

Objetos

Labenu_



O que vamos ver hoje?

- Objetos
- Acessando e alterando propriedades de um objeto



Objetos

Labenu_



Objetos

- **Objetos** são estruturas de sintaxe que nos permitem representar **dados mais complexos** de uma maneira mais **organizada**
- Com os objetos conseguimos criar **modelos do mundo real** de forma mais intuitiva/humanizada



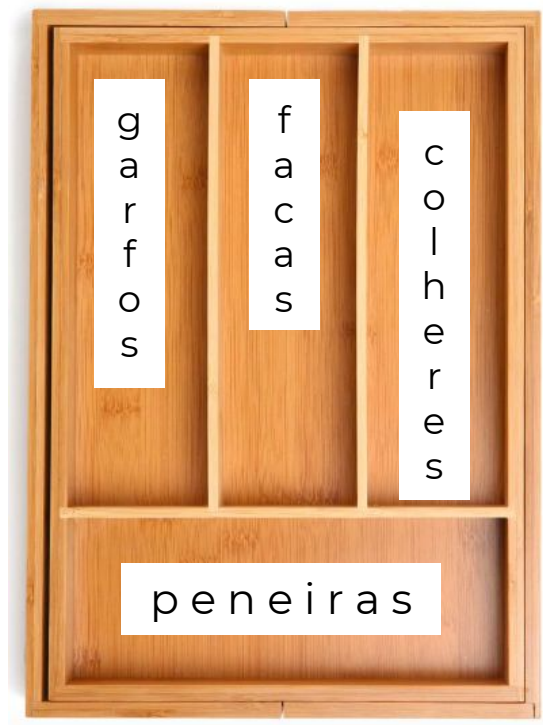
Objetos

- Se fizéssemos uma comparação com a cozinha, as variáveis com valores dos tipos: **string**, **number** e **boolean** seriam gavetas pequenas e simples



Objetos

- Os **objetos** seriam uma gaveta maior com um organizador de talheres dentro, onde cada separação possui uma **etiqueta** com um nome de fácil identificação



Objetos

- As propriedades dos objetos podem assumir **quaisquer valores**
 - String, number, boolean, array, etc.
 - Funções (neste caso, são chamados de **método**)



Estrutura de um objeto



propriedade			chave
nome	idade	email	
"Amanda Rangel"	27	"mandinha_rock@gmail.com"	valor

instrutora



Sintaxe da criação um objeto

```
const instrutora = {  
  nome: 'Amanda Rangel',  
  idade: 27,  
  email: 'mandinha_rock@gmail.com'  
}
```

-  objeto
-  chave
-  valor

Vamos ver na prática! 



Estrutura de um objeto



		propriedade	método	
nome	frontender	responsabilidades	contaPiada()	chave
"Chijo"	true	"Dar aula", "Criar novas aulas", "Causar"	console.log("É pavê ou pa comer?")	valor

instrutora



Sintaxe da criação um objeto

```
const instrutora = {  
  nome: 'Chijo',  
  frontender: true,  
  responsabilidades: ["Dar aula", "Criar novas aulas", "Causar"],  
  contaPiada: () => {  
    console.log("É pa ver ou pa comer?")  
  }  
}
```

