INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA

JavaScript
etores e Objetos

JavaScript

Vetores e Objetos

Prof. Mr. Fabio Abrantes Diniz
fabio.abrantes.diniz@gmail.com

Array

Array

- Um **objeto especial** usado para armazenar uma **coleção ordenada** de valores
  - **typeof** [“fabio”] retorna "**object**“
  - Usar **Array.isArray()**
    - ✓para verificar se é mesmo um array:

- Indexados **sequencialmente**
  - O primeiro elemento está na posição 0

- Cada valor é chamado de ***elemento*** - P osição numérica conhecida como ***índice***

Array

Array

- Dentro do array pode ter mistura de tipos de dados
  - **Number**, **String**, **Objetos**, **booleanos** ou outros **arrays**

```
let misto = ["texto", 10, true];
```

- Os arrays em JavaScript são *dinâmicos*:
  - Pode crescer ou diminuir de tamanho

- Rico de métodos de manipulação de array
  - como **map**, **filter**, **push**, etc.

Array

Array

- **Criando um array:**

  - **let** meuArray = **new** Array(5,'casa',2,false,1.2);
  - **let**

meuArray = [1.1,true,”a”, {x:1},[1,2,3]];

```
var count = [1,,3];   // Elementos nos índices 0 e 2. count[1] => undefined
var undefs = [,,];    // Array sem elementos mas com comprimento 2
```

```
let numeros = [1, 2, 3, 4, 5];
let misto = ["texto", 10, true];
let vazio = new Array(); // menos comum
```

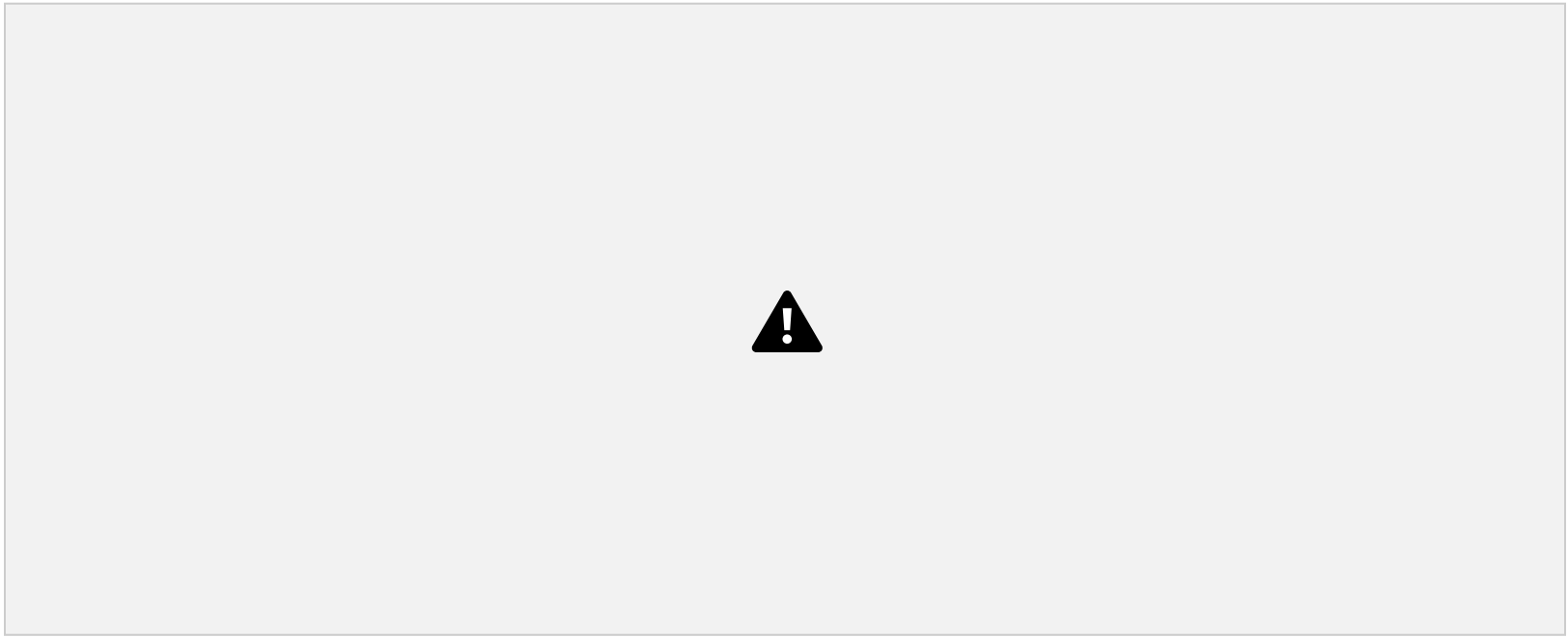
Array

- **Adicionando e Removendo**

elementos no array:

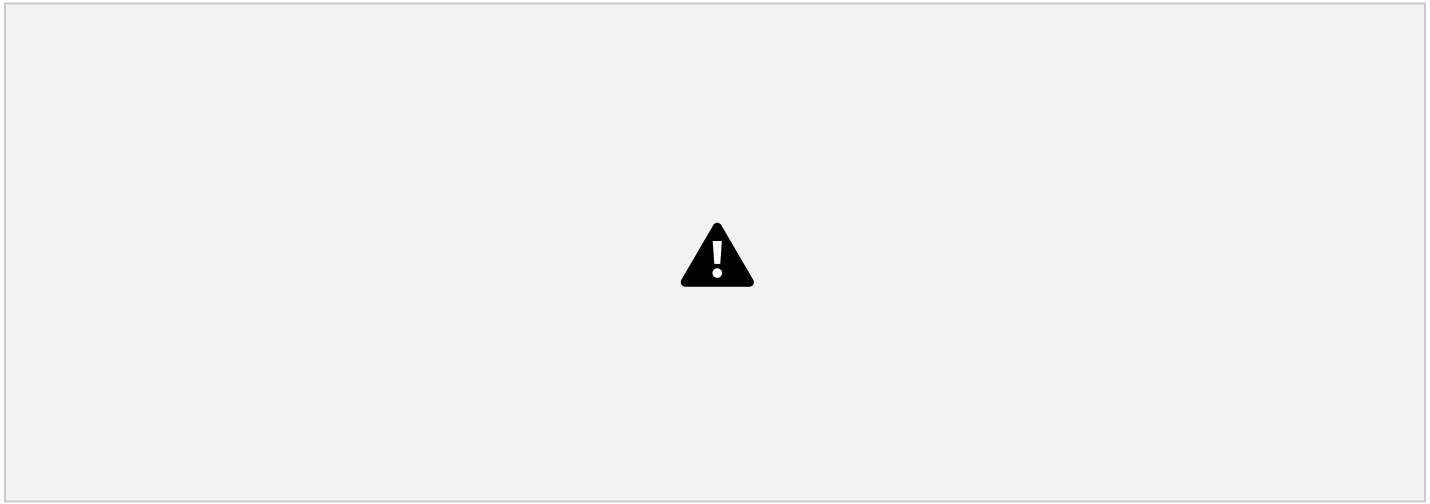
✓Adicionando: **push()** e **unshift()**

