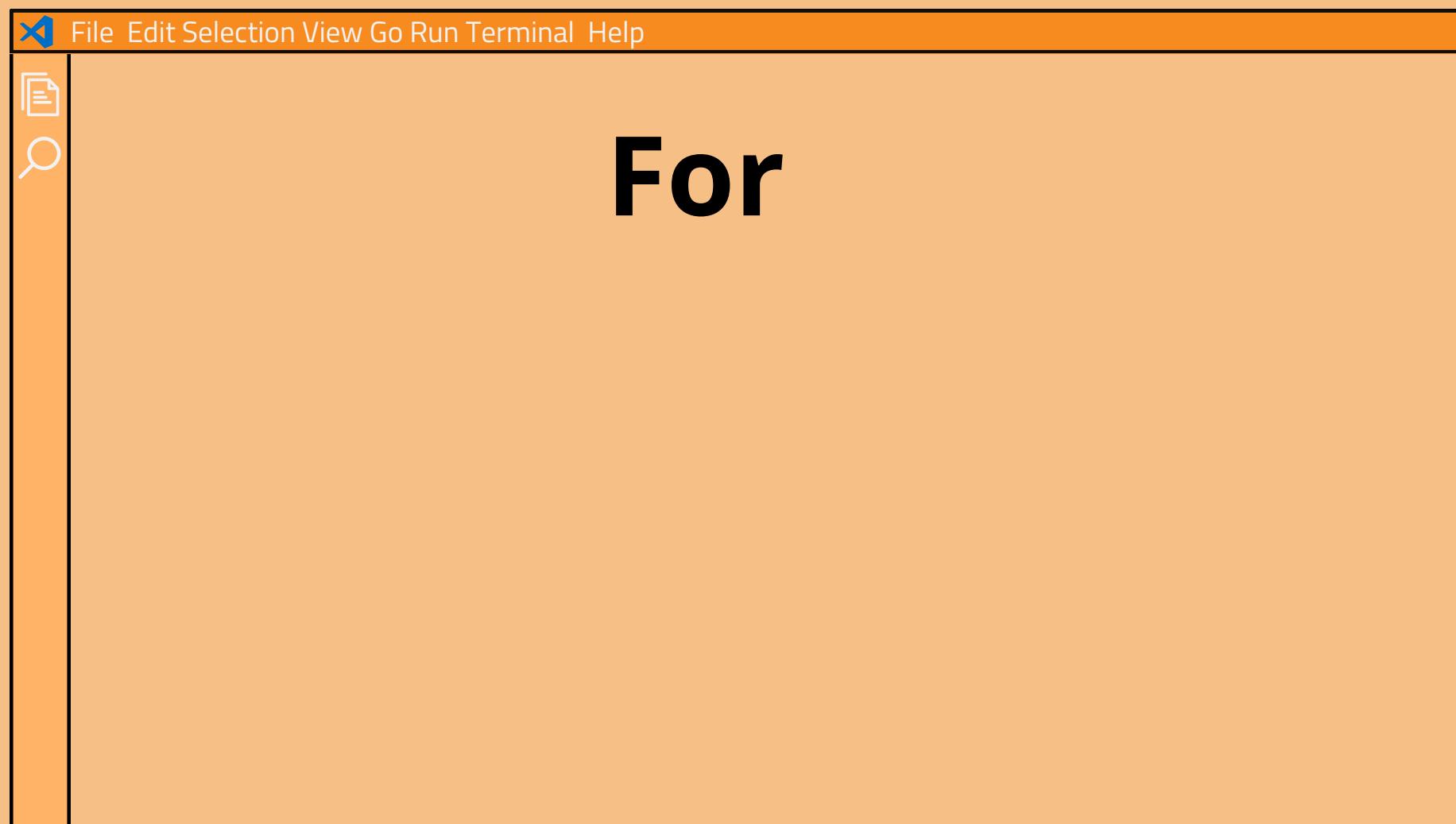
File Edit Format View Help

Cool New Font 12 B I U

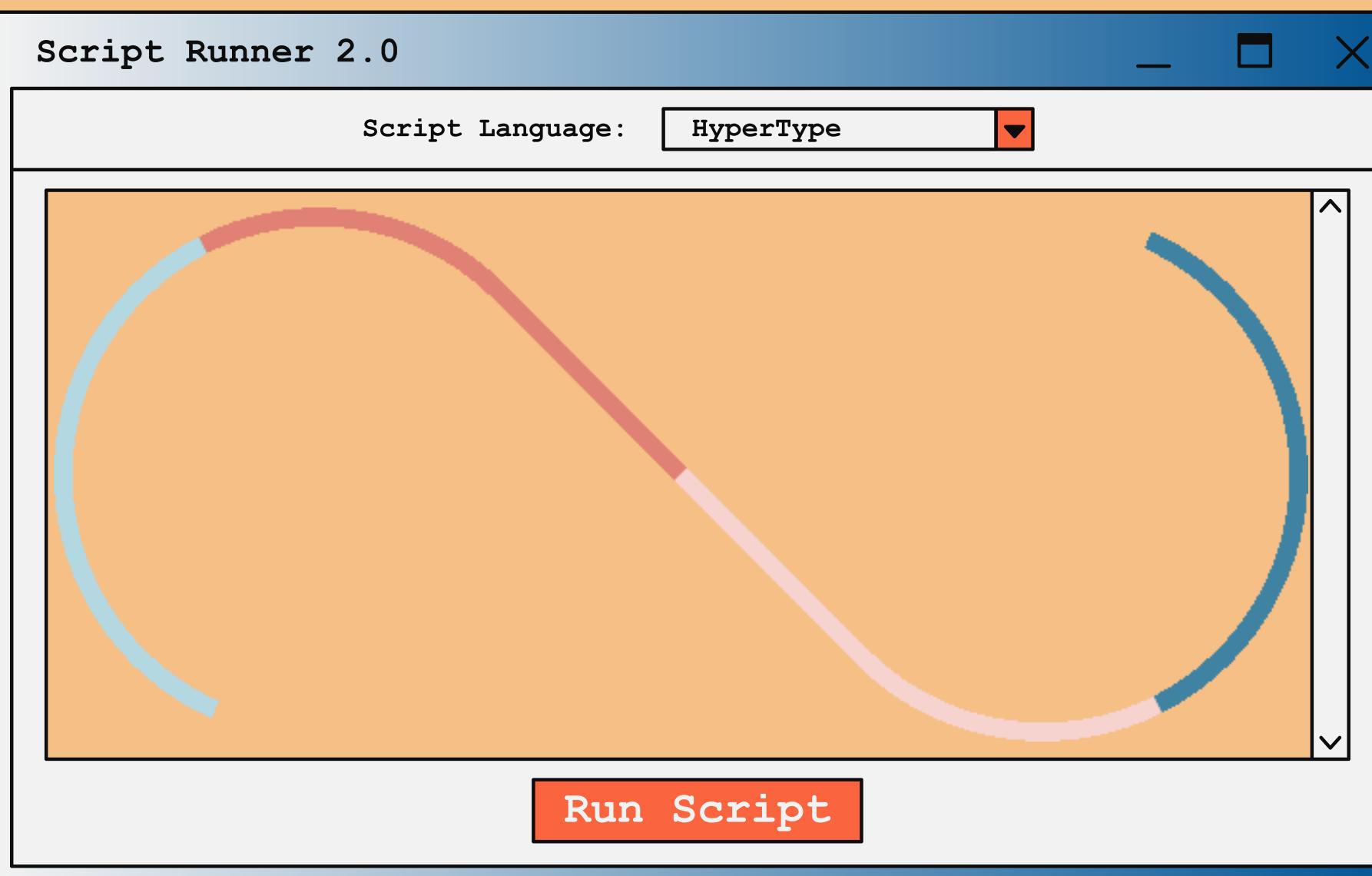
Algumas estruturas condicionais utilizam loops na sua composição.

