

```
1  /*
2    01
3
4    - Crie um objeto "cat" que possui as seguintes propriedades:
5      - name, que recebe o nome do gato;
6      - age, que recebe a idade;
7      - color, que recebe a cor;
8      - bestFriends, que recebe um array com os melhores amigos do gato;
9      - sound, que é um método que retorna o miado do gato.
10   - Exiba o objeto no console.
11  */
12  const cat = {
13    name: 'Leopoldo',
14    age: 7,
15    color: 'white',
16    bestFriends: ['Juuh', 'Ma'],
17    sound: function () {
18      return 'miauuuu'
19    }
20  }
21
22  console.log(cat)
23  /*
24    02
25
26    - Exiba a mensagem abaixo no console, substituindo os "X" pelas informações
27      corretas.
28
29    Até aqui, o objeto "cat" possui as seguintes propriedades e valores: "name", que
30    recebeu "X", "age", que recebeu "X", "color", que recebeu "X", "bestFriends", que
31    recebeu um array com os itens "X" e "X", e "sound", que recebeu uma função que
32    retorna "X".
33  */
34  console.log(`Até aqui, o objeto "cat" possui as seguintes propriedades e valores:
35    "name", que recebeu ${cat.name}, "age", que recebeu ${cat.age}, "color", que recebeu
36    ${cat.color}, "bestFriends", que recebeu um array com os itens ${cat.bestFriends[0]}
37    e ${cat.bestFriends[1]}, e "sound", que recebeu uma função que retorna
38    ${cat.sound()}.`)
39  /*
40    03
41
42    - Adicione 2 anos à idade do gato e exiba a idade atualizada no console.
43  */
44  cat.age += 2
45  console.log(cat.age)
46  /*
47    04
48
49    - Crie uma função que adiciona um amigo(a) no array de amigos do gato;
50    - Exiba o array de amigos no console para verificar se o novo amigo(a) foi
51      adicionado.
52  */
53  const addFriend = friend => cat.bestFriends.push(friend)
54
55  addFriend('Cilene')
56
57  console.log(cat.bestFriends)
58
59  /*
60    05
```

```
54
55 - Crie uma função que adiciona mais uma cor à cor do gato;
56 - Utilize a sintaxe de colchetes para fazer isso;
57 - Exiba a nova cor do gato no console, também utilizando a sintaxe de
58   colchetes.
59 */
60 const addColor = object => {
61   object['color'] += ` e cinza`
62 }
63
64 addColor(cat)
65
66 const colorProperty = 'color'
67
68 console.log(cat[colorProperty])
69 /*
70   06
71
72 - Crie uma função que retorna um boolean indicando se um valor recebido por
73   parâmetro é um objeto;
74 - Utilize a função para exibir no console se "cat" é um objeto.
75 */
76 const isObject = value => typeof value === 'object'
77
78 console.log(isObject(cat))
79 /*
80   07
81
82 - Crie um objeto "dog" que possui as mesmas propriedades do objeto "cat";
83 - Crie uma função que retorna a mensagem abaixo, com as informações corretas;
84 - Exiba a mensagem no console.
85
86   "A soma das idades de NOME_DO_GATO e NOME_DO_CACHORRO é RESULTADO_DA_SOMA."
87 */
88 const dog = {
89   name: 'Theo',
90   age: 7,
91   color: 'preto',
92   bestFriends: ['Lilica', 'Luna'],
93   sound: function () {
94     return 'auauau'
95   }
96 }
97
98 console.log(`A soma das idades de ${cat.name} e ${dog.name} é ${cat.age + dog.age}.`)
99 /*
100   08
101
102 - Se você descomentar os 2 console.log abaixo, verá que a função "isAnSUV"
103   está funcionando corretamente. Mas ela pode ser melhorada;
104 - Como você refatoraria esta função?
105 */
106 const cars = ['Honda HR-V', 'Jeep Renegade', 'Ford EcoSport', 'Hyundai iX35']
107
108 const isAnSUV = car => {
109   if (cars.includes(car)) {
110     return true
111   } else {
112     return false
113   }
114 }
```

```
114 }
115
116 console.log(isAnSUV('Honda Civic'))
117 console.log(isAnSUV('Ford EcoSport'))
118
119 /*
120 09
121
122 - Crie uma função que recebe um parâmetro 'type';
123 - Dentro da função, crie um objeto com as seguintes propriedades:
124   - null, que recebe a mensagem 'Seta, explicitamente, uma variável sem valor.'
125   - undefined, que recebe a mensagem 'Representa um valor não-setado.'
126   - object, que recebe a mensagem 'Arrays, Datas, Objetos literais, Funções, etc.'
127 - Se o valor que o parâmetro type receber corresponder à alguma das 3
128 propriedades, retorne a mensagem que a propriedade armazena;
129 - Teste a função, exibindo no console a mensagem de cada propriedade.
130 */
131
132 const getTypeDefinition = type => {
133   const obj = {
134     null: 'Seta, explicitamente, uma variável sem valor.',
135     undefined: 'Representa um valor não-setado.',
136     object: 'Arrays, Datas, Objetos literais, Funções, etc.'
137   }
138
139   return obj[type]
140 }
141
142 console.log(getTypeDefinition('null'))
143 console.log(getTypeDefinition('undefined'))
144 console.log(getTypeDefinition('object'))
145
```