

FOCUS

# JavaScript

Semana Acadêmica - 2025.2

# Aviso!

---

- Em diversos exemplos ao longo do treinamento, misturei inglês e português, não façam isso!
- Escolham em conjunto um idioma de preferência e permaneçam nele durante o desenvolvimento de **todo o projeto**.



# Aviso!

---

- Temos aqui pessoas de diversos períodos do curso.

Quaisquer questionamentos ou conceitos que não estejam familiarizados, avisem e tirarei as dúvidas que surgirem.



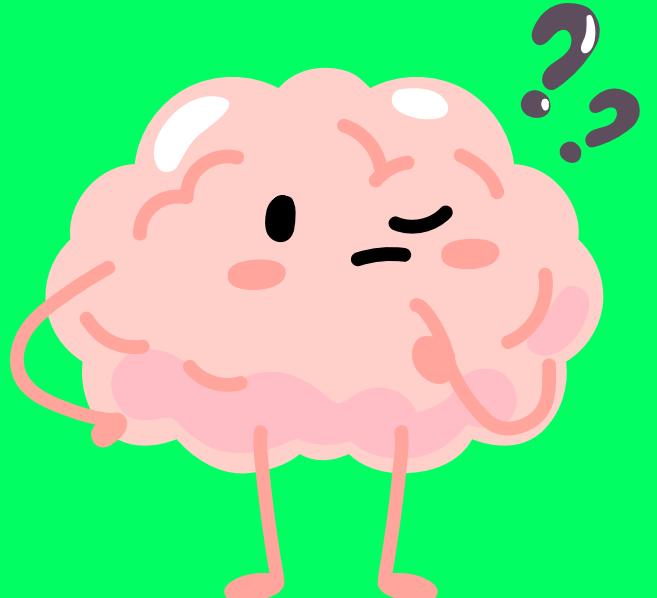
# Tópicos do Minicurso

---

- HTML & CSS
- JavaScript
- APIs (criação e consumo)
- Front-end/back-end: noções básicas
- Git & GitHub



# Alguém aí já teve experiência com JS?



# Sobre o JS

---

- Linguagem simples
- A maioria dos navegadores modernos suporta JavaScript
- Funções nativas para manipulação de páginas web
- Presente tanto no Front-End quanto no Back-end (NodeJS)
- Sem necessidade da declaração explícita de tipos
- Java != JavaScript

# Variáveis e Tipos

---



```
1 // não poderá mais ser modificada
2 const mol = 6e-23
3 // pode ser modificada posteriormente
4 let acumulador = 0
5 // NÃO USE ESSA DECLARAÇÃO!
6 var variavel = "teste"
```

**Use os identificadores!!!**



```
1 // Tipos primitivos: String, Number, Boolean,
  Null e Undefined
2 const inteiro = 12
3 const fracionario = 12.42
4 const string = "breno"
5 const nulo = null
6 const indefinido = undefined
7 const verdadeiro = true
8
9 console.log(typeof inteiro) // number
10 console.log(typeof fracionario) // number
11 console.log(typeof verdadeiro) // boolean
12 console.log(typeof string) // string
```

# Variáveis e Tipos

---

JavaScript não tem declaração explícita de tipos



```
1 console.log(1 + 1)
```



```
1 console.log("1" + 1)
```



```
1 const parametro1 = "1"
2 const parametro2 = 1
3
4 console.log(
5   Number(parametro1) + Number(parametro2)
6 )
```

# Arrays

---

Arrays no JavaScript são como vetores C, mas com uma boa dose de esteróides.

```
1 const array = [5, 9, 7, 1, -2, 8, 5]
2
3 array.
4
5   at
6   concat
7   copyWithin
8   entries
9   every
10  fill
11  filter
12  find
13  ...
14  ...
```

Apertando **Ctrl + Espaço** no VSCode, ele sugere automaticamente diversos métodos para os arrays.

# Funções (Rápida Interrupção!)

---



```
1 // Definição convencional
2 function soma(a, b) {
3   return a + b
4 }
5
6 // Arrow Function
7 const soma = (a, b) => a + b
8 // ou
9 const soma = (a, b) => {
10   return a + b
11 }
```

Não se assuste com as arrow functions!

Chamando a função **soma** definida:



```
1 const resultado = soma(2, 3)
```

# Funções (Rápida Interrupção!)

---

Exemplo de uso de arrow functions: funções genéricas com **high order functions!** É um conceito originado da programação funcional; são funções que recebem funções como parâmetro.

Isso facilita a personalização de funções, pois uma mesma estrutura pode ser reaproveitada para diferentes regras de negócio.



```
1 const array = [5, 9, 7, 1, -2, 8, 5]
2
3 function ordena(unordered_array, func) {
4     for (let i = 0; i < array.length; i++) {
5         for (let j = 0; j < array.length - 1; j++) {
6             if (!func(array[j], array[j + 1])) {
7                 let aux = array[j + 1]
8                 array[j + 1] = array[j]
9                 array[j] = aux
10            }
11        }
12    }
13
14 ordena(array, (primeiro, segundo) => primeiro < segundo)
15
16 console.log(array) // [-2, 1, 5, 5, 7, 8, 9]
17
18 ordena(array, (primeiro, segundo) => primeiro > segundo)
19
20 console.log(array) // [9, 8, 7, 5, 5, 1, -2]
```

# Arrays (Retomando...)

---



```
1 for (let i = 0; i < array.length; i++)  
2   console.log(array[i])
```

Não pense só no jeito C de programar. Agora você tem superpoderes!



```
1 array.forEach((value) => {  
2   console.log(value)  
3 })
```