Elas executam um **mesmo código inúmeras vezes** até que se dê uma condição para terminar e esta é atingida.

Page 1 Words: 88



# While



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

É um bloco de código que é executado até que um teste condicional se torne **falso**, sendo importante lembrar que a condição analisada para a execução do laço de repetição deverá retornar um valor **booleano**.

Page 1 Words: 88

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1  
2 let salario = 1
```

```
4 // Repetição  
5 while (salario < 5000)
```

```
6 | salario += 100
```

8 // Most

9 | console.log()

18

11

12

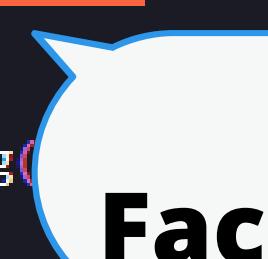
# Condição

**enquanto for verdadeiro faça**

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1
2     let salario = 1000
3
4     // Repetição
5     while (salario < 5000)
6         salario += 100;
7
8     // Mostra
9     console.log(
10    )
11
12
```





// Mostra  
console.log("O salário ainda é R\$" + salario);  
}

O salário ainda é R\$1100 VM1208:9  
O salário ainda é R\$1200 VM1208:9  
O salário ainda é R\$1300 VM1208:9  
O salário ainda é R\$1400 VM1208:9  
O salário ainda é R\$1500 VM1208:9  
O salário ainda é R\$1600 VM1208:9  
O salário ainda é R\$1700 VM1208:9  
O salário ainda é R\$1800 VM1208:9  
O salário ainda é R\$1900 VM1208:9  
O salário ainda é R\$2000 VM1208:9  
O salário ainda é R\$2100 VM1208:9  
O salário ainda é R\$2200 VM1208:9  
O salário ainda é R\$2300 VM1208:9  
O salário ainda é R\$2400 VM1208:9  
O salário ainda é R\$2500 VM1208:9  
O salário ainda é R\$2600 VM1208:9  
O salário ainda é R\$2700 VM1208:9  
O salário ainda é R\$2800 VM1208:9



The screenshot shows a browser developer tools window with the 'Console' tab selected. The console output displays a series of identical log statements: "O salário ainda é R\$1100" repeated 28 times. Each log entry is timestamped as "VM1208:9". A red box highlights the first few entries of the log.

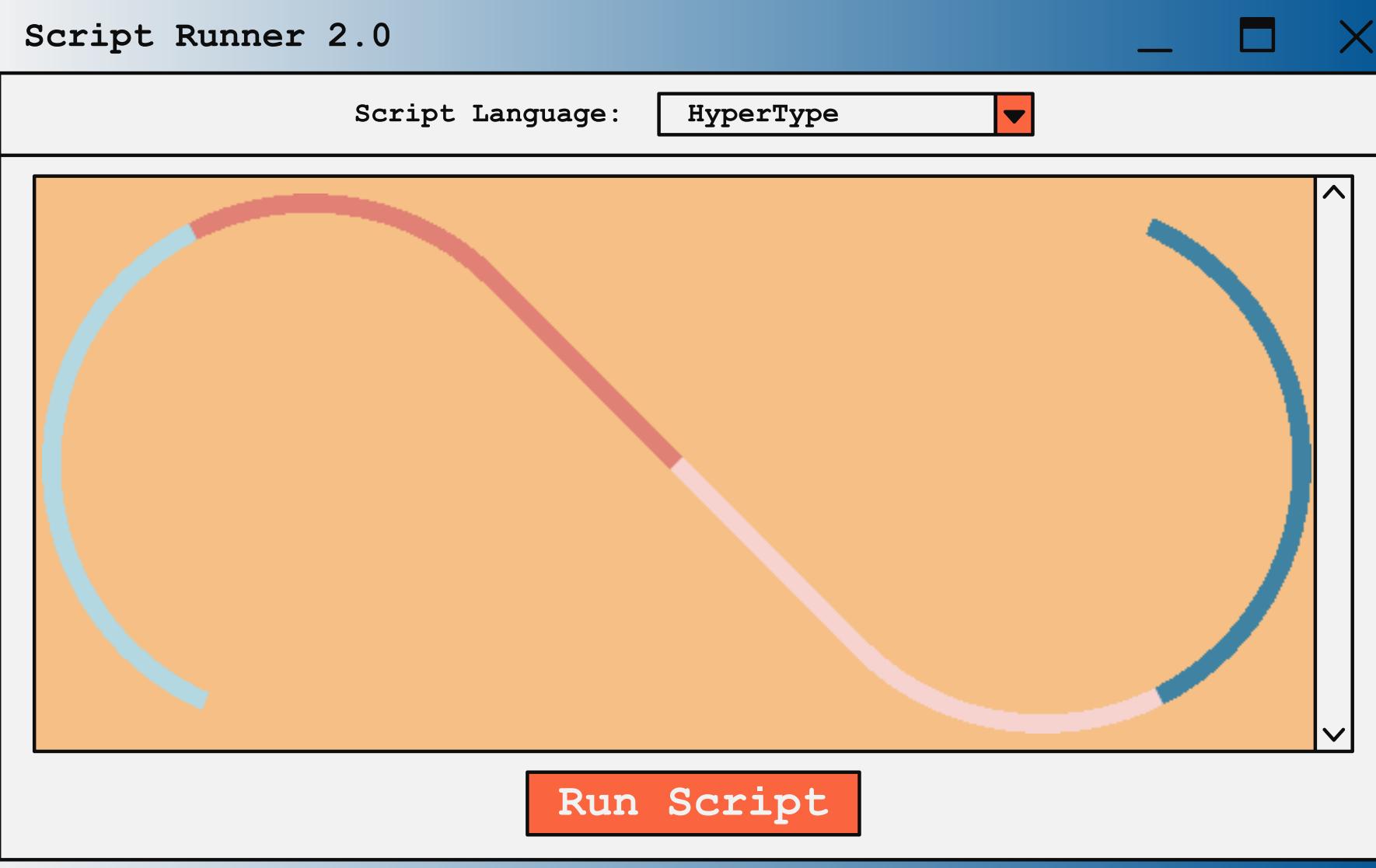
```
// Mostra
console.log("O salário ainda é R$" + salario);
}

O salário ainda é R$1100
O salário ainda é R$1200
O salário ainda é R$1300
O salário ainda é R$1400
O salário ainda é R$1500
O salário ainda é R$1600
O salário ainda é R$1700
O salário ainda é R$1800
O salário ainda é R$1900
O salário ainda é R$2000
O salário ainda é R$2100
O salário ainda é R$2200
O salário ainda é R$2300
O salário ainda é R$2400
O salário ainda é R$2500
O salário ainda é R$2600
O salário ainda é R$2700
O salário ainda é R$2800
```

**Rotina**

**Repetiu a rotina até ser falsa**

# Do ... while



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

É basicamente a mesma ideia do do, com a diferença que mesmo que a condição seja **falsa** o bloco de código **será executado** ao menos uma vez.

