

# 3º Curso Python para PLN - Edição 50 anos do ICMC: Avaliação

Olá pessoal! Para que vocês possam receber o certificado de participação e aprovação no 3º Curso Python para PLN - Edição 50 anos do ICMC, vocês devem responder essas 10 questões sobre todo o conteúdo do nosso curso.

Você precisa acertar pelo menos 7 questões para ser aprovado!

O formulário ficará aberto até domingo, dia 17 de outubro de 2021 às 23h59 para você responder! Bom divertimento! =)



sousagomide@gmail.com (não compartilhado) [Alternar conta](#)



**\*Obrigatório**

Nome: \*

Sua resposta

Email (de preferência, o que usou para se cadastrar no curso): \*

Sua resposta

Um exemplo prático de aplicação em PLN é um sistema de tradução automática. \*

1 ponto



Verdadeiro



Falso



Marque a opção que represente o resultado final da execução desse código: \*

1 ponto

```
>>> v = "1"  
>>> q = 1  
>>> v + q
```

- ☐ "1"1
- ☐ "11"
- ☐ O sistema retorna um erro
- ☐ 11

Qual das opções representa uma tupla em Python? \*

1 ponto

- ☐ [0, 1, 2, 3, 4]
- ☐ (0, 1, 2, 3, 4)
- ☐ "0, 1, 2, 3, 4"
- ☐ {0, 1, 2, 3, 4}

Dadas as linhas de código abaixo, qual opção representa o resultado da execução do código? \*

1 ponto

```
>>> estrutura = [10, 20, 30, 40, 50]  
>>> print(estrutura[:3])
```

- ☐ [10, 20, 30, 40]
- ☐ [30]
- ☐ [30, 40, 50]
- ☐ [10, 20, 30]



A linha de código abaixo tem como objetivo: \*

1 ponto

```
>>> s = 'instituto de ciências matemáticas e de computação'  
>>> len(s.split())
```

- ☐ Saber a quantidade de palavras da string s. O resultado é 49.
- ☐ Saber a quantidade de palavras da string s. O resultado é 7.
- ☐ Saber o tamanho da string s em caracteres e espaços. O resultado é 49.
- ☐ Saber o tamanho da string s em caracteres e espaços. O resultado é 7.

O trecho de código abaixo retorna uma lista de tokens, que são: \*

1 ponto

```
>>> texto = "O jogador, que está com a camisa 10, marcou o gol da vitória!"  
>>> from nltk.tokenize import RegexpTokenizer  
>>> tokenizer = RegexpTokenizer(r'[A-z]\w*')  
>>> tokens = tokenizer.tokenize(texto)  
>>> tokens
```

- ☐ Números
- ☐ Palavras
- ☐ Pontuações
- ☐ Todos os tipos de tokens

No NLTK, a contagem/frequência de tokens é feita pela classe FreqDist. \*

1 ponto

- ☐ Verdadeiro
- ☐ Falso



Usando o spaCy, caso se queira retornar todas as pontuações do texto/dataset, qual das linhas de código abaixo seria utilizada? \*

1 ponto

- ☐ `tokens_pontuacoes = [token.orth_ for token in doc if token.is_alpha]`
- ☐ `tokens_pontuacoes = [token.orth_ for token in doc if token.is_digit]`
- ☐ `tokens_pontuacoes = [token.orth_ for token in doc if token.is_punct]`
- ☐ `tokens_pontuacoes = [token.orth_ for token in doc]`

No spaCy, para retornar todas as classes gramaticais de um certo texto ou dataset, deve-se usar o atributo `.pos_` do token. \*

1 ponto

- ☐ Verdadeiro
- ☐ Falso

O reconhecimento de entidades nomeadas é uma técnica que consiste na identificação de nomes de entidades-chave, presentes na forma livre de dados textuais. Como podemos reconhecer as entidades nomeadas de um texto usando o spaCy? \*

1 ponto

- ☐ Não podemos reconhecer entidades nomeadas com o spaCy
- ☐ Todas as afirmações estão erradas
- ☐ Precisamos usar o atributo `'.ents'` presente no texto analisado pelo modelo do spaCy
- ☐ Somente se extraírmos os bigramas/trigramas/n-gramas e verificarmos quais tokens iniciam com maiúsculas.

Página 1 de 1

Enviar

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Universidade de São Paulo. [Denunciar abuso](#)

Google Formulários