# Arrays

---



```
1 const triple_array = array.map((value) =>
2   {
3     value * 3
4   }
5 )
6
7 console.log(array)
8 // [ 5, 9, 7, 1, -2, 8, 5 ]
9 console.log(triple_array)
10 // [ 15, 27, 21, 3, -6, 24, 15 ]
```



```
1 const impares =
2   array.filter((value) =>
3     value % 2 != 0
4   )
5
6 console.log(impares)
7 // [5, 9, 7, 1, 5]
```

# Objetos

---

Semelhantes a structs em C. Objetos JS possuem propriedades que são acessadas por seu nome (ou chave).



```
1 const nomePessoa = "Breno"  
2 const idadePessoa = 19
```



```
1 const pessoa = {  
2   nome: "Breno",  
3   idade: 19  
4 }
```

Por que deixar jogado dessa forma...

...se podemos agrupar e manter organizado?

# Objetos

---

E se eu quiser usar objetos como parâmetros de funções?



```
1 function imprimePessoa(pessoa) {  
2   console.log(pessoa.nome)  
3   console.log(pessoa.idade)  
4 }  
5  
6 imprimePessoa(pessoa)
```

Bom, agora sim.

Parece sem emoção. Vamos deixar mais divertido.

```
● ○ ■  
1 const personagem = {  
2   nome: "Thalion Sonbrassol",  
3   classe: "Arqueiro Sombrio",  
4   raça: "Elfo Noturno",  
5   nível: 27,  
6   experiência: 184500,  
7   pontosDeVida: 320,  
8   pontosDeMana: 150,  
9   energia: 92,  
10  força: 14,  
11  agilidade: 28,  
12  inteligência: 21,  
13  carisma: 17,  
14  constituição: 16,  
15  percepção: 22,  
16  sorte: 8,  
17  equipamentos: {  
18   armaPrincipal: "Arco Longo de Ébano",  
19   armaSecundária: "Adaga da Névoa",  
20   armadura: "Manto das Sombras",  
21   botas: "Botas Silenciosas",  
22   acessórios: ["Anel da Visão Noturna",  
23   "Colar da Camuflagem"],  
24 },  
25  inventário: {  
26   ouro: 523,  
27   prata: 78,  
28   poções: {  
29     oura: 5,  
30     mana: 3,  
31     invisibilidade: 1,  
32   },  
33   itensDiversos: ["Mapa Antigo", "Chave Rúnica", "Fena de Oritó"],  
34 },  
35  habilidades: [  
36   "Tiro Triplo",  
37   "Flecha Envenenada",  
38   "Passo Fantasma",  
39   "Visão Aguçada",  
40   "Camuflagem Natural",  
41 ],  
42  missõesItinerante: "Recuperar o Olho do Dragão",  
43  missõesConcluídas: 10,  
44  reputação: {  
45   guilda: "Neutro",  
46   reino: "Afiado",  
47   aldeões: "Respeitado",  
48 },  
49  conquistas: [  
50   "Cacador de Sombras",  
51   "Mestre das Armas",  
52   "Sombrio e Letal",  
53   "Arqueiro Sombrio",  
54   "Visão Noturna",  
55   "Silencioso",  
56   "Agilidade Aumentada",  
57   "Constituição Aumentada",  
58   "Percepção Aumentada",  
59   "Carisma Aumentado",  
60   "Inteligência Aumentada",  
61   "Força Aumentada",  
62   "Agilidade Aumentada",  
63   "Invisibilidade",  
64   "Visão Noturna",  
65   "Silencioso",  
66   "Arqueiro Sombrio",  
67   "Cacador de Sombras",  
68   "Mestre das Armas",  
69   "Sombrio e Letal",  
70   "Visão Noturna",  
71   "Silencioso",  
72   "Arqueiro Sombrio",  
73   "Cacador de Sombras",  
74   "Mestre das Armas",  
75   "Sombrio e Letal",  
76   "Visão Noturna",  
77   "Silencioso",  
78   "Arqueiro Sombrio",  
79   "Cacador de Sombras",  
80   "Mestre das Armas",  
81   "Sombrio e Letal",  
82   "Visão Noturna",  
83   "Silencioso",  
84   "Arqueiro Sombrio",  
85   "Cacador de Sombras",  
86   "Mestre das Armas",  
87   "Sombrio e Letal",  
88   "Visão Noturna",  
89   "Silencioso",  
90   "Arqueiro Sombrio",  
91   "Cacador de Sombras",  
92   "Mestre das Armas",  
93   "Sombrio e Letal",  
94   "Visão Noturna",  
95   "Silencioso",  
96   "Arqueiro Sombrio",  
97   "Cacador de Sombras",  
98   "Mestre das Armas",  
99   "Sombrio e Letal",  
100  "Visão Noturna",  
101  "Silencioso",  
102  "Arqueiro Sombrio",  
103  "Cacador de Sombras",  
104  "Mestre das Armas",  
105  "Sombrio e Letal",  
106  "Visão Noturna",  
107  "Silencioso",  
108  "Arqueiro Sombrio",  
109  "Cacador de Sombras",  
110  "Mestre das Armas",  
111  "Sombrio e Letal",  
112  "Visão Noturna",  
113  "Silencioso",  
114  "Arqueiro Sombrio",  
115  "Cacador de Sombras",  
116  "Mestre das Armas",  
117  "Sombrio e Letal",  
118  "Visão Noturna",  
119  "Silencioso",  
120  "Arqueiro Sombrio",  
121  "Cacador de Sombras",  
122  "Mestre das Armas",  
123  "Sombrio e Letal",  
124  "Visão Noturna",  
125  "Silencioso",  
126  "Arqueiro Sombrio",  
127  "Cacador de Sombras",  
128  "Mestre das Armas",  
129  "Sombrio e Letal",  
130  "Visão Noturna",  
131  "Silencioso",  
132  "Arqueiro Sombrio",  
133  "Cacador de Sombras",  
134  "Mestre das Armas",  
135  "Sombrio e Letal",  
136  "Visão Noturna",  
137  "Silencioso",  
138  "Arqueiro Sombrio",  
139  "Cacador de Sombras",  
140  "Mestre das Armas",  
141  "Sombrio e Letal",  
142  "Visão Noturna",  
143  "Silencioso",  
144  "Arqueiro Sombrio",  
145  "Cacador de Sombras",  
146  "Mestre das Armas",  
147  "Sombrio e Letal",  
148  "Visão Noturna",  
149  "Silencioso",  
150  "Arqueiro Sombrio",  