Page 1 Words: 88

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1 var resultado = "";
2 var i = 0;
3
4 do {
5     i += 1;
6     resultado += i + " ";
7 } while (i < 5);
```



JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1 var resultado = "";  
2 var i = 0;  
3  
4 do {  
5     i += 1;  
6     resultado += i + "  
7 } while (i < 5);
```



JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1 var resultado = "";
2 var i = 0;
3
4 do {
5     i += 1;
6     resultado += i + " ";
7 } while (i < 5);
```



Elements    **Console**    Sources    Network    Performance    Memory    Application    Security    Lighthouse    >    1    :    X

Filter    Default levels ▾    1 Issue: 1   

```
> var resultado = "";
  var i = 0;

  do {
    i += 1;
    resultado += i + " ";
  } while (i < 5);
< '1 2 3 4 5 '
```

Goo

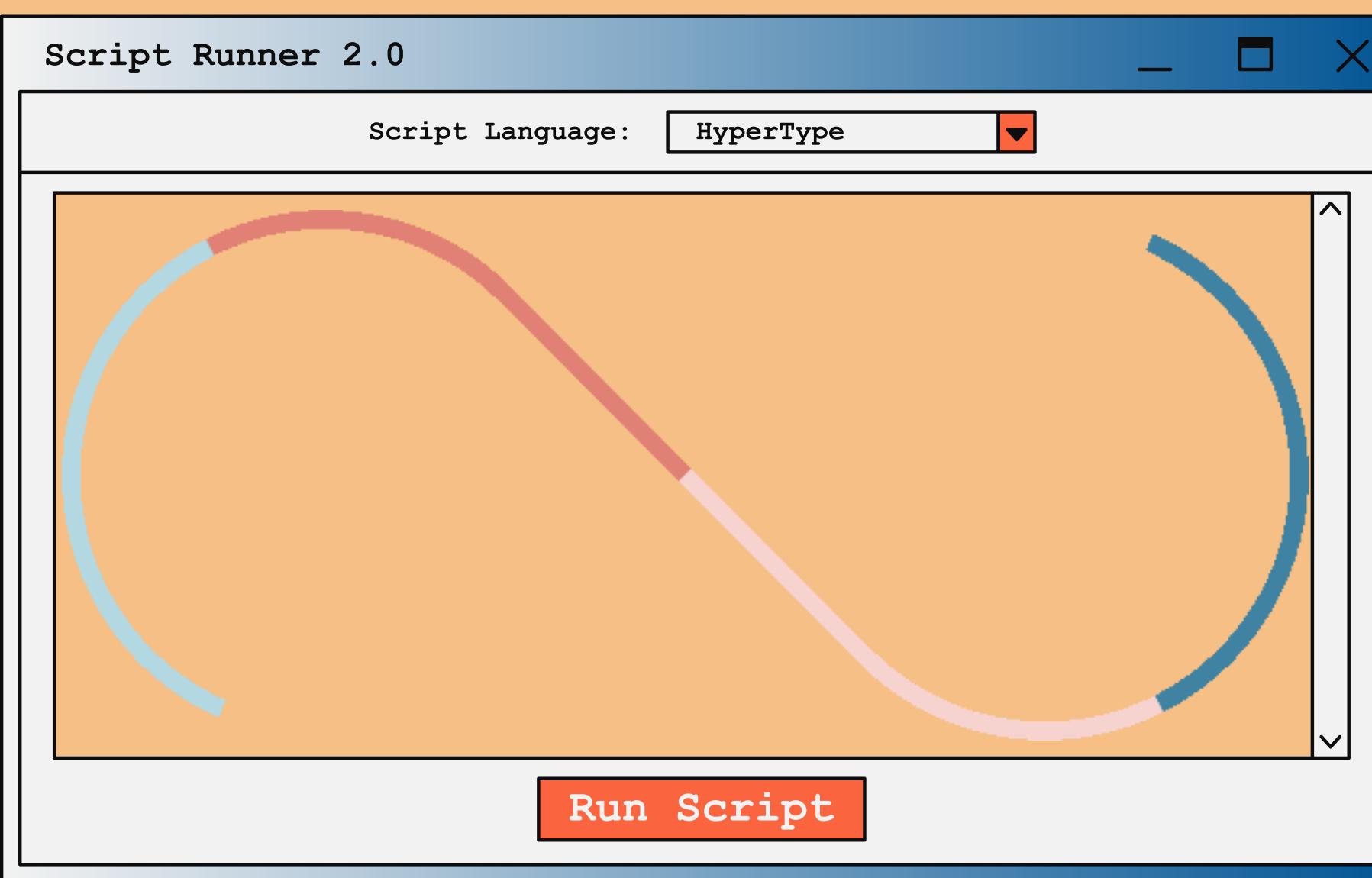
Pesquise n



One Driv



# For



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

Usamos a estrutura de **loop for** quando precisamos repetir um mesmo bloco por determinadas vezes.

Precisamos de três declarações:

- **variável** que será o controlador do nosso laço de repetição,
- **condição** a ser atendida,
- **expressão** que será executado ao final de cada iteração do for, normalmente utilizamos para incrementar a variável que será utilizada como controlador do nosso laço.

Page 1 Words: 88



```
1
2   for(let x = 0; x < 11; x++){
3     console.log("5 x " + x + " = " + 5*x);
4   }
5
6   5 x 0 = 0
7   5 x 1 = 5
8   5 x 2 = 10
9   5 x 3 = 15
10  5 x 4 = 20
11  5 x 5 = 25
12  5 x 6 = 30
13  5 x 7 = 35
14  5 x 8 = 40
15  5 x 9 = 45
16  5 x 10 = 50
17
18
```



```
1  
2  for(let x = 0; x < 11; x++){  
3      console.log(x)  
4  }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

```
for(let x = 0; x < 11; x++){  
    console.log(x)  
}
```

**Variável  
tem o valor 0**



```
1  
2   for(let x = 0; x < 11; x++){  
3     console.log("5")  
4   }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

Condição

x precisa ter o valor 11



```
1  
2      for(let x = 0; x < 11; x++){  
3          x + " = " + 5*x);  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

