

### Exercícios sobre POO, Exceções e DOM

1) Crie uma classe que represente um dado Dado, cujo número de lados é fornecido em sua criação. Caso o número de lados fornecido seja inferior a um, uma exceção do tipo DadoError deve ser lançada, com uma mensagem que informe a restrição.

Adicione um método lançar() à classe Dado que gera e retorna um número aleatório entre 1 e o número de lados do dado (dica: use [Math.random](#)).

Então, crie uma lista com um número aleatório de dados, contendo um número aleatório de lados. Lance todos os dados e indique qual o índice da lista teve o maior número gerado e quantos lados o respectivo dado possui.

2) Crie uma classe DadoSeisLados, que estenda a classe Dado, criada no exercício anterior, e cujo construtor não receba argumentos.

Então, crie uma classe Jogador, que contenha um nome e um dado de seis lados.

Por fim, crie uma página HTML que permita dois jogadores jogarem seus dados de seis lados e que indique qual o vencedor. Para isso, a página um botão Novo Jogo e um botão Jogar. Ao clicar em Novo Jogo, os nomes de dois jogadores devem ser perguntados – usando [window.prompt](#) – e os nomes devem ser exibidos como títulos de uma tabela que contém duas colunas e duas linhas (sendo a primeira linha de título). Quando o botão Jogar for clicado, os dados de seis lados de cada jogador devem ser jogados e figuras correspondentes aos resultados das jogadas devem ser exibidas na segunda linha da tabela. Em um H1 abaixo da tabela, deve-se exibir uma mensagem que indique qual jogador foi o vencedor, ou se houve empate.

Como figuras dos dados, utilize as imagens constantes nos links abaixo:

Lado 1	<a href="https://bit.ly/dado-1">https://bit.ly/dado-1</a>
Lado 2	<a href="https://bit.ly/dado-2">https://bit.ly/dado-2</a>
Lado 3	<a href="https://bit.ly/dado-3">https://bit.ly/dado-3</a>
Lado 4	<a href="https://bit.ly/dado-4">https://bit.ly/dado-4</a>
Lado 5	<a href="https://bit.ly/dado-5">https://bit.ly/dado-5</a>
Lado 6	<a href="https://bit.ly/dado-6">https://bit.ly/dado-6</a>

3) Crie uma página HTML com um formulário para cadastro de clientes, em que um cliente pode ser uma pessoa física ou uma pessoa jurídica. Se for uma pessoa física, o formulário deve solicitar Nome, CPF, Telefone e E-mail. Do contrário, solicitar Nome Fantasia, Razão Social, CPNJ, Telefone e E-mail. O formulário deve exibir apenas os campos correspondentes ao tipo de cliente escolhido.

Junto ao formulário, construa classes para representar um cliente pessoa física e um cliente pessoa jurídica, que devem estender uma classe comum, Cliente. Ao disparar a opção para cadastrar um cliente, os dados do formulário devem ser passados para um objeto da classe correspondente, que

deve ser fornecido para uma função `exibirCliente`, que apresente os dados do objeto em uma divisão (`div`), abaixo do formulário.

4) Com base no exercício anterior, adicione à classe `Cliente` um método para validação de seus dados. A classe `Cliente` deve validar apenas seus próprios atributos. O método de validação deve retornar uma lista contendo mensagens correspondentes a cada problema encontrado, ou uma lista vazia se não houver problemas. As classes filhas devem sobrescrever esse método para realizar suas próprias validações, mas mantendo o comportamento da classe pai. As validações devem considerar que:

- \* Nome, Nome Fantasia e Razão Social devem ter entre 2 e 100 caracteres.

- \* CPF deve ter 11 caracteres numéricos.

- \* CNPJ deve ter 14 caracteres numéricos.

- \* Telefone deve ter 9 ou 11 caracteres numéricos.

- \* Email deve ter de 6 a 100 caracteres e ter arroba, não sendo esse o primeiro ou o último caractere.

Caso algum dado do cliente esteja incorreto, segundo as validações acima, as respectivas mensagens devem ser exibidas como uma lista não ordenada (`ul`), em que cada item (`li`) corresponda a uma mensagem. Exiba essa lista dentro da divisão (`div`) criada no exercício anterior. Crie uma função `exibirMensagens` (que recebe a lista de mensagens a ser exibida) para esse fim.

5) Crie uma página HTML que apresente uma tabela, com linhas criadas dinamicamente, que corresponda à abaixo:

Nome/Nome Fantasia	CPF/CNPJ	Razão Social	Telefone	E-mail
Acme	44.265.896/0001-06	Acme & Cia Ltda.	(22) 3344-5566	contato@acme.com
Ana da Silva	966.293.550-91		(21) 98877-6655	ana@silva.com
Beatriz Souza	268.218.110-40		(22) 99988-7744	bia@souza.com
Constelação	78.780.362/0001-30	Constelação S/A	(21) 4567-8901	sac@constelacao.com

Os dados da tabela devem ser obtidos de uma constante com uma lista contendo os respectivos objetos, instanciados com as classes criadas anteriormente.