151  "Cacador de Sombras",  
152  "Mestre das Armas",  
153  "Sombrio e Letal",  
154  "Visão Noturna",  
155  "Silencioso",  
156  "Arqueiro Sombrio",  
157  "Cacador de Sombras",  
158  "Mestre das Armas",  
159  "Sombrio e Letal",  
160  "Visão Noturna",  
161  "Silencioso",  
162  "Arqueiro Sombrio",  
163  "Cacador de Sombras",  
164  "Mestre das Armas",  
165  "Sombrio e Letal",  
166  "Visão Noturna",  
167  "Silencioso",  
168  "Arqueiro Sombrio",  
169  "Cacador de Sombras",  
170  "Mestre das Armas",  
171  "Sombrio e Letal",  
172  "Visão Noturna",  
173  "Silencioso",  
174  "Arqueiro Sombrio",  
175  "Cacador de Sombras",  
176  "Mestre das Armas",  
177  "Sombrio e Letal",  
178  "Visão Noturna",  
179  "Silencioso",  
180  "Arqueiro Sombrio",  
181  "Cacador de Sombras",  
182  "Mestre das Armas",  
183  "Sombrio e Letal",  
184  "Visão Noturna",  
185  "Silencioso",  
186  "Arqueiro Sombrio",  
187  "Cacador de Sombras",  
188  "Mestre das Armas",  
189  "Sombrio e Letal",  
190  "Visão Noturna",  
191  "Silencioso",  
192  "Arqueiro Sombrio",  
193  "Cacador de Sombras",  
194  "Mestre das Armas",  
195  "Sombrio e Letal",  
196  "Visão Noturna",  
197  "Silencioso",  
198  "Arqueiro Sombrio",  
199  "Cacador de Sombras",  
200  "Mestre das Armas",  
201  "Sombrio e Letal",  
202  "Visão Noturna",  
203  "Silencioso",  
204  "Arqueiro Sombrio",  
205  "Cacador de Sombras",  
206  "Mestre das Armas",  
207  "Sombrio e Letal",  
208  "Visão Noturna",  
209  "Silencioso",  
210  "Arqueiro Sombrio",  
211  "Cacador de Sombras",  
212  "Mestre das Armas",  
213  "Sombrio e Letal",  
214  "Visão Noturna",  
215  "Silencioso",  
216  "Arqueiro Sombrio",  
217  "Cacador de Sombras",  
218  "Mestre das Armas",  
219  "Sombrio e Letal",  
220  "Visão Noturna",  
221  "Silencioso",  
222  "Arqueiro Sombrio",  
223  "Cacador de Sombras",  
224  "Mestre das Armas",  
225  "Sombrio e Letal",  
226  "Visão Noturna",  
227  "Silencioso",  
228  "Arqueiro Sombrio",  
229  "Cacador de Sombras",  
230  "Mestre das Armas",  
231  "Sombrio e Letal",  
232  "Visão Noturna",  
233  "Silencioso",  
234  "Arqueiro Sombrio",  
235  "Cacador de Sombras",  
236  "Mestre das Armas",  
237  "Sombrio e Letal",  
238  "Visão Noturna",  
239  "Silencioso",  
240  "Arqueiro Sombrio",  
241  "Cacador de Sombras",  
242  "Mestre das Armas",  
243  "Sombrio e Letal",  
244  "Visão Noturna",  
245  "Silencioso",  
246  "Arqueiro Sombrio",  
247  "Cacador de Sombras",  
248  "Mestre das Armas",  
249  "Sombrio e Letal",  
250  "Visão Noturna",  
251  "Silencioso",  
252  "Arqueiro Sombrio",  
253  "Cacador de Sombras",  
254  "Mestre das Armas",  
255  "Sombrio e Letal",  
256  "Visão Noturna",  
257  "Silencioso",  
258  "Arqueiro Sombrio",  
259  "Cacador de Sombras",  
260  "Mestre das Armas",  
261  "Sombrio e Letal",  
262  "Visão Noturna",  
263  "Silencioso",  
264  "Arqueiro Sombrio",  
265  "Cacador de Sombras",  
266  "Mestre das Armas",  
267  "Sombrio e Letal",  
268  "Visão Noturna",  
269  "Silencioso",  
270  "Arqueiro Sombrio",  
271  "Cacador de Sombras",  
272  "Mestre das Armas",  
273  "Sombrio e Letal",  
274  "Visão Noturna",  
275  "Silencioso",  
276  "Arqueiro Sombrio",  
277  "Cacador de Sombras",  
278  "Mestre das Armas",  
279  "Sombrio e Letal",  
280  "Visão Noturna",  
281  "Silencioso",  
282  "Arqueiro Sombrio",  
283  "Cacador de Sombras",  
284  "Mestre das Armas",  
285  "Sombrio e Letal",  
286  "Visão Noturna",  
287  "Silencioso",  
288  "Arqueiro Sombrio",  
289  "Cacador de Sombras",  
290  "Mestre das Armas",  
291  "Sombrio e Letal",  
292  "Visão Noturna",  
293  "Silencioso",  
294  "Arqueiro Sombrio",  
295  "Cacador de Sombras",  
296  "Mestre das Armas",  
297  "Sombrio e Letal",  
298  "Visão Noturna",  
299  "Silencioso",  
300  "Arqueiro Sombrio",  
301  "Cacador de Sombras",  
302  "Mestre das Armas",  
303  "Sombrio e Letal",  
304  "Visão Noturna",  
305  "Silencioso",  
306  "Arqueiro Sombrio",  
307  "Cacador de Sombras",  
308  "Mestre das Armas",  
309  "Sombrio e Letal",  
310  "Visão Noturna",  
311  "Silencioso",  
312  "Arqueiro Sombrio",  
313  "Cacador de Sombras",  
314  "Mestre das Armas",  
315  "Sombrio e Letal",  
316  "Visão Noturna",  
317  "Silencioso",  
318  "Arqueiro Sombrio",  
319  "Cacador de Sombras",  
320  "Mestre das Armas",  
321  "Sombrio e Letal",  
322  "Visão Noturna",  
323  "Silencioso",  
324  "Arqueiro Sombrio",  
325  "Cacador de Sombras",  
326  "Mestre das Armas",  
327  "Sombrio e Letal",  
328  "Visão Noturna",  
329  "Silencioso",  
330  "Arqueiro Sombrio",  
331  "Cacador de Sombras",  
332  "Mestre das Armas",  
333  "Sombrio e Letal",  
334  "Visão Noturna",  
335  "Silencioso",  
336  "Arqueiro Sombrio",  
337  "Cacador de Sombras",  
338  "Mestre das Armas",  
339  "Sombrio e Letal",  