**expressão**

**se não for então faça ++**



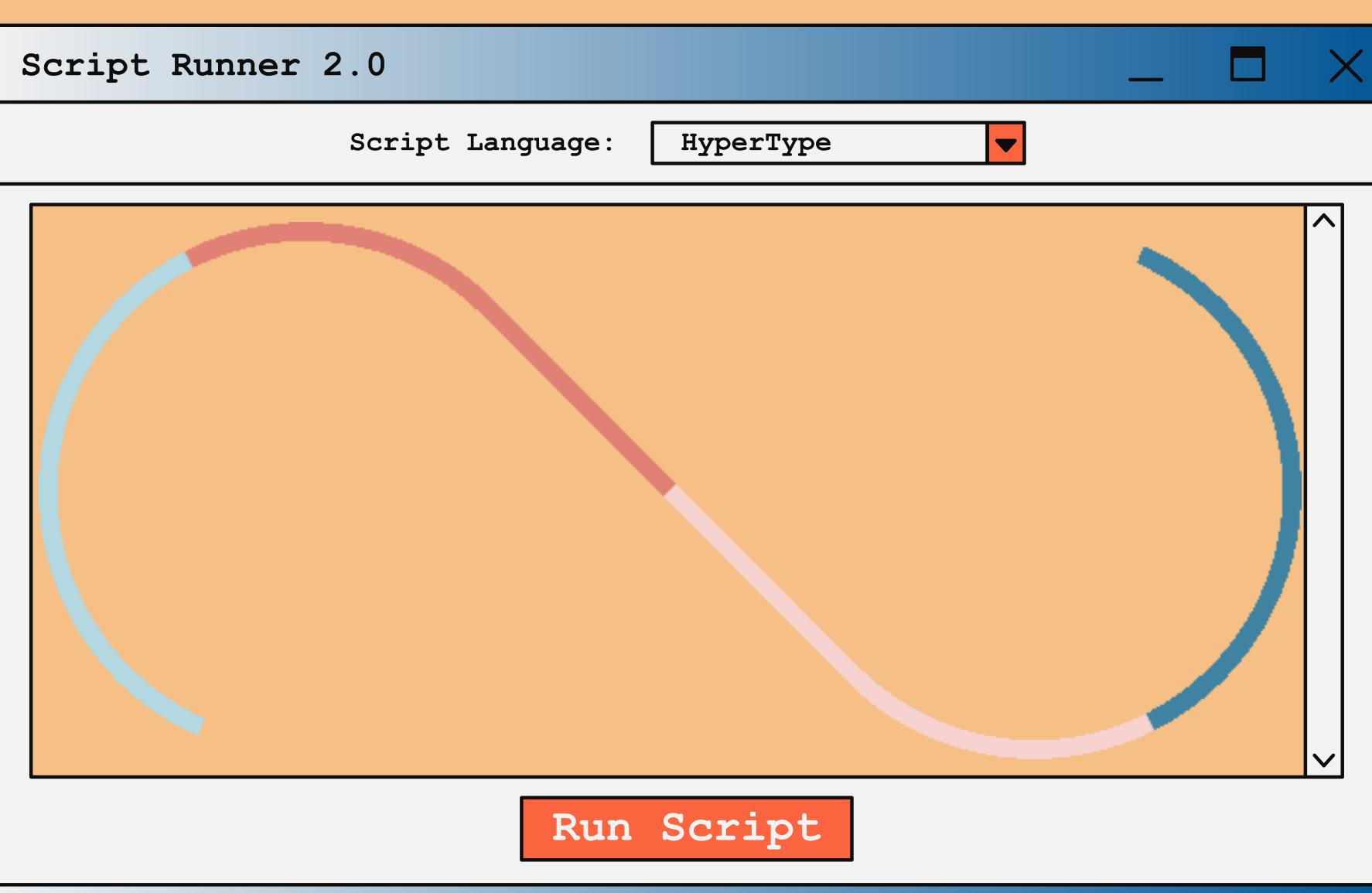
```
1  
2   for(let x = 0; x < 11; x++){  
3     console.log("5 x " + x + " = " + 5*x);  
4   }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

**me mostre o loop**