- objeto
- chave
- valor

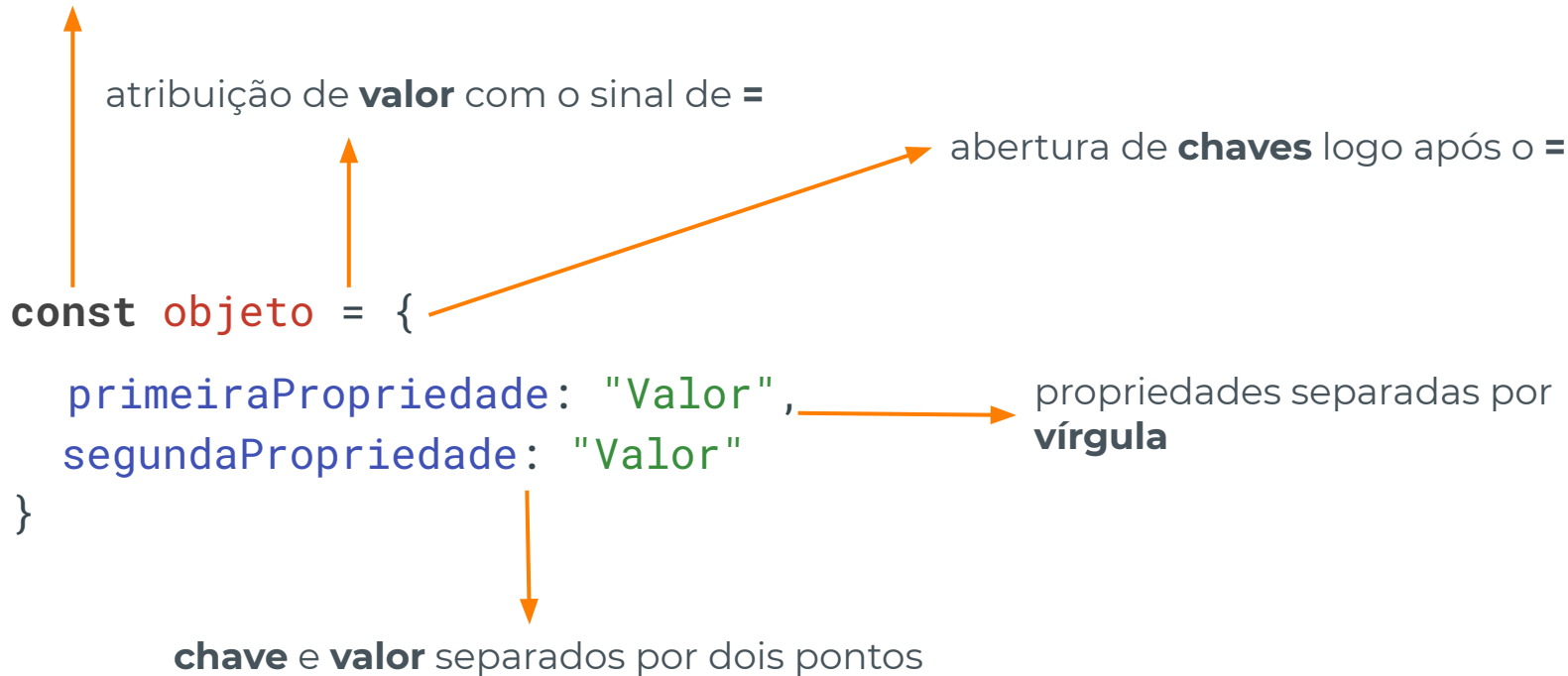
Vamos ver na prática! 



Estrutura padrão de um objeto



declaração com `let` ou `const`
seguido do **nome** do objeto



Acessando e alterando valores do objeto

Labenu_



Acessando e alterando propriedades

- Para **acessar** ou **alterar** as **propriedades** dos objetos, há duas sintaxes interessantes:
 - Notação do **ponto** (a mais "comum" entre as linguagens de programação)
 - Notação dos **colchetes**



Acessando e alterando propriedades

- **Notação do Ponto**

```
const instrutora = {  
  nome: "Amanda Rangel",  
  idade: 27,  
  email: 'mandinha_rock@gmail.com'  
}
```

- objeto
- chave
- valor

```
const nomeDaInstrutora = instrutora.nome;  
console.log(nomeDaInstrutora)
```

Vamos ver na prática! 



Acessando e alterando propriedades

- Notação de Colchetes

```
const instrutora = {  
  nome: "Amanda Rangel",  
  idade: 27,  
  email: 'mandinha_rock@gmail.com'  
}
```

● objeto

● chave

● valor



string com a
chave

```
const nomeDaInstrutora = instrutora["nome"];  
console.log(nomeDaInstrutora)
```

Vamos ver na prática! 



Acessando e alterando propriedades

- Redefinindo valores

```
const instrutora = {  
  nome: "Amanda Rangel",  
  idade: 27,  
  email: 'mandinha_rock@gmail.com'  
}
```

- objeto
- chave
- valor

```
instrutora.nome = 'Lais';  
instrutora['email'] = 'petra_cervejaria@gmail.com';
```

Vamos ver na prática! 





Exercício 1

- Crie um objeto que represente um filme. Ele deve ter dados da direção, o nome, o ano de lançamento, uma lista com o elenco e uma propriedade que diga se você já viu ou não.
- Acesse e imprima no console cada uma das propriedades: metade usando notação do ponto e a outra metade com notação de colchetes.



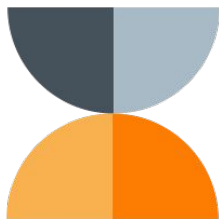


Exercício 2

- Crie um objeto que represente uma pessoa. Essa pessoa precisa ter nome, idade, gênero musical preferido.
- Imprima no console as propriedades desse objeto, seguindo o modelo abaixo:

"O nome da pessoa é ____, ela tem ____ anos e gosta muito de ____."





Pausa para relaxar 🧘

10 min

- Objetos são estruturas que permitem a **representação** do mundo à nossa volta de uma maneira **mais intuitiva**
- Possuem **propriedades** com **chave e valor**
- Para acessar o conteúdo de dentro do objeto, existem as sintaxes do **ponto** e dos **colchetes**



Acessando valores diferentes 🧐

Labenu_



Acessando valores diferentes 🤪

- Não é incomum a existência de objetos dentro de objetos, objetos dentro de arrays, arrays de objetos...
- Pode parecer complicado, mas fica mais simples se pensarmos em **caminhos**



Acessando objetos dentro de objetos 🤔

```
const paiDePet = {  
  nome: "Caio Teixeira",  
  pet: {  
    nome: "Lupin",  
    raca: "Salsicha",  
    idade: 1  
  }  
}
```