340  "Visão Noturna",  
341  "Silencioso",  
342  "Arqueiro Sombrio",  
343  "Cacador de Sombras",  
344  "Mestre das Armas",  
345  "Sombrio e Letal",  
346  "Visão Noturna",  
347  "Silencioso",  
348  "Arqueiro Sombrio",  
349  "Cacador de Sombras",  
350  "Mestre das Armas",  
351  "Sombrio e Letal",  
352  "Visão Noturna",  
353  "Silencioso",  
354  "Arqueiro Sombrio",  
355  "Cacador de Sombras",  
356  "Mestre das Armas",  
357  "Sombrio e Letal",  
358  "Visão Noturna",  
359  "Silencioso",  
360  "Arqueiro Sombrio",  
361  "Cacador de Sombras",  
362  "Mestre das Armas",  
363  "Sombrio e Letal",  
364  "Visão Noturna",  
365  "Silencioso",  
366  "Arqueiro Sombrio",  
367  "Cacador de Sombras",  
368  "Mestre das Armas",  
369  "Sombrio e Letal",  
370  "Visão Noturna",  
371  "Silencioso",  
372  "Arqueiro Sombrio",  
373  "Cacador de Sombras",  
374  "Mestre das Armas",  
375  "Sombrio e Letal",  
376  "Visão Noturna",  
377  "Silencioso",  
378  "Arqueiro Sombrio",  
379  "Cacador de Sombras",  
380  "Mestre das Armas",  
381  "Sombrio e Letal",  
382  "Visão Noturna",  
383  "Silencioso",  
384  "Arqueiro Sombrio",  
385  "Cacador de Sombras",  
386  "Mestre das Armas",  
387  "Sombrio e Letal",  
388  "Visão Noturna",  
389  "Silencioso",  
390  "Arqueiro Sombrio",  
391  "Cacador de Sombras",  
392  "Mestre das Armas",  
393  "Sombrio e Letal",  
394  "Visão Noturna",  
395  "Silencioso",  
396  "Arqueiro Sombrio",  
397  "Cacador de Sombras",  
398  "Mestre das Armas",  
399  "Sombrio e Letal",  
400  "Visão Noturna",  
401  "Silencioso",  
402  "Arqueiro Sombrio",  
403  "Cacador de Sombras",  
404  "Mestre das Armas",  
405  "Sombrio e Letal",  
406  "Visão Noturna",  
407  "Silencioso",  
408  "Arqueiro Sombrio",  
409  "Cacador de Sombras",  
410  "Mestre das Armas",  
411  "Sombrio e Letal",  
412  "Visão Noturna",  
413  "Silencioso",  
414  "Arqueiro Sombrio",  
415  "Cacador de Sombras",  
416  "Mestre das Armas",  
417  "Sombrio e Letal",  
418  "Visão Noturna",  
419  "Silencioso",  
420  "Arqueiro Sombrio",  
421  "Cacador de Sombras",  
422  "Mestre das Armas",  
423  "Sombrio e Letal",  
424  "Visão Noturna",  
425  "Silencioso",  
426  "Arqueiro Sombrio",  
427  "Cacador de Sombras",  
428  "Mestre das Armas",  
429  "Sombrio e Letal",  
430  "Visão Noturna",  
431  "Silencioso",  
432  "Arqueiro Sombrio",  
433  "Cacador de Sombras",  
434  "Mestre das Armas",  
435  "Sombrio e Letal",  
436  "Visão Noturna",  
437  "Silencioso",  
438  "Arqueiro Sombrio",  
439  "Cacador de Sombras",  
440  "Mestre das Armas",  
441  "Sombrio e Letal",  
442  "Visão Noturna",  
443  "Silencioso",  
444  "Arqueiro Sombrio",  
445  "Cacador de Sombras",  
446  "Mestre das Armas",  
447  "Sombrio e Letal",  
448  "Visão Noturna",  
449  "Silencioso",  
450  "Arqueiro Sombrio",  
451  "Cacador de Sombras",  
452  "Mestre das Armas",  
453  "Sombrio e Letal",  
454  "Visão Noturna",  
455  "Silencioso",  
456  "Arqueiro Sombrio",  
457  "Cacador de Sombras",  
458  "Mestre das Armas",  
459  "Sombrio e Letal",  
460  "Visão Noturna",  
461  "Silencioso",  
462  "Arqueiro Sombrio",  
463  "Cacador de Sombras",  
464  "Mestre das Armas",  
465  "Sombrio e Letal",  
466  "Visão Noturna",  
467  "Silencioso",  
468  "Arqueiro Sombrio",  
469  "Cacador de Sombras",  
470  "Mestre das Armas",  
471  "Sombrio e Letal",  
472  "Visão Noturna",  
473  "Silencioso",  
474  "Arqueiro Sombrio",  
475  "Cacador de Sombras",  
476  "Mestre das Armas",  
477  "Sombrio e Letal",  
478  "Visão Noturna",  
479  "Silencioso",  
480  "Arqueiro Sombrio",  
481  "Cacador de Sombras",  
482  "Mestre das Armas",  
483  "Sombrio e Letal",  
484  "Visão Noturna",  
485  "Silencioso",  
486  "Arqueiro Sombrio",  
487  "Cacador de Sombras",  
488  "Mestre das Armas",  
489  "Sombrio e Letal",  
490  "Visão Noturna",  
491  "Silencioso",  
492  "Arqueiro Sombrio",  
493  "Cacador de Sombras",  
494  "Mestre das Armas",  
495  "Sombrio e Letal",  
496  "Visão Noturna",  
497  "Silencioso",  
498  "Arqueiro Sombrio",  
499  "Cacador de Sombras",  
500  "Mestre das Armas",  
501  "Sombrio e Letal",  
502  "Visão Noturna",  
503  "Silencioso",  
504  "Arqueiro Sombrio",  
505  "Cacador de Sombras",  
506  "Mestre das Armas",  
507  "Sombrio e Letal",  
508  "Visão Noturna",  
509  "Silencioso",  
510  "Arqueiro Sombrio",  
511  "Cacador de Sombras",  
512  "Mestre das Armas",  
513  "Sombrio e Letal",  
514  "Visão Noturna",  
515  "Silencioso",  
516  "Arqueiro Sombrio",  
517  "Cacador de Sombras",  
518  "Mestre das Armas",  
519  "Sombrio e Letal",  
520  "Visão Noturna",  
521  "Silencioso",  
522  "Arqueiro Sombrio",  
523  "Cacador de Sombras",  
524  "Mestre das Armas",  
525  "Sombrio e Letal",  
526  "Visão Noturna",  
527  "Silencioso",  
528  "Arqueiro Sombrio",  
529
```