```
1
2   for(let x = 0; x < 11; x++){
3     console.log("5 x " + x + " = " + 5*x);
4   }
5
6   x 0 = 0
7   5 x 1 = 5
8   5 x 2 = 10
9   5 x 3 = 15
10  5 x 4 = 20
11  5 x 5 = 25
12  5 x 6 = 30
13  5 x 7 = 35
14  5 x 8 = 40
15  5 x 9 = 45
16  5 x 10 = 50
17
18
```

# For ... in



Document2 - Macrostuff Board

File Edit Format View Help

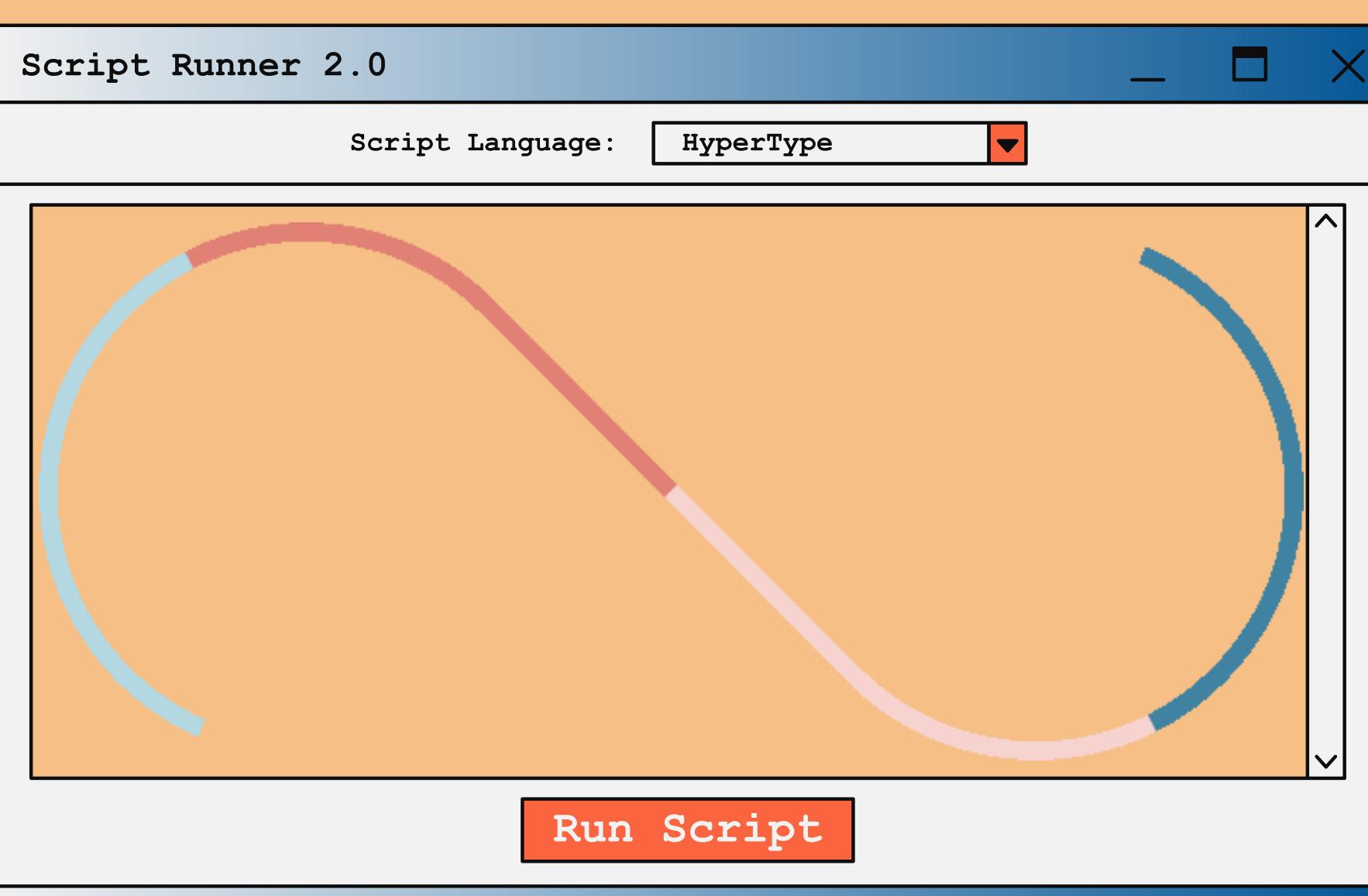
Cool New Font 12 B I U

É utilizado para loops entre as **propriedades de um objeto** e os **valores de um array**.

O processo é **mais lento** do que do For e ele **não dá prioridade** conforme a ordem das propriedades e dos elementos.

Page 1 Words: 88

# For ... in



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

É utilizado para loops entre as propriedades de um **objeto** e os valores de um **array**.

O processo é **mais lento** do que do For e ele **não dá prioridade** conforme a ordem das propriedades e dos elementos.

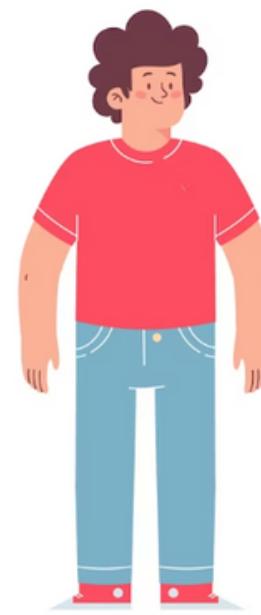
Page 1 Words: 88



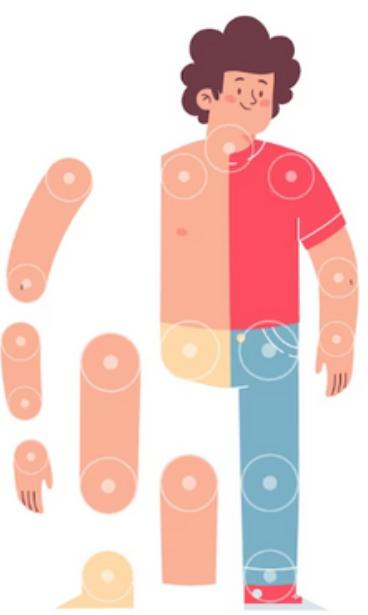
# O que é um objeto?

É uma **coleção** de **dados** e/ou **funcionalidades** relacionadas (que geralmente consistem em diversas variáveis e funções — que são chamadas de **propriedades e métodos** quando estão dentro de objetos).

**Objeto**



**Dados**



## Para quê?

Objetos são como uma espécie de "**super variáveis**" que **armazenam** uma "coleção de **valores**" referenciados por nome, e que podem ser recuperados para serem utilizados em diversas outras partes de um programa. Em JavaScript praticamente qualquer tipo de dado é um objeto.

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > .

```
1  
2 // Objeto Literal
```

# Objeto Literal

**é composto por um par de chaves "{}"**



JS Aula 12.js •

C:\Users\Carol\Desktop\JS Aula 12.js&gt; [e] pessoa



```
1
2 // Objeto Literal //
3
4 var pessoa = [
5
6 ]
7
```



0 0 ▲ ▲ ▲ ▲ Connect

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript ⚙ Go Live ⚙

JS Aula 12.js 4

C: > Users > Carol > Desktop > JS Aula 12.js >

```
1
2 // Objeto Literal
3
4 var pessoa = {
5   // Propriedades do Objeto
6   nome: 'Ana',
7   idade: 25,
8   sexo: 'F',
9   interesses: ['Ler', 'Correr'],
10 }
```

# **Propriedades**

**seguem o formato "nome: valor" e são separadas por vírgula.**



JS Aula 12.js ●

C:\Users\Carol\Desktop\JS Aula 12.js&gt;...



```
1
2 // Objeto Literal
3
4 var pessoa = {
5 // Propriedades do Objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10 }
```

11

12

13





JS Aula 12.js 2 ●

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; [e] pessoa &gt; [b] bio

1  
2 // Objeto Literal3  
4 var pessoa = {  
5 // Propriedades do Objeto  
6 nome: ["Bob", "Smith"],  
7 idade: 32,  
8 sexo: "masculino",  
9 interesses: ["música", "esquiar"],  
10 bio: function () {

11 alert(

12



13 }

14

15



16



⊗ 2 △ 0 ⏪ ⏴ Connect

Ln 12, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript ⓧ Go Live ⓧ ⓧ

JS Aula 12.js ●

C: > Users > Carol > Desktop > JS Aula 12.js > [e] pessoa

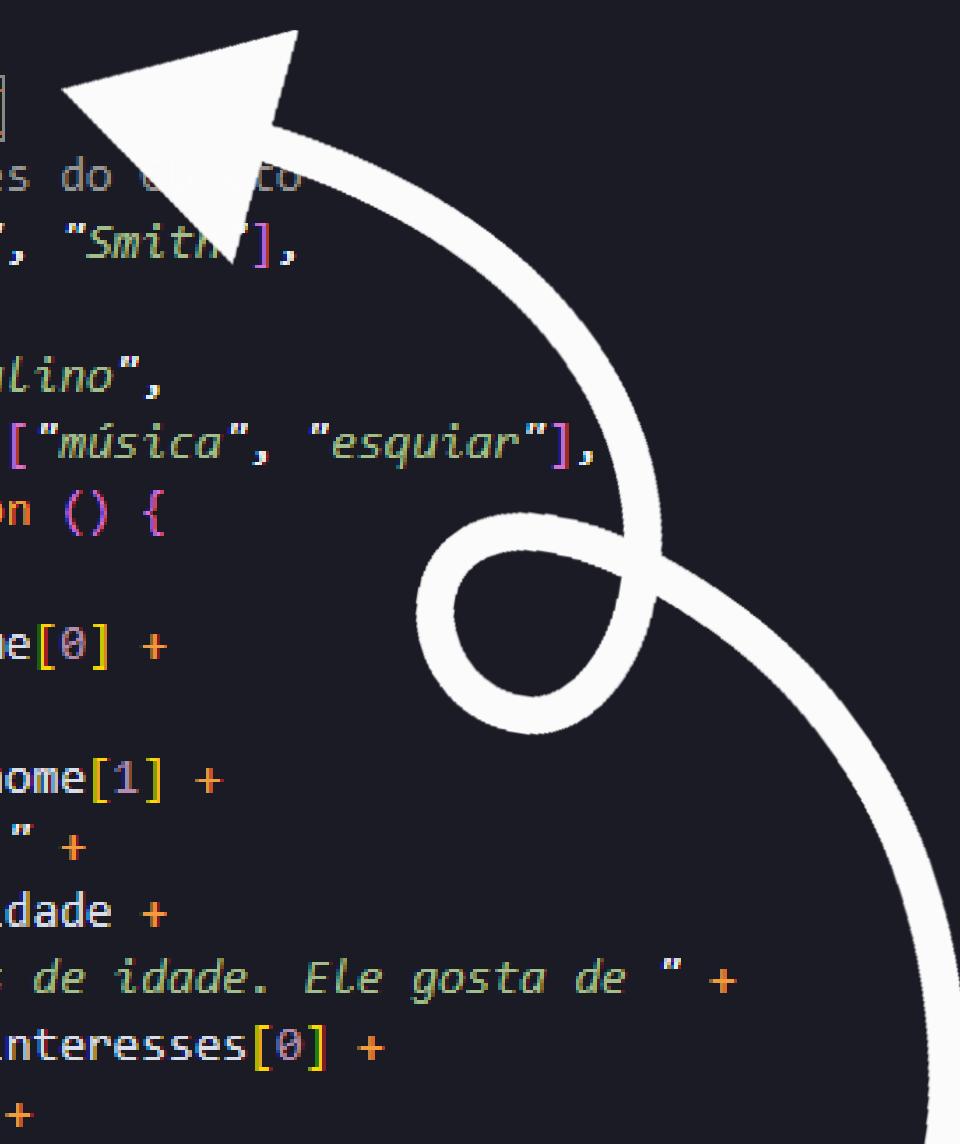
```
1
2 // Objeto Literal
3
4 var pessoa = [
5 // Propriedades do Objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10  bio: function () {
11    alert(
12      this.nome[0] +
13        " " +
14      this.nome[1] +
15        " tem " +
16      this.idade +
17        " anos de idade. Ele gosta de " +
18      this.interesses[0] +
19        " e " +
20      this.interesses[1] +
21        ".");
22  );
23 },
24 ]
```



JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > [e] pessoa