- objeto
- chave
- valor

```
const pet = paiDePet.pet  
console.log(pet)  
const nomeDoPet = pet.nome  
console.log(nomeDoPet)
```



Acessando objetos dentro de objetos 🤔

```
const paiDePet = {  
  nome: "Caio Teixeira",  
  pet: {  
    nome: "Lupin",  
    raca: "Salsicha",  
    idade: 1  
  }  
}
```

- objeto
- chave
- valor

```
const nomePet = paiDePet.pet.nome  
console.log(nomePet)
```



Acessando arrays dentro de objetos 🤔

```
const curso = {  
  nome: "Noturno Frontend",  
  linguagens: ["JS", "CSS", "HTML"]  
}
```

- objeto
- chave
- valor

```
const primeiraLinguagem = curso.linguagens[0]  
console.log(primeiraLinguagem)
```



Acessando arrays dentro de objetos 🤔

```
const curso = {  
  nome: "Noturno Frontend",  
  linguagens: ["JS", "CSS", "HTML"]  
}
```

- objeto
- chave
- valor

```
const linguagens = curso.linguagens  
console.log(linguagens)  
const primeiraLinguagem = linguagens[0]  
console.log(primeiraLinguagem)
```



Array de objetos 🤔

```
const instrutoras = [  
  {nome: "Lais", modulo: 1},  
  {nome: "Amanda", modulo: 2},  
  {nome: "Chijo", modulo: 3}  
]
```

- objeto
- chave
- valor

```
const nomeInstrutora = instrutoras[0].nome  
console.log(nomeInstrutora)
```



Adicionando propiedades

Labenu_



Adicionando propriedades

- Para **adicionar propriedades** aos objetos, podemos usar a seguinte sintaxe

```
const curso = {  
  nome: "Noturno Frontend",  
  linguagens: ["JS", "CSS", "HTML"]  
}
```

- Notação de ponto: `curso.numeroEstudantes = 50`
- Notação de colchetes: `curso['numeroEstudantes'] = 50`



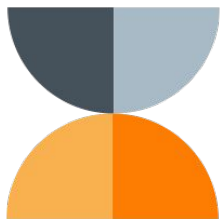


Exercício 3

- Adicione ao objeto do exercício 1 uma lista com três nomes de personagens do filme.
 - Acesse e imprima no console cada pessoas do elenco.
 - Altere a primeira pessoa do elenco por "Xuxa".
 - Imprima no console as propriedades desse objeto, seguindo o modelo abaixo:

"O nome do filme é _____. Seus personagens são ____, ____ e ____"





- Podemos propriedades dos objetos que são arrays ou outro objetos
- Para acessar esses valores seguimos o caminho, usando a notação de pontos (ou colchetes) e a posição dos elementos no array (ex: [0])
- Podemos adicionar novas propriedades ao objeto: `objeto.novaPropriedade = 'Valor'`



Espalhamento ou Spread

Labenu_



Espalhamento ou spread 🇧🇷

- Existe uma sintaxe interessante, através da qual conseguimos realizar uma **cópia inteira de um objeto para outro** e mudar (ou adicionar), se quisermos, apenas uma ou outra propriedade
- Essa sintaxe é chamada de **espalhamento (ou spread)**



Espalhamento ou spread

- Espalhamento - objetos

```
const usuario = {  
  nome: 'Astrodev',  
  idade: 25,  
  email: 'astrodev@labenu.com.br'  
}
```

```
const novoUsuario = {  
  ...usuario,  
  nome: 'Caio',  
  sobrenome: 'Teixeira'  
}
```

Vamos ver na prática! 



Espalhamento ou spread

- Espalhamento - arrays

```
const listaDeNomes = ["Pedro", "Chijo", "Amanda"]
```

```
const copiaListaDeNomes = [...listaDeNomes]
```

Vamos ver na prática! 





Exercício 4

- Crie uma função que receba um objeto de pessoa e retorne um novo objeto mantendo as propriedades originais e acrescentando ao objeto original:
 - Uma propriedade com a **lista** de suas comidas preferidas
 - e outra propriedade que seja um **objeto**, com nome e idade, que represente o melhor amigo da pessoa.
- Depois, imprima no console as propriedades desse objeto seguindo o modelo abaixo:

"O nome da pessoa é ____ e suas comidas preferidas são ____, ____ e _____. Seu melhor amigo se chama ____ e tem ____ anos"



Resumo

Labenu_



Resumo

- **Objetos** são uma sintaxe que permite que a gente modele o mundo real de uma maneira mais fiel
- Os objetos possuem **propriedades**, que possuem **chave** e **valor**
- O valor das propriedades pode ser qualquer tipo, inclusive **funções**



Resumo

- Se o valor da propriedade é uma função, chamamos ela de **método** do objeto
- Conseguimos fazer uma cópia do objeto, ou então acessar só algumas das propriedades dele utilizando as sintaxes de **spread**. O mesmo vale para arrays



Dúvidas? 🧐

Labenu_





Obrigado(a)!