# Desestruturação

---

Basicamente a ideia dessa funcionalidade é permitir acessar campos nomeados de um objeto.

```
● ● ●  
1 function resumoPersonagem({ nome, classe, nivel, raca, montaria }) {  
2   console.log(`🧙 ${nome}, ${classe} de nível ${nivel} (${raca})`)  
3   if (montaria) {  
4     console.log(`🦊 Montaria: ${montaria.tipo} chamado "${montaria.nome}"`)  
5   }  
6 }
```

# Desestruturação

---

O uso dessa funcionalidade nos permite, de forma enxuta, selecionar os atributos de um objeto.

É importante entender que esses campos **precisam existir no objeto**, senão teremos um erro de execução.

Mas... quem falou que ela só funciona pra objetos?

# Desestruturação

---

A desestruturação em arrays leva em conta a ordem dos elementos.



```
1 const [a, b] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 console.log(a) // 1
3 console.log(b) // 2
```



```
1 const [b, a] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 console.log(a) // 2
3 console.log(b) // 1
```

# Desestruturação

---

Já nos objetos, somente o nome do atributo importa, sendo a ordem irrelevante.



```
1 const { nome, idade } = pessoa
2 console.log(nome)
3 // "Breno"
4 console.log(idade)
5 // 19
```



```
1 const { idade, nome } = pessoa
2 console.log(nome)
3 // "Breno"
4 console.log(idade)
5 // 19
```

# Operadores Spread & Rest

---

Agora estamos lidando com estruturas de dados, então temos muitas informações para lidar. Não faz sentido trabalhar com esses dados um a um.

Agora tem a parte divertida. Vou apresentar dois operadores chamados **rest** e **spread**, que têm uma sintaxe bem... curiosa.

# Operadores Spread & Rest

**Spread: ...**

**Rest: ...**

Não é brincadeira.

Esses operadores apresentam a exata mesma sintaxe. Porém, seu uso vai ocorrer em casos distintos.

# Operadores Spread & Rest

---

**Spread** significa **espalhar**, e é exatamente isso que esse operador faz. Ele vai “abrir” uma array ou objeto, espalhando cópias de seus elementos. É amplamente utilizado para duplicar arrays e objetos.