```
1
2 // Objeto Literal
3
4 var pessoa = [
5 // Propriedades do objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10  bio: function () {
11    alert(
12      this.nome[0] +
13      " " +
14      this.nome[1] +
15      " tem " +
16      this.idade +
17      " anos de idade. Ele gosta de " +
18      this.interesses[0] +
19      " e " +
20      this.interesses[1] +
21      ".",
22    );
23  },
24 ];
```



This  
se refere ao objeto atual



JS Aula 12.js

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; [?] pessoa



```
1
2 // Objeto Literal
3
4 var pessoa = [
5   // Propriedades do Objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10  bio: function () {
11    alert(
12      this.nome[0] +
13        " " +
14      this.nome[1] +
15        " tem " +
16      this.idade +
17        " anos de idade. Ele gosta de " +
18      this.interesses[0] +
19        " e " +
20      this.interesses[1] +
21        ".",
22    );
23  },
24  saudacao: function () {
25    alert("Olá! Eu sou " + this.nome[0] + ".");
26  },
27 ]
```



JS Aula 12.js

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; [?] pessoa

```
1
2 // Objeto Literal
3
4 var pessoa = [
5   // Propriedades do Objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10  bio: function () {
11    alert(
12      this.nome[0] +
13        " " +
14      this.nome[1] +
15        " tem " +
16      this.idade +
17        " anos de idade. Ele gosta de " +
18      this.interesses[0] +
19        " e " +
20      this.interesses[1] +
21        ".",
22    );
23  },
24  saudacao: function () {
25    alert("Olá! Eu sou " + this.nome[0] + ".");
26  },
27 ]
```

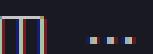
## Arrays

variável com mais de um valor



JS Aula 12.js

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; [?] pessoa

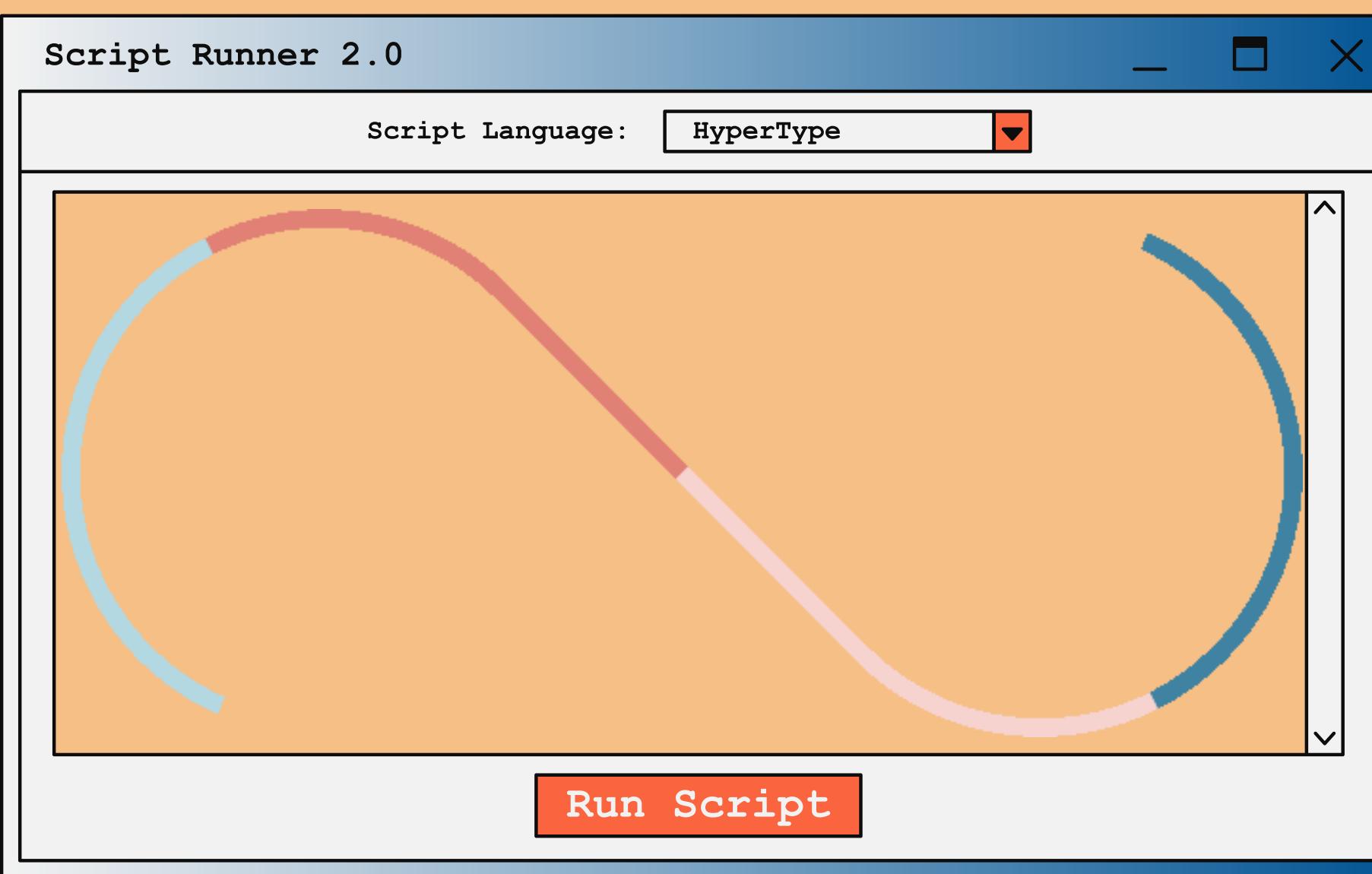