✓Removendo: **pop()** e **shift()**



Array

- **Acessando um valor do array**
  - Usando os índices nos colchetes
  - Propriedade **length**

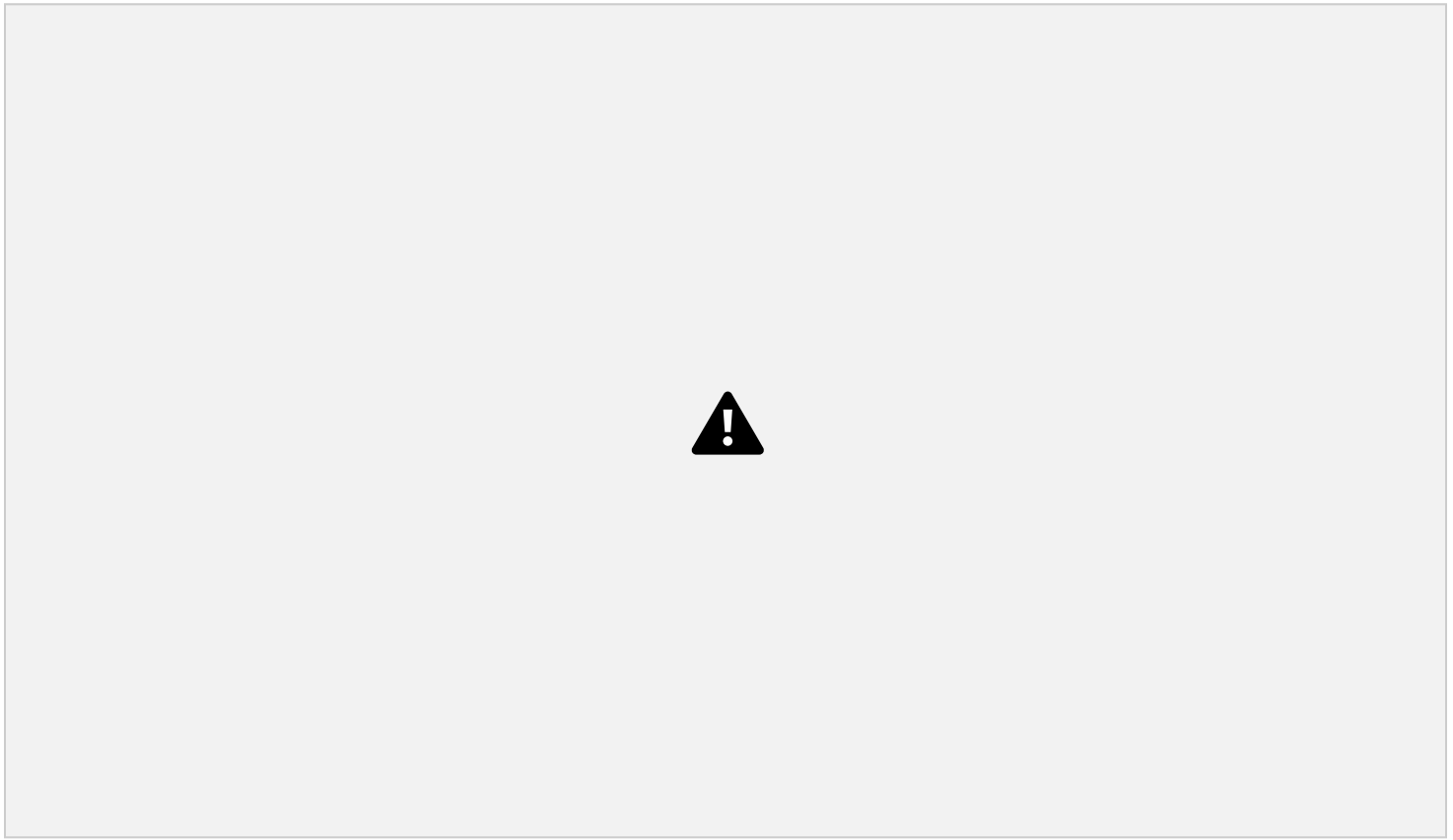


Array

- **Acessando uma propriedade de objeto dentro do array**
  - **E realizando**

multidimensionais

- **Um array dentro de outro array**



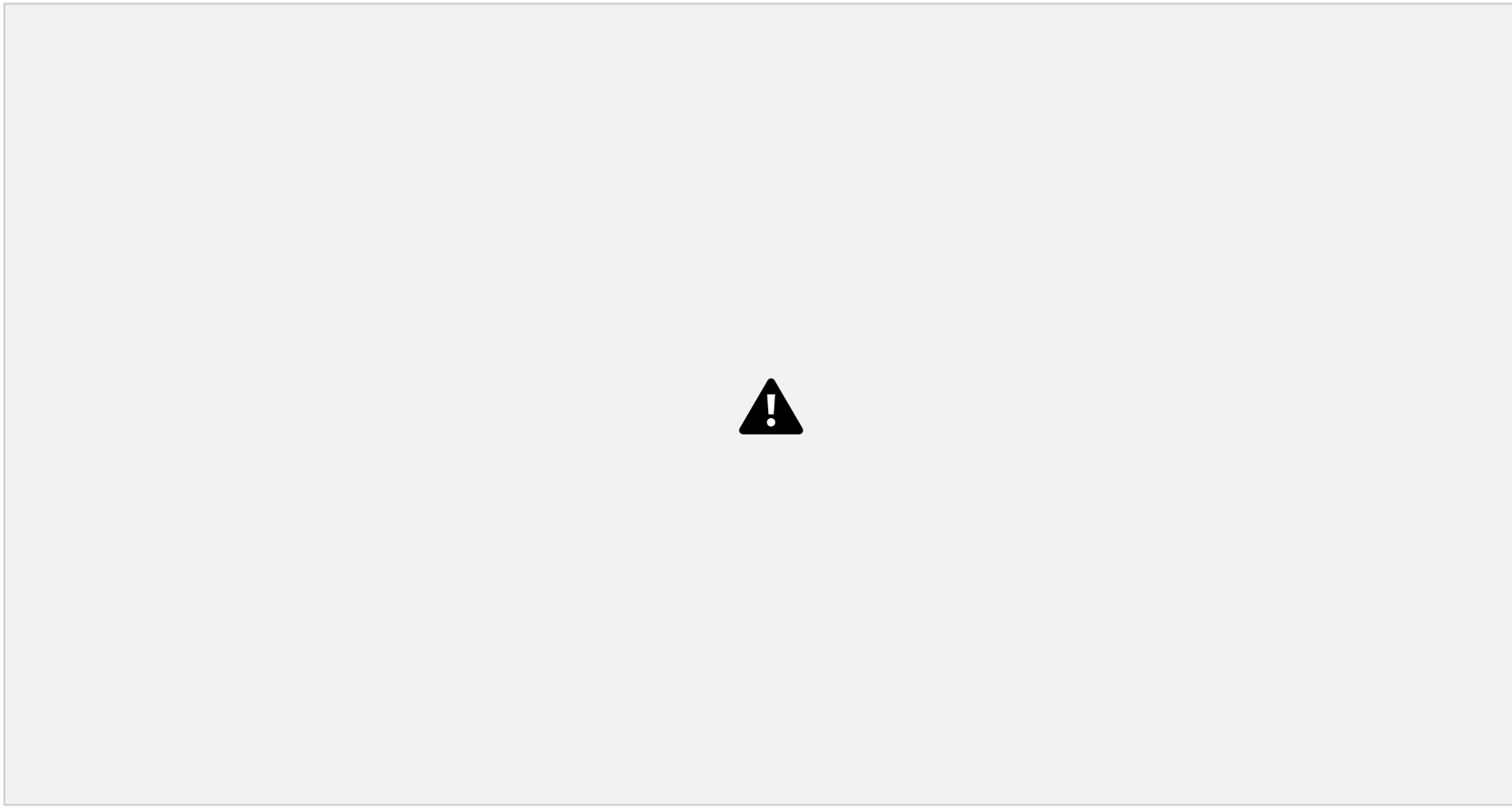
Atividade prático em aula