# Operadores Spread & Rest

---



```
1 const array = [1, 2, 3, 4, 5, 6]
2 const novo_array = array
3 novo_array[0] = 20
4 console.log(array)
5 // [ 20, 2, 3, 4, 5, 6 ]
6 console.log(novo_array)
7 // [ 20, 2, 3, 4, 5, 6 ]
```



```
1 const array = [1, 2, 3, 4, 5, 6]
2 const novo_array = [...array]
3 novo_array[0] = 20
4 console.log(array)
5 // [ 1, 2, 3, 4, 5, 6 ]
6 console.log(novo_array)
7 // [ 20, 2, 3, 4, 5, 6 ]
```

# Operadores Spread & Rest

---

Já **rest** significa (que surpresa) **resto**! Então, basicamente ele vai armazenar o restante dos dados de um array ou objeto.

```
● ● ●  
1 const [a, b, ...resto] = [1, 2, 3, 4, 5]  
2  
3 console.log(a) // 1  
4 console.log(b) // 2  
5 console.log(resto) // [ 3, 4, 5]
```

A variável **resto** vai armazenar um **array** de dados, correspondente ao restante das informações, que não foram atribuídas.

# Operadores Spread & Rest

---

No caso dos objetos, o **rest** vai armazenar um objeto, com uma cópia das chaves e respectivas propriedades restantes.



```
1 const { nome, ...resto } = pessoa
2 console.log(nome)
3 // "Breno"
4 console.log(resto)
5 /*
6   sobrenome: "Pinna",
7   idade: 19,
8   altura: 1.8,
9 */
```

Alterações em **resto** não vão alterar o objeto original. Isso é importante por questões de consistência da informação ao longo do seu programa.

# Classes Prontas do JavaScript

---

O JS traz várias classes prontas para que possamos utilizar:

```
● ● ●  
1 // Constantes  
2 Math.E // 2.718281828459045  
3 Math.PI // 3.141592653589793  
4  
5 // Funções  
6 Math.floor(Math.PI) // 3  
7 Math.ceil(Math.PI) // 4  
8  
9 Math.max(1, 9, 4, 5, 2, -2, 6) // 9  
10 // ... entre outras
```

```
● ● ●  
1 // Cria um objeto de data com as  
2 // informações da data atual  
3 const date = new Date()  
4 // Retorna a data atual local  
5 date.toLocaleDateString() // '13/05/2025'  
6 // Retorna o horário local  
7 date.toLocaleTimeString() // '09:46:11'
```

# Classes Prontas do JavaScript

---

Temos uma classe que traz métodos para a **manipulação de JSON**.  
(JavaScript Object Notation)

```
● ● ●  
1 <  
2   "nome": "Breno",  
3   "sobrenome": "Pinna",  
4   "idade": 19,  
5   "altura": 1.8  
6 >
```

```
● ● ●  
1 <  
2   nome: 'Breno',  
3   sobrenome: 'Pinna',  
4   idade: 19,  
5   altura: 1.8  
6 >
```

Repare na  
semelhança entre  
o JSON e o objeto  
do JS

# Classes Prontas do JavaScript

---

O envio e recebimento de dados na web geralmente é feito por strings. Por isso há a conversão de JSON para a string e vice-versa.

```
> const pessoaString = JSON.stringify(pessoa)
  console.log(pessoaString)

  {"nome": "Breno", "sobrenome": "Pinna", "idade": 19, "altura": 1.8}

< undefined

> const pessoaObjeto = JSON.parse(pessoaString)
  console.log(pessoaObjeto)

  ▶ {nome: 'Breno', sobrenome: 'Pinna', idade: 19, altura: 1.8}

< undefined
```

# APIs e Web

---

API (Application Programming Interface) é um conceito muito presente no mundo da computação, principalmente na web.

Representa uma forma de nos comunicarmos com outras aplicações sem ter o contexto interno delas revelado.

# APIs e Web

---



# APIs e Web

---



# APIs e Web

---



# APIs e Web

---



# APIs e Web

---



# APIs e Web

---



# APIs e Web

---

O nosso exemplo será baseado em uma das APIs públicas mais famosas de todas...



# APIs e Web

---

## Conceitos importantes:

- **endpoints**: qual recurso da API eu quero acessar?
- **métodos HTTP**: o que quero fazer com esse recurso?
- **assincronismo**: preciso aguardar a API me responder para depois trabalhar com os dados dela

# Consumo de APIs

---

```
1  fetch("https://pokeapi.co/api/v2/pokemon/ditto")
2    .then(async (response) => {
3      const data = await response.json()
4      console.log(data)
5    })
6    .catch((error) => {
7      console.log(error)
8    })
9    .finally(() => {
10      console.log("Requisição finalizada.")
11    })

```

# Consumo de APIs

---

```
1 < abilities: [
2   {
3     ability: {
4       name: 'limber',
5       url: 'https://pokeapi.co/api/v2/ability/7/'
6     },
7     is_hidden: false,
8     slot: 1
9   },
10  {
11    ability: {
12      name: 'imposter',
13      url: 'https://pokeapi.co/api/v2/ability/150/'
14    },
15    is_hidden: true,
16    slot: 3
17  }
18 ],
19 base_experience: 101,
20 cries: {
21   latest:
22     'https://raw.githubusercontent.com/PokeAPI/cries/main/cries/pokemon/latest/132.ogg',
23   legacy:
24     'https://raw.githubusercontent.com/PokeAPI/cries/main/cries/pokemon/legacy/132.ogg'
25 },
26 forms: [
27   {
28     name: 'ditto',
29     url: 'https://pokeapi.co/api/v2/pokemon-form/132/'
30   }
31 ],
32 ...
33 ...
```

A partir do fim da requisição,  
teremos os dados disponíveis  
localmente!