```
1
2 // Objeto Literal
3
4 var pessoa = [
5   // Propriedades do Objeto
6   nome: ["Bob", "Smith"],
7   idade: 32,
8   sexo: "masculino",
9   interesses: ["música", "esquiar"],
10  bio: function () {
11    alert(
12      this.nome[0] +
13        " " +
14      this.nome[1] +
15        " tem " +
16      this.idade +
17        " anos de idade. Ele gosta de " +
18      this.interesses[0] +
19        " e " +
20      this.interesses[1] +
21        ".",
22    );
23  },
24  saudacao: function () {
25    alert("Olá! Eu sou " + this.nome[0] + ".");
26  },
27 ]
```

## Arrays

A qual array se refere [#]

# For ... in



Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

É utilizado para loops entre as **propriedades de um objeto** e os **valores de um array**.

O processo é **mais lento** do que do For e ele **não dá prioridade** conforme a ordem das propriedades e dos elementos.

Page 1 Words: 88



JS Aula 12.js

C:\Users\Carol\Desktop\JS Aula 12.js&gt;...



```
1
2 // Objeto
3 let carro = {
4     marca: 'ford',
5     modelo: 'fiesta',
6     cor: 'prata',
7 }
```

8

9 |



JS Aula 12.js ●  
C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1
2 // Objeto
3 let carro = {
4     marca: 'ford',
5     modelo: 'fiesta',
6     cor: 'prata',
7 }
8
9 // Condicional for
10 for (propriedade in carro) {
11     console.log(propriedade);
12 }
```



1 JS Aula 12.js ●

C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1
2 // Objeto
3 let carro = {
4     marca: 'ford',
5     modelo: 'fiesta',
6     cor: 'prata',
7 }
8
9 // Condicional for
10 for (propriedade in carro) {
11     console.log(propriedade);
12 }
```

For  
declarando a repetição



JS Aula 12.js •  
C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1 // Objeto
2
3 let carro = {
4     marca: 'ford',
5     modelo: 'fiesta',
6     cor: 'prata',
7 }
8
9 // Condisional for
10 for (propriedade in carro) {
11     console.log(propriedade);
12 }
```

**variável**  
**nomeando uma variável que contem os valores  
das chaves do objeto**



1 JS Aula 12.js ●  
C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1  
2 // Objeto  
3 let carro = {  
4     marca: 'ford',  
5     modelo: 'fiesta',  
6     cor: 'prata',  
7 }  
8  
9 // Condicional for  
10 for (propriedade in carro) {  
11     console.log(propriedade);  
12 }  
13  
14  
15
```

in

**nomeando de qual objeto vem as propriedades**



[←](#) [→](#) [C](#) [Home](#)[Pesquisar no Google ou digitar URL](#)[Drive](#) [YouTube](#) [Pinterest](#) [Reddit](#) [WallHaven](#) [Rainy Mood](#) [PTI](#) [Profissionais TI](#) [Meio Bit](#) [Fontvir.us](#) [»](#) [Outros favoritos](#)

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. In the main pane, there is a code editor containing the following JavaScript code:

```
// Objeto
let carro = {
  marca: 'ford',
  modelo: 'fiesta',
  cor: 'prata',
}

// Condicional for
for (propriedade in carro) {
  console.log(propriedade);
}
```

Below the code, the output of the console.log statements is displayed:

```
marca
VM1200:11
modelo
VM1200:11
cor
VM1200:11
< undefined
>
```

The 'Console' tab is highlighted in blue, and the status bar at the bottom right shows '1 Issue: 1'.



JS Aula 12.js •

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; ...

1  
2 // Objeto3 let carro = {  
4 | marca: 'ford',  
5 | modelo: 'fiesta',  
6 | cor: 'prata',  
7 }

8



9 // Condicional for

10 for (propriedade in carro) {  
11 | console.log(carro[propriedade]);  
12 }

13

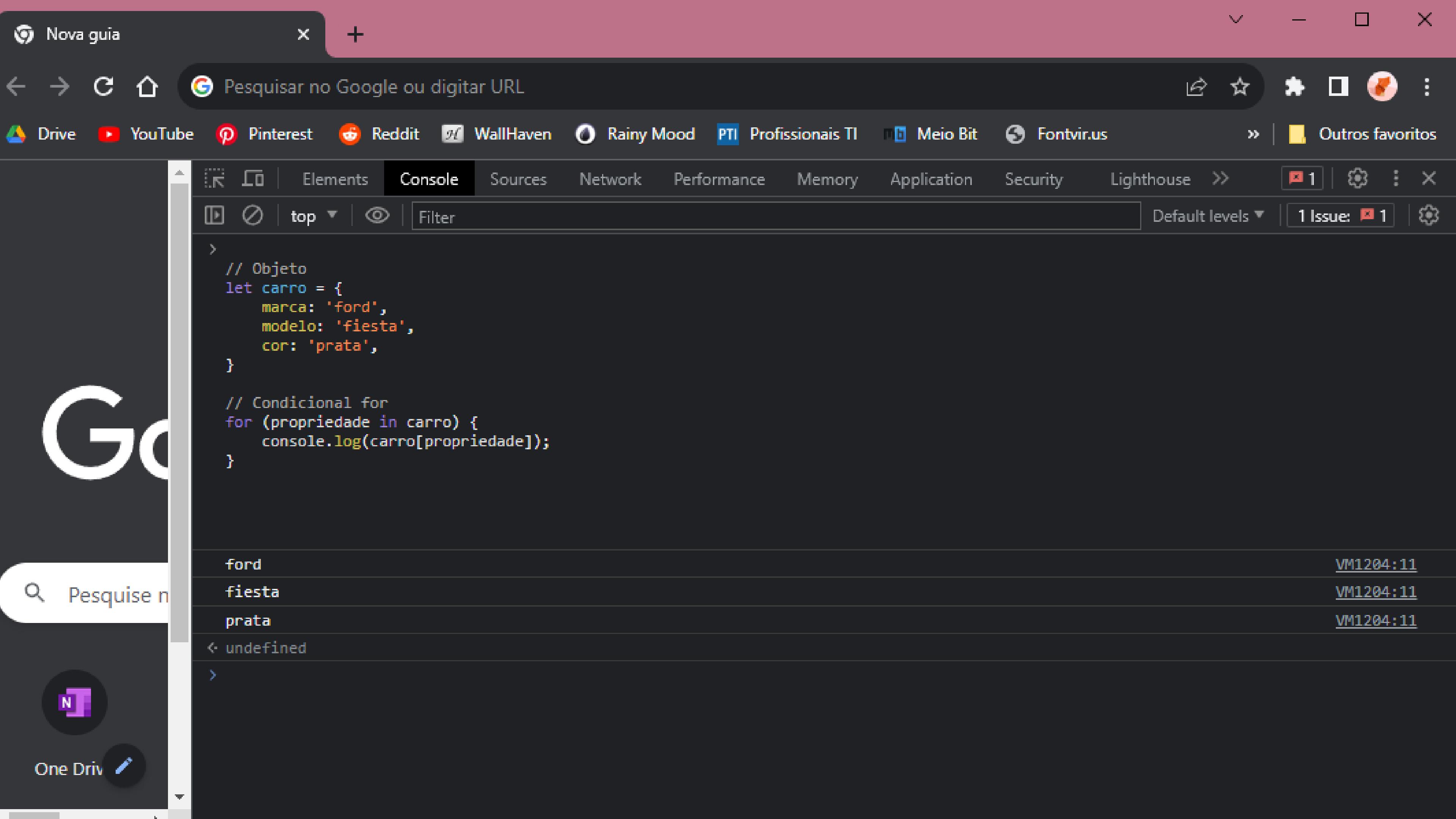
14

15

16

17





# Funções



Document2 - Macrostuff Board

São mini programas dentro de um programa - um **conjunto de instruções** que executa uma tarefa ou calcula um valor. Para usar uma função, devemos defini-la em algum lugar no escopo do qual precisa ser chamada.

Geralmente criamos funções quando precisamos que certos blocos executem as tarefas propostas em diferentes partes do código.

Page 1 Words: 88

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > addNumber