ARRAY

- **Iterando em todos os elementos do Array**
  - Usando o **loop for, while**

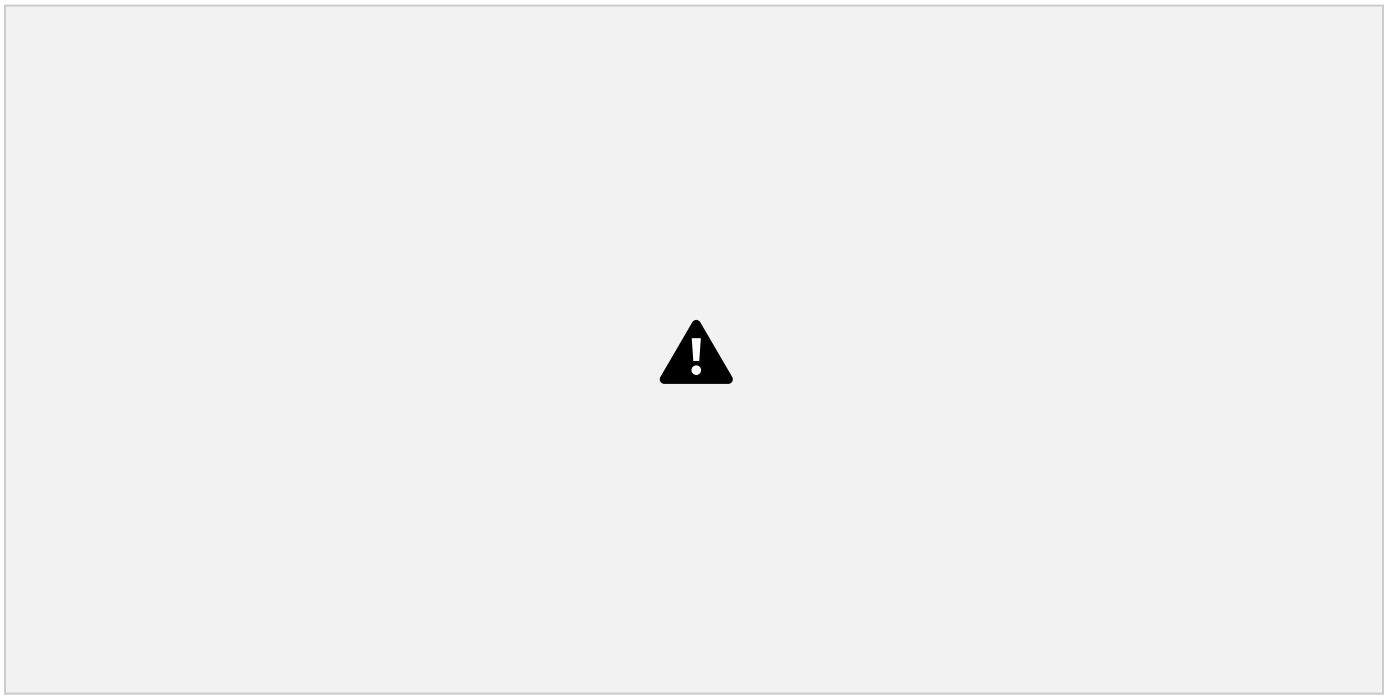
ARRAY

- **Iterando em objetos dentro do Array**



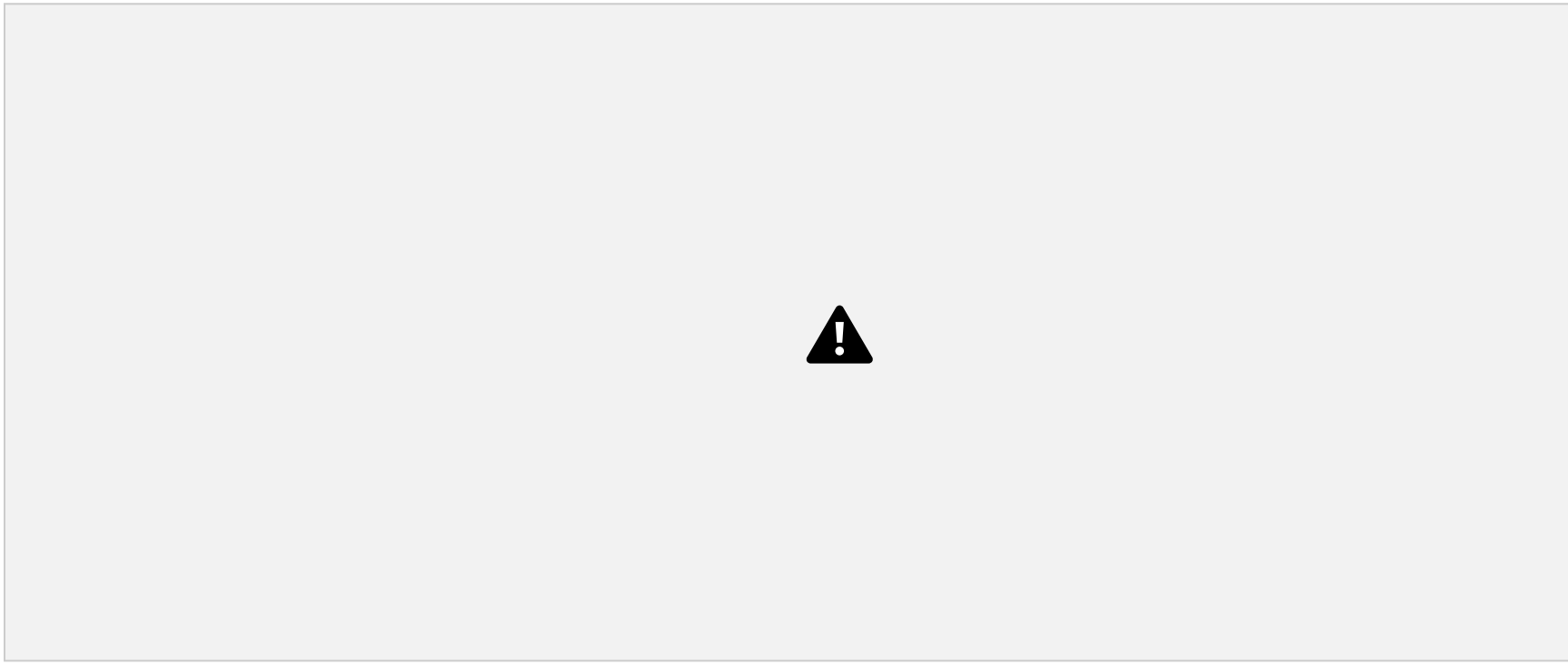
Objetos

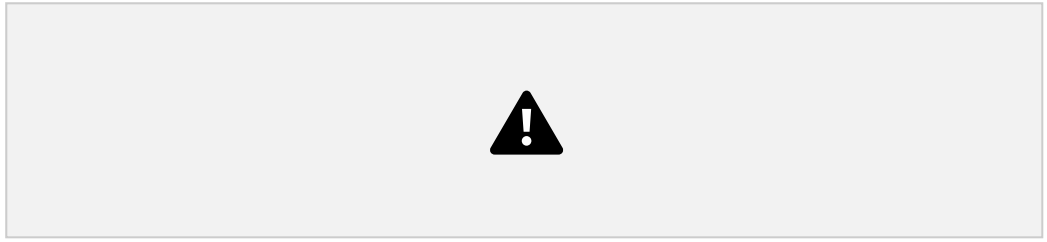
- Uma estrutura que contém
  - coleção de dados e funcionalidades
    - ✓ Propriedades e métodos
    - – organizados em pares **chave: valor**