# Document Object Model (DOM)

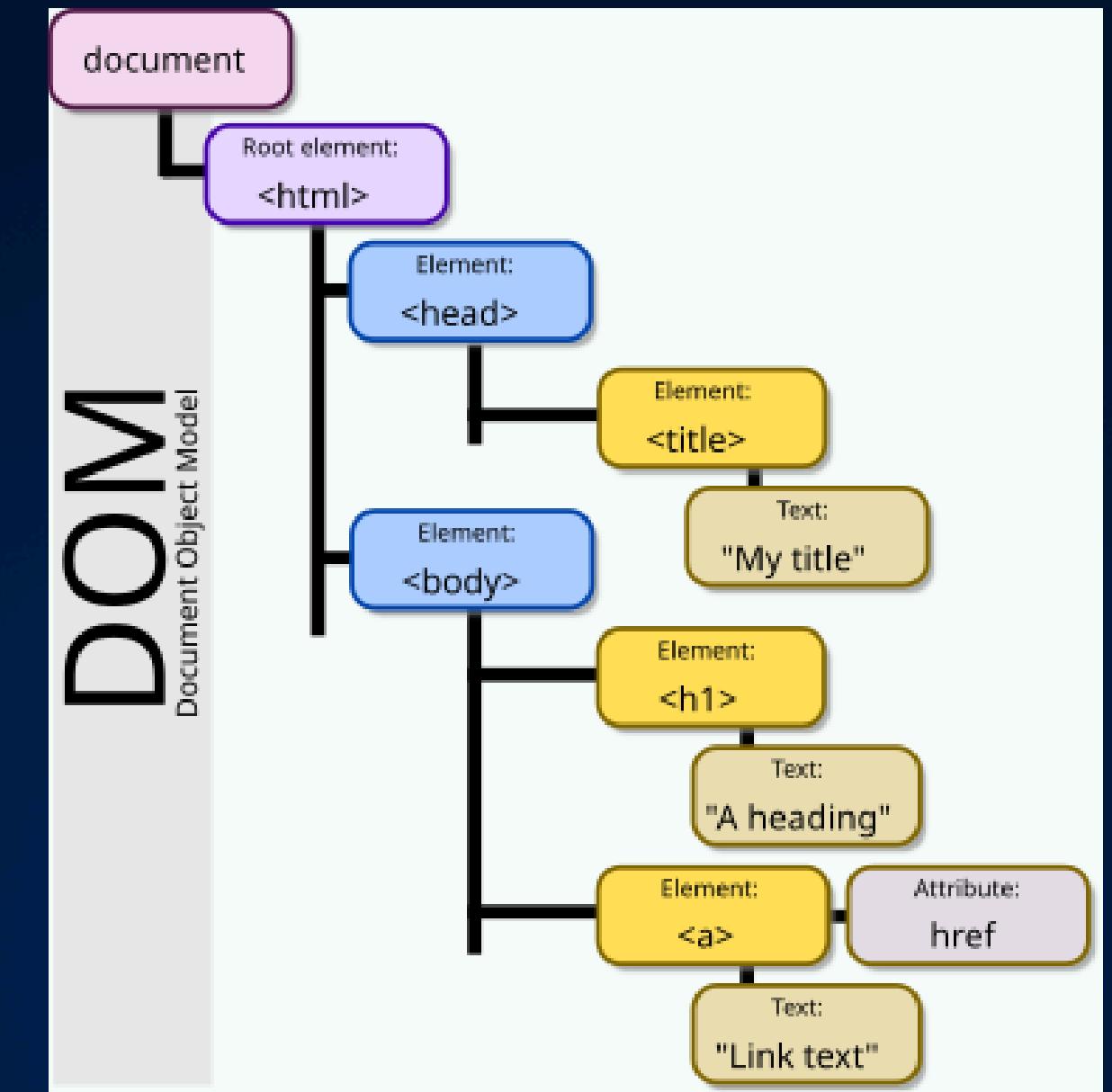
---

O mais importante conceito a se entender na manipulação de páginas web é o **Document Object Model (DOM)**. Basicamente, o JavaScript vai enxergar toda a página HTML como um objeto, e poderemos manipulá-lo através das funções que a linguagem oferece.

# Document Object Model (DOM)

---

O DOM pode ser visto como uma árvore de elementos HTML, acessível através do objeto **document**, que também possui métodos para manipulação do DOM.



[Endereço da Imagem](#)

# Manipulação do DOM

---

Considere o seguinte <body> no HTML:

```
1 <body>
2   <p class="generico">Um parágrafo interessante</p>
3   <p id="p-com-id">Esse é diferenciado!</p>
4   <p class="generico vermelho">Essa cor... está muito sem graça.</p>
5   <script src="dom.js"></script>
6 </body>
```



# Manipulação do DOM

---

Para começar a manipular o DOM, podemos começar pela leitura dos elementos:



```
1 const paragrafos = document.querySelectorAll("p")
```

# Manipulação do DOM

---



```
1 const vermelho = document.querySelector(".vermelho")
```



```
1 const pComId = document.getElementById("p-com-id")
```

# Manipulação do DOM

---

Podemos ainda **estilizar** os elementos através do JS!



```
1  paragrafos.forEach((paragrafo) => {
2    paragrafo.style.fontFamily = "Segoe UI"
3    paragrafo.style.fontWeight = "bold"
4  })
5
6  pComId.style.backgroundColor = "blue"
```

**Um parágrafo interessante**

**Esse é diferenciado!**

**Essa cor... está muito sem graça.**

# Manipulação do DOM

---

É possível também **criar** elementos pelo JS, e depois inserir no HTML!

```
● ● ●  
1 function criaBlocoDeTexto(titulo, corpo) {  
2   const div = document.createElement("div")  
3   const h1 = document.createElement("h1")  
4   const p = document.createElement("p")  
5   h1.innerText = titulo  
6   p.innerText = corpo  
7   div.appendChild(h1)  
8   div.appendChild(p)  
9   document.body.appendChild(div)  
10 }  
11  
12 criaBlocoDeTexto("Que", "Legal!")
```

Que

Legal!

