Coleções de dados

Praticamente todos os exemplos vistos até então tratavam de problemas cuja solução passava pela manipulação de alguns poucos dados; ou seja, exigia-se a definição de funções com **pequena quantidade** de parâmetros.

(ex.) Função que calcula a área de uma elipse exige 2 parâmetros apenas: $f(x, y) = xy\pi$

(ex.) Função que calcula a média aritmética entre as 4 notas de um aluno exige naturalmente 4 parâmetros: $g(x, y, z, w) = \frac{x+y+z+w}{4}$

Mas...
(ex.) ...e se quiséssemos medir o desempenho médio final de uma turma de 50 alunos? Deveríamos criar uma função passando a nota final de cada um para que calculássemos a média entre todos; isso daria 50 parâmetros: $h(x, y, z, ..., w, k) = \frac{x+y+z+...+w+k}{50}$

Isso **não seria muito prático**, considerando, inclusive, que teríamos que usar **vários símbolos/nomes/letras diferentes** para representar as notas.

Sabendo que a maioria absoluta dos problemas que buscam soluções computacionais precisam lidar com **grandes coleções de valores**, precisamos buscar uma alternativa de **representação de entrada de dados mais compacta** para esses casos.

A estrutura LISTA

Em Javascript, podemos utilizar o conceito de **listas de valores** como forma de representação para coleções de dados. A notação consiste do uso de **COLCHETES de abertura e fechamento** com **valores compreendidos** entre estes e separados por vírgula.

REPRESENTAÇÃO

```
lista = [5,8,10,3,11]
```

ACESSO (aos valores)

`lista[0]` , `lista[1]` , `lista[4]` , ...

OPERAÇÕES

`lista[3] * lista[4]` $\rightarrow 33$

TAMANHO (da lista)

`lista.length` $\rightarrow 5$

DEFINIÇÃO DE FUNÇÃO

```
const soma = (l) => l[0]+l[1]+l[2]+l[3]+l[4]
```

A estrutura REGISTRO

Outro recurso útil para melhor organizar a solução de problemas com quantidade maior e **mais diversa** de dados é o uso de **registros**. Um registro permite que representemos de forma agrupada diferentes aspectos de uma mesma entidade através de **atributos**.

Um software de gestão de uma instituição de ensino, por exemplo, poderia representar cada um dos discentes utilizando-se de registro.

REPRESENTAÇÃO

```
const aluno = { matricula: 20220507, nome: 'Fulano da Silva', nascimento: 2001, curso: 'Computação', mgp: 7.60, formando: true }
```

ACESSO (aos valores)

`aluno.matricula` $\rightarrow 20220507$,
`aluno.nome` $\rightarrow$ Fulano da Silva,

...,
`aluno.formando` $\rightarrow true$