



THE DEVELOPER'S CONFERENCE

EXPRESSÕES REGULARES PARA NOVATOS, USANDO PYTHON



AMOM MENDES



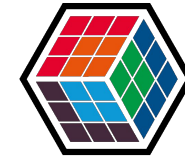
AGENDA



THE
DEVELOPER'S
CONFERENCE

- O QUE SÃO EXPRESSÕES REGULARES
- PARA QUE SERVEM?
- METACARACTERES
- NOTEBOOK

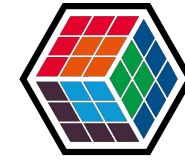
AGENDA



THE
DEVELOPER'S
CONFERENCE

- O QUE SÃO EXPRESSÕES REGULARES
- PARA QUE SERVEM?
- METACARACTERES
- NOTEBOOK

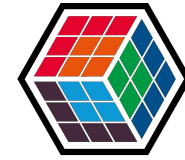
AGENDA



THE
DEVELOPER'S
CONFERENCE

- O QUE SÃO EXPRESSÕES REGULARES
- PARA QUE SERVEM?
- METACARACTERES
- NOTEBOOK

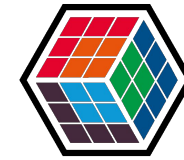
AGENDA



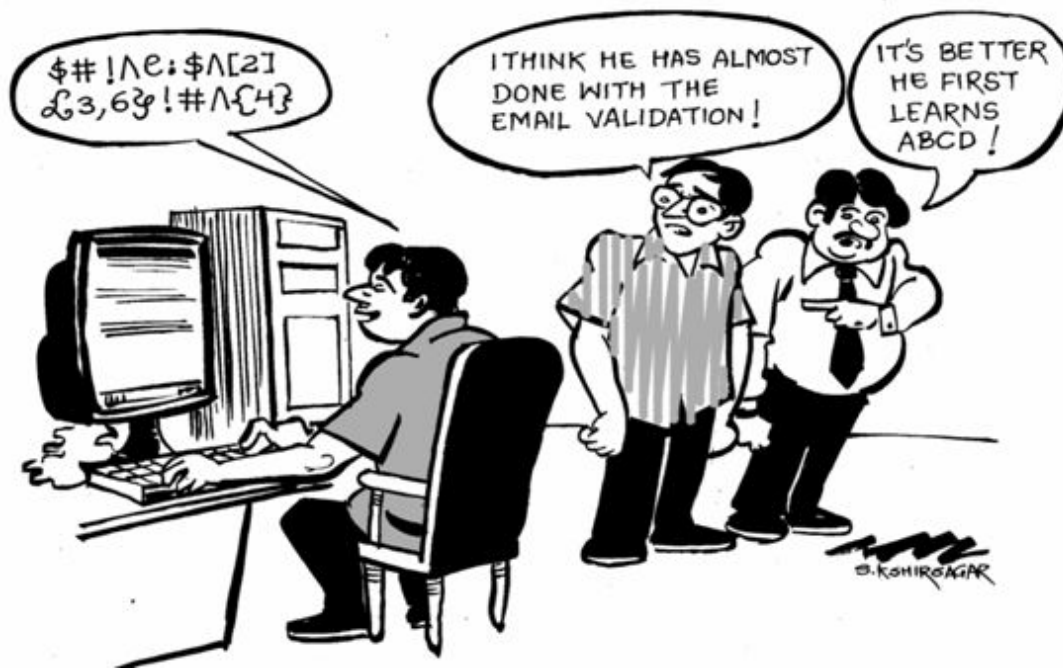
THE
DEVELOPER'S
CONFERENCE

- O QUE SÃO EXPRESSÕES REGULARES
- PARA QUE SERVEM?
- METACARACTERES
- NOTEBOOK (TALVEZ!)