# Eventos

---

De forma resumida, eventos serão todas as ações que podem ocorrer numa página web: cliques na tela, foco em inputs, carregamento de conteúdo, passar o mouse por cima, entre muitos outros.

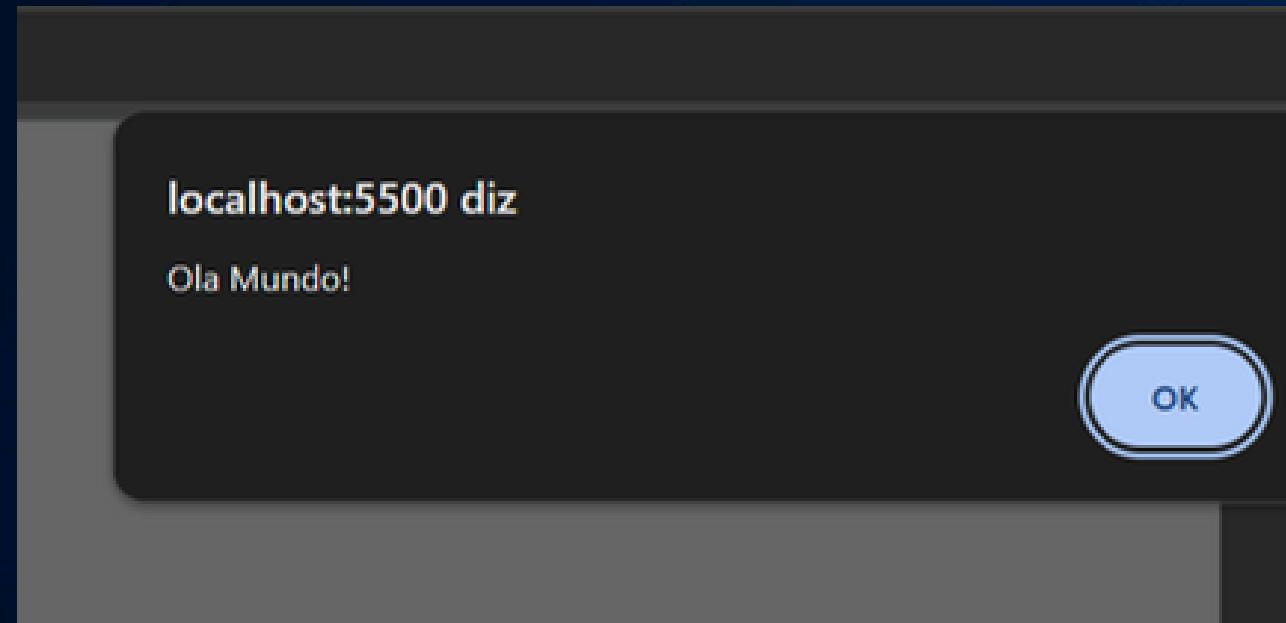
Para lidar com esses eventos, temos os **event listeners** (tradução livre: “ouvintes de eventos”)

# Eventos

---



```
1 document.addEventListener("click", () => {  
2   alert("Olá Mundo!")  
3 })
```



# Eventos

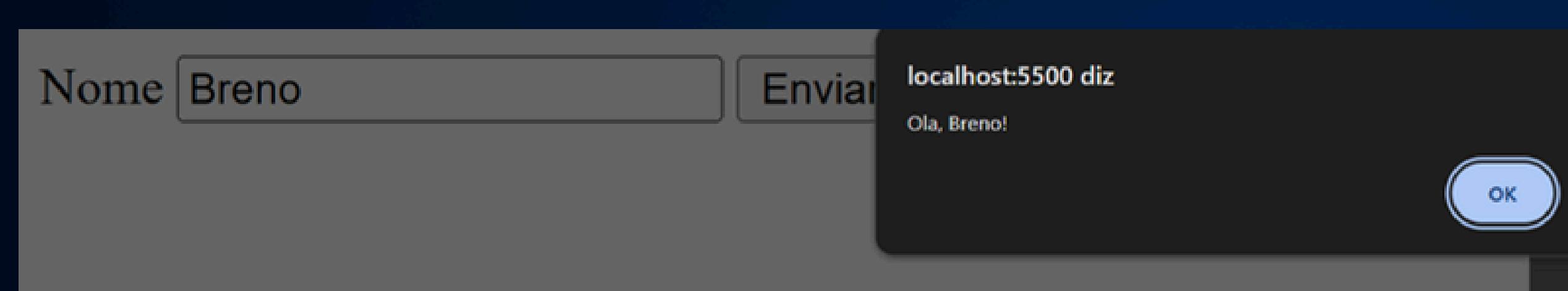
---



```
1 <form>
2   <label for="name">Nome</label>
3   <input type="text" name="name" id="name" />
4   <button type="submit">Enviar</button>
5 </form>
```



```
1 const btn = document.querySelector("button")
2 const input = document.getElementById("name")
3 btn.addEventListener("click", () => {
4   alert(`Olá, ${input.value}!`)
5 })
```



# Alguma pergunta?