Objetos

- São como objetos na vida real
  - São entidades independentes com propriedades e tipo.

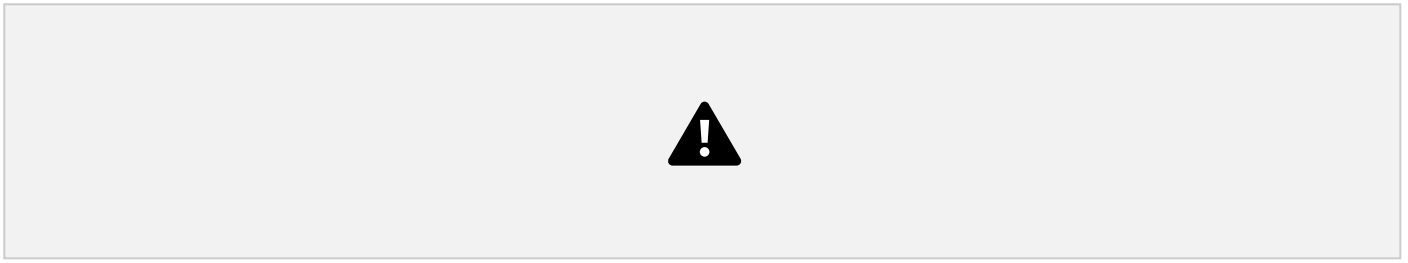




Tipos de Objetos

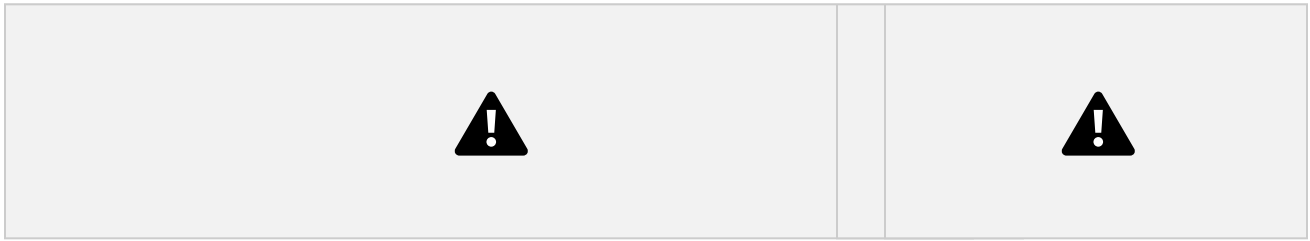
- **Um *objeto nativo***
  - Um objeto feito pelo ECMAScript.
    - ✓Ex: Arrays, funções, datas ...
- **Um objeto do navegador**
  - Elementos do Html ( )
- **Definido pelo usuário**

Objetos Nativos

- Objetos que **fazem parte da linguagem JavaScript**,
  - definidos pela especificação **ECMAScript**.
  - disponíveis em qualquer ambiente JavaScript
    - ✓(navegador, Node.js, etc).
- **Exemplos:**
  - 

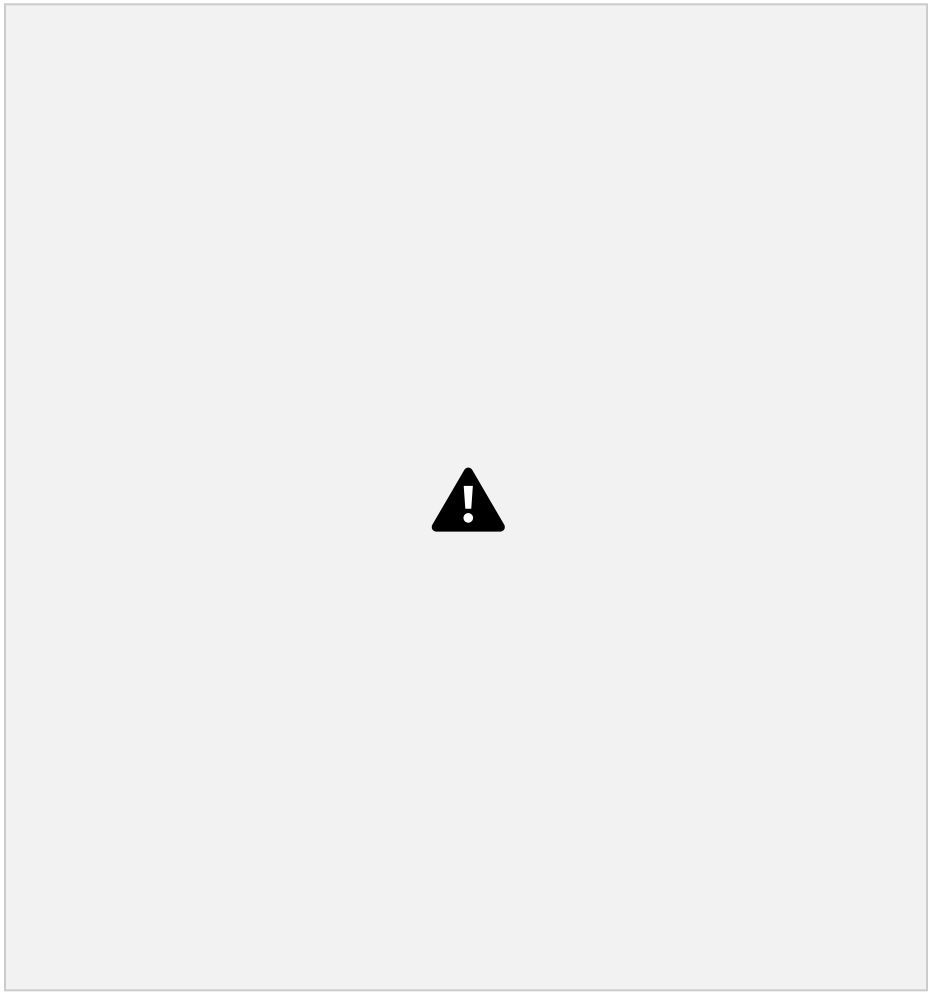
    Array

- Date
- Math
- JSON
- String, Number e Boolean



Objetos Nativo -JSON

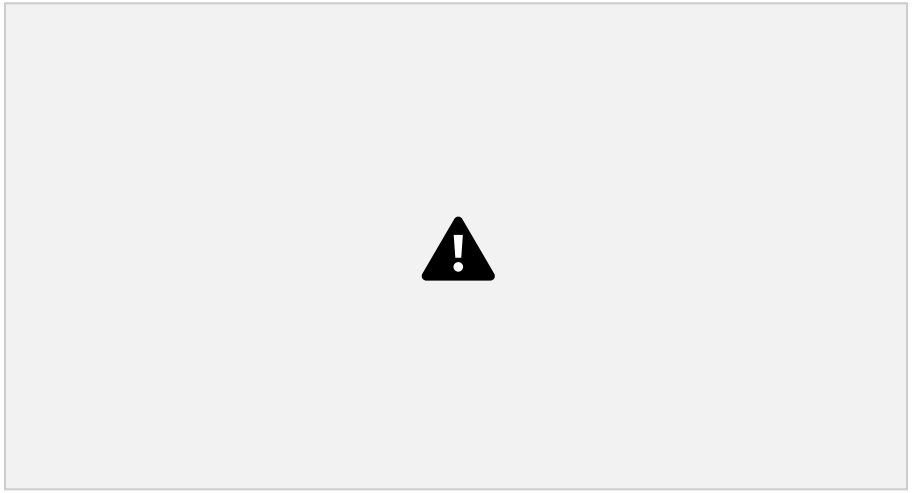
- é um **formato leve de troca de dados**, baseado em texto, muito usado em APIs.

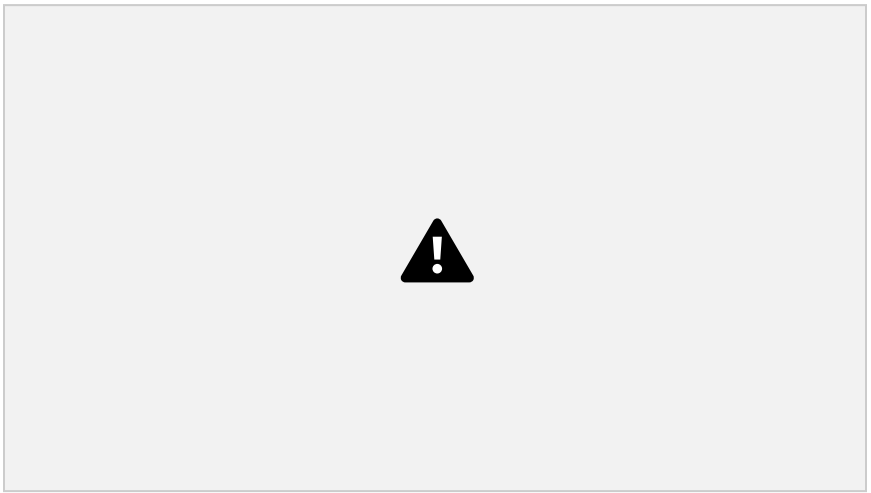


Objetos

Nativos -JSON

Conversão entre Objeto <-> JSON





Objeto do Navegador

- Objetos que **o navegador fornece**

Permitir interação com o ambiente da página

✓DOM, eventos,

✓**Não existem** no Node.js puro.

