

**Estudante: Mateus Catureba Santana**  
**Email: [caturebadev@gmail.com](mailto:caturebadev@gmail.com)**  
**RA: 12722133037**  
**Numero para contato: [\(71\) 99725-5502](tel:(71)99725-5502)**  
**Github: [catureba](#)**

**1. Escreva uma expressão regular que:**

- a. Reconheça as palavras-chave da linguagem Python: if, else, for, while, def.

**R)** `\b(if|else|for|while|def)\b`

- b. Reconheça identificadores de variáveis válidos. Um identificador válido começa com uma letra (maiúscula ou minúscula) ou um sublinhado, seguido por letras, dígitos ou sublinhados.

**R)** `^[a-zA-Z_][a-zA-Z0-9_]*$`

- c. Identifique números inteiros e de ponto flutuante válidos. Um número válido pode ser positivo ou negativo e pode conter parte decimal.

**R)** `^-?\d+(\.\d+)?$`

- d. Reconheça datas no formato "dd/mm/aaaa". As datas devem estar dentro de um intervalo específico (por exemplo, entre 01/01/1900 e 31/12/2099).

**R)** `^(0[1-9]|[12][0-9]|3[01])/((0[1-9]|1[0-2])/((19|20)\d\d))$`

- e. Corresponda a endereços de e-mail válidos.

**R)** `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`

- f. Reconheça URLs válidas. A URL deve incluir o protocolo (por exemplo, "http" ou "https"), o domínio e pode incluir caminhos e parâmetros opcionais.

**R)** `^https?:\V[a-zA-Z0-9.-]+(?:\V[^\s]*)?$`

**2. Faça um código em Python usando regEx para extrair todos e-mails encontrados em uma string**

**R)**

```
import re
```

```
def extract_emails(text):
```

```
    email_regex = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
```

```
return re.findall(email_regex, text)
```

```
text = "Meu nome é Mateus Catureba e meus emails são é caturebadev@gmail.com e  
contato@mateuscatureba.com.br"  
emails = extract_emails(text)  
print(emails)
```

OBS: este código da questão 2 já aceitaria subdomínios conforme o senhor pediu na questão 3

3. Altere a regex `^\w+@\w+\.\w+$` para que aceite e-mails com subdomínio(s)

**R)** `^\w+@(\w+\.)+\w+$`