```
1
2
3 //Função adicionar números
4 function addNumeros(numero aAdd) {
5
6 }
7
```





JS Aula 12.js •

C: &gt; Users &gt; Carol &gt; Desktop &gt; JS Aula 12.js &gt; addNumeros



1

2

3 //Função adicionar números

4 function addNumeros(numberosaAdd) {

5

6 }

7

**Definindo a função**  
*ela não vai rodar ainda*





JS Aula 12.js

C:\Users\Carol\Desktop\JS Aula 12.js...



1

2

3 //Função adicionar números

4 function addNumberos(numberosaAdd) {

5 var sum = 0;

6 for (umNumber in numberosaAdd) {

7 sum = sum + numberosaAdd[umNumber];

8 }

9 return sum;

10 }

11





Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | >> | 1 | | |

| top | Filter | Default levels | 1 Issue: 1 |

&gt;

```
//Função adicionar números
function addNumberos(numberosaAdd) {
    var sum = 0;
    for (umNumber in numberosaAdd) {
        sum = sum + numberosaAdd[umNumber];
    }
    return sum;
}
< undefined
> |
```

Pesquise n



One Drive

JS Aula 12.js

C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1
2
3 //Função adicionar números
4 function addNumeros(numberosaAdd) {
5     var sum = 0;
6     for (umNumber in numberosaAdd) {
7         sum = sum + numberosaAdd[umNumber];
8     }
9     return sum;
10}
11
12 // Variáveis
13 var meusNumeros = [2,4,2,7];
14 var meusNumeros2 = [333,222,111];
15 var meusNumeros3 = [777,555,777,555]
```

JS Aula 12.js

```
C: > Users > Carol > Desktop > JS Aula 12.js > ...

1
2
3 //Função adicionar números
4 function addNumeros(numberosaAdd) {
5     var sum = 0;
6     for (umNumber in numberosaAdd) {
7         sum = sum + numberosaAdd[umNumber]
8     }
9     return sum;
10 }
11
12 // Variáveis
13 var meusNumeros = [2,4,2,7];
14 var meusNumeros2 = [333,222,111];
15 var meusNumeros3 = [777,555,777,555]
```

# **informando dados para trabalhar**



Pesquisar no Google ou digitar URL



Drive

YouTube

Pinterest

Reddit

WallHaven

Rainy Mood

PTI Profissionais TI

Meio Bit

Fontvir.us

&gt;

Outros favoritos



Elements    Console    Sources    Network    Performance    Memory    Application    Security    Lighthouse    >    1 |       :

|    top    |    Filter    Default levels    1 Issue: 1 |   

```
>

//Função adicionar números
function addNumberos(numberosaAdd) {
    var sum = 0;
    for (umNumber in numberosaAdd) {
        sum = sum + numberosaAdd[umNumber];
    }
    return sum;
}

// Variáveis
var meusNumberos = [2,4,2,7];
var meusNumberos2 = [3333,222,111];
var meusNumberos3 = [777,555,777,555];
< undefined
> |
```



JS Aula 12.js ●

C: > Users > Carol > Desktop > JS Aula 12.js > ...

```
1
2
3 //Função adicionar números
4 function addNumeros(numeroAdd) {
5     var sum = 0;
6     for (umNumber in numeroAdd) {
7         sum = sum + numeroAdd[umNumber];
8     }
9     return sum;
10 }
11
12 // Variáveis
13 var meusNumeros = [2,4,2,7];
14 var meusNumeros2 = [333,222,111];
15 var meusNumeros3 = [777,555,777,555];
16
17 // Chamando a função
18 var sum1 = addNumeros(meusNumeros);
19 var sum2 = addNumeros(meusNumeros2);
20 var sum3 = addNumeros(meusNumeros3);
21
22 // Exibir conteúdo
23 console.log(sum1);
24 console.log(sum2);
25 console.log(sum3);
26
```

 Pesquisar no Google ou digitar URL[Drive](#) [YouTube](#) [Pinterest](#) [Reddit](#) [WallHaven](#) [Rainy Mood](#) [PTI](#) [Profissionais TI](#) [Meio Bit](#) [Fontvir.us](#) [»](#) [Outros favoritos](#)

Elements    Console    Sources    Network    Performance    Memory    Application    Security    Lighthouse    >>    1    ⚙    :

Filter    Default levels    1 Issue: 1    ⚙

```
function addNumeros(numberosaAdd) {  
    var sum = 0;  
    for (umNumber in numberosaAdd) {  
        sum = sum + numberosaAdd[umNumber];  
    }  
    return sum;  
}  
  
// Variáveis  
var meusNumeros = [2,4,2,7];  
var meusNumeros2 = [333,222,111];  
var meusNumeros3 = [777,555,777,555];  
  
// Chamando a função  
var sum1 = addNumeros(meusNumeros);  
var sum2 = addNumeros(meusNumeros2);  
var sum3 = addNumeros(meusNumeros3);
```

```
// Exibir conteúdo  
console.log(sum1);  
console.log(sum2);  
console.log(sum3);
```

```
15
```

VM54:23

```
3666
```

VM54:24

```
2664
```

VM54:25

```
< undefined
```

```
> |
```



## Welcome to Software Install Wizard!

This Wizard  
on your comp

It is strong  
programs bef

To continue,



< Back

Next >

Cancel