- **Exemplos**

- Windows, document, navigator, localstorage, cookies, fetch, history, location

Objeto criado pelo desenvolvedor

- São os objetos que você mesmo define no seu código para organizar dados.
  - Como podemos criar?
    - ✓Objetos literais
      - Usando **{}**
    - ✓Usando **New Object()**

- Menos usado atualmente

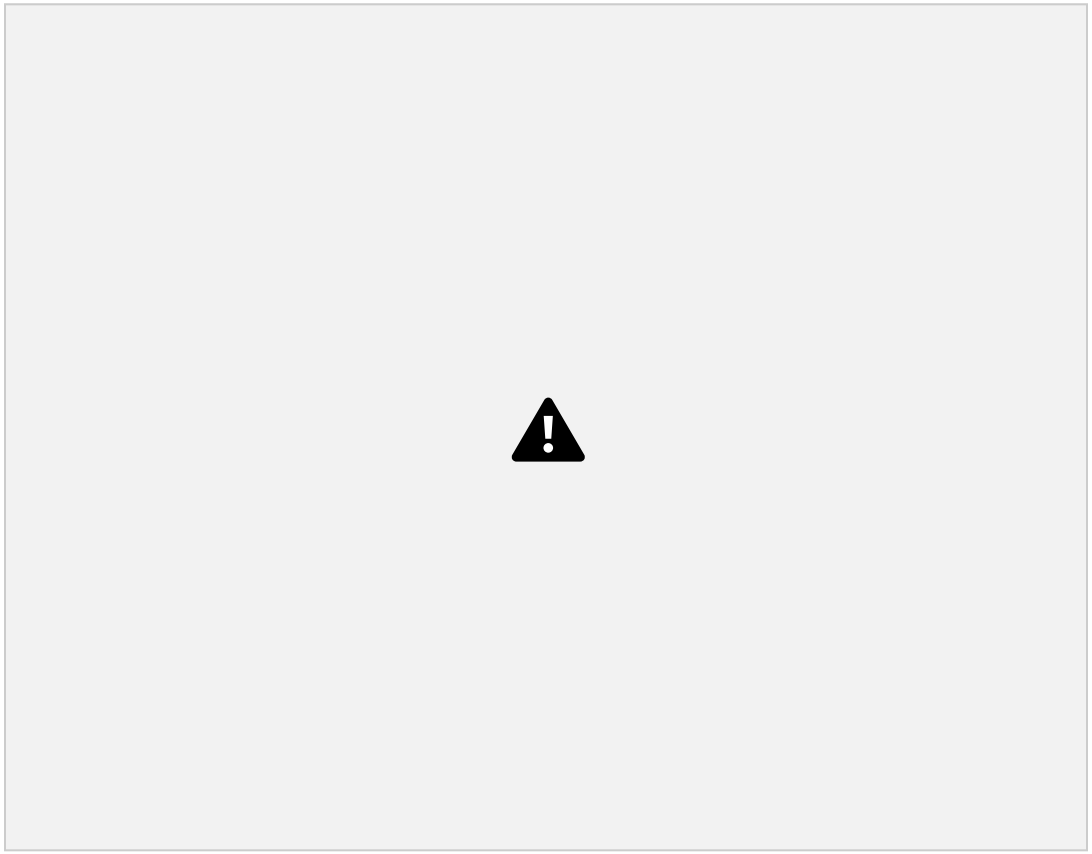
✓Funções construtoras (**function)**

- **new**

✓Usando **Class** (ES6+)

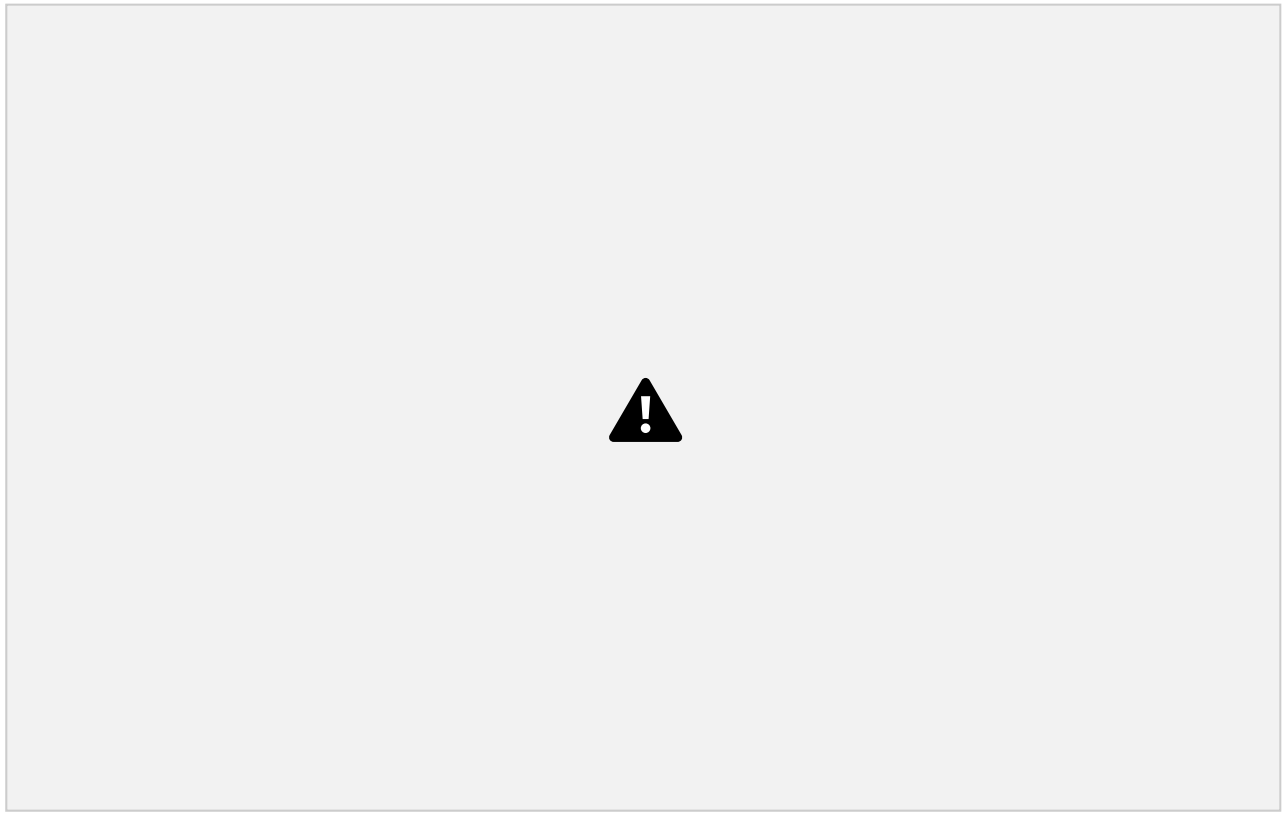
- Forma moderna de criar objetos com herança.

Objeto criado pelo desenvolvedor • **Objetos literais**



Objeto criado pelo desenvolvedor

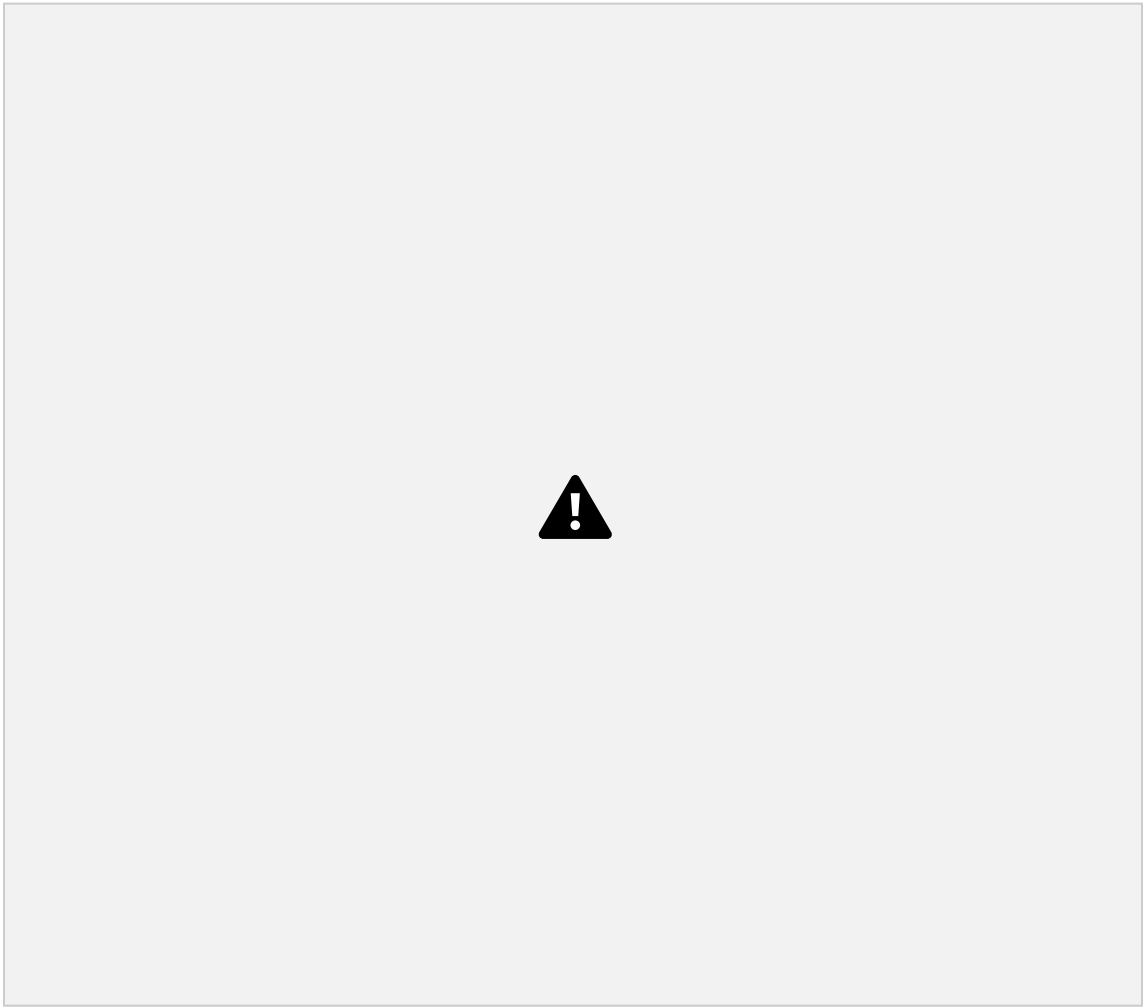
- Funções construtoras (**function)**

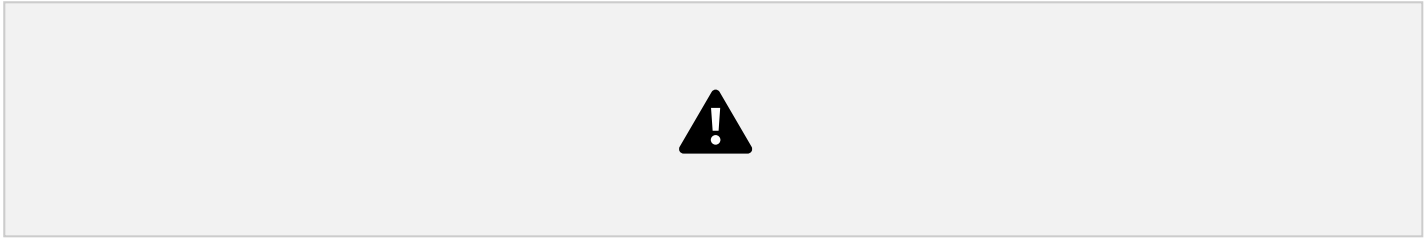


✓ .

Objeto criado pelo desenvolvedor

- Usando **Class** (ES6+)



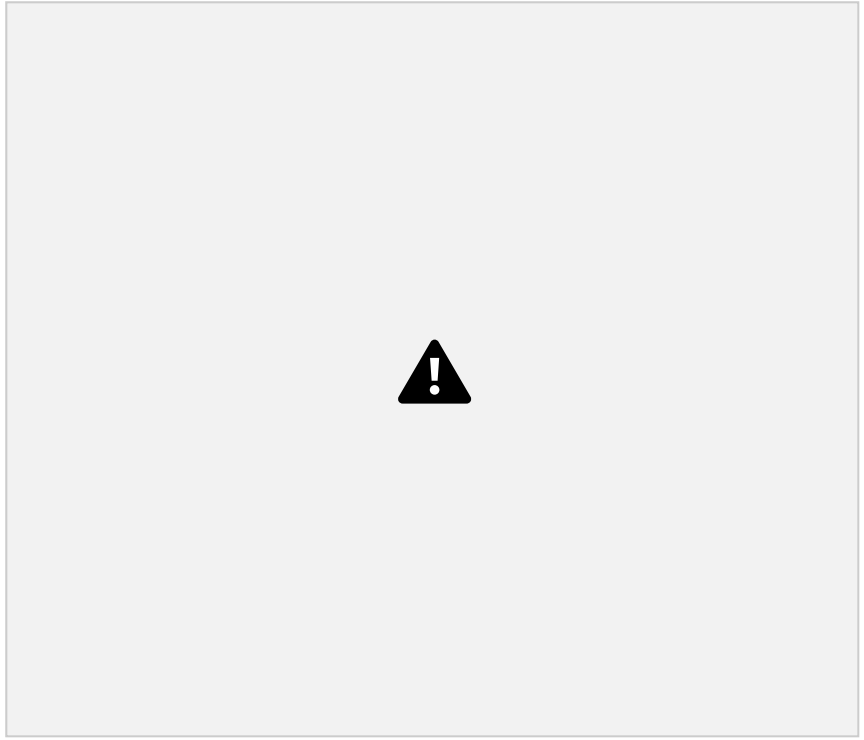


Manipulação de Objetos

Acessando propriedades

- Usando o ponto entre nome do objeto e propriedade

- usando-se a notação de **colchete [“nomeDaChave”] .**



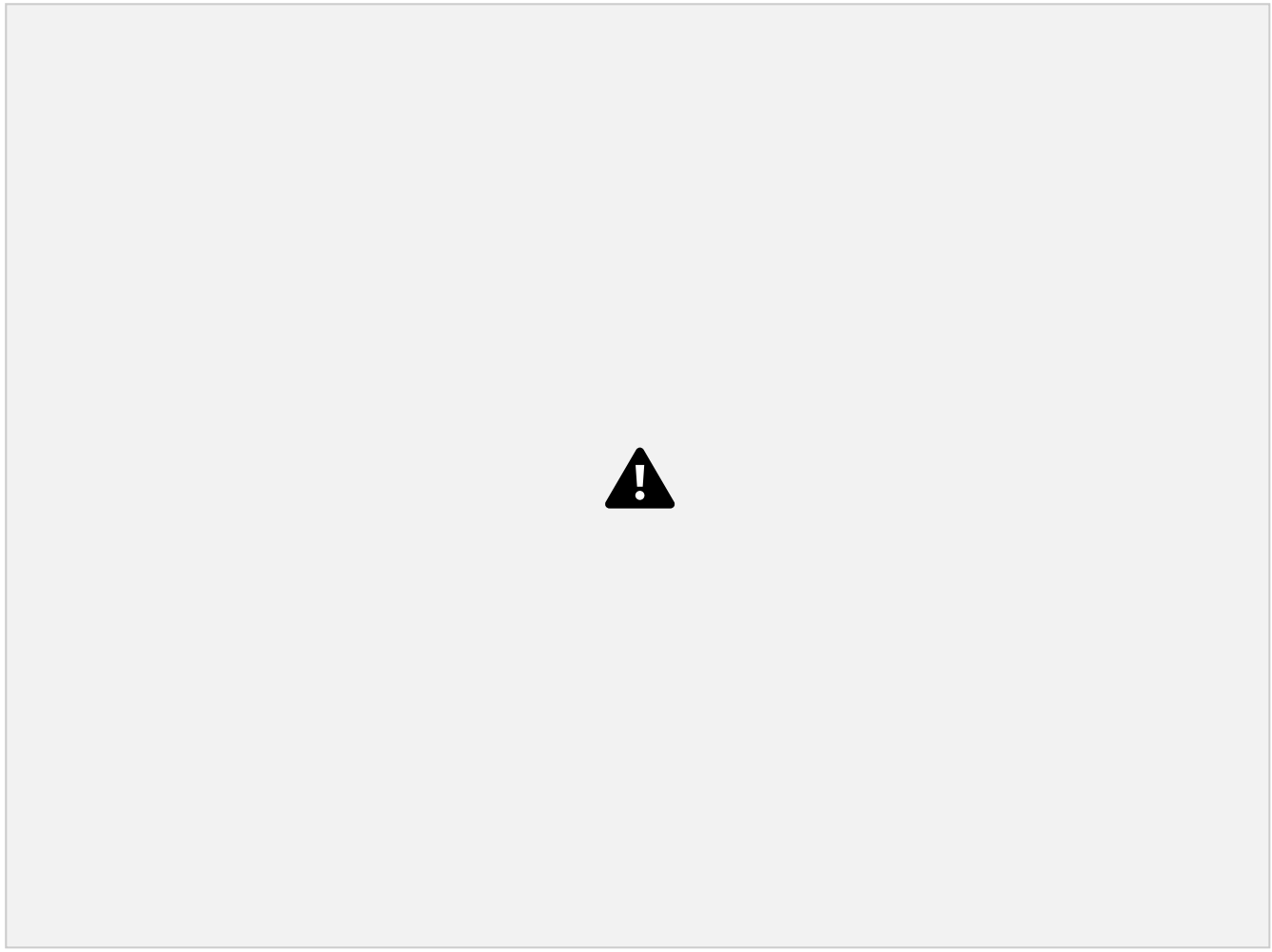
Manipulação de Objetos

Alterando valores das props

- Adicionando novas props

- Removendo props

Iterando sobre os objetos



Iterando sobre os objetos

Object.Keys()

- método retorna uma Array dos nomes das propriedades do próprio objeto

Métodos de objetos

- Duas sintaxes mais usadas (com funções anônimas e sem function)

Métodos de objetos

- **Referenciando uma função fora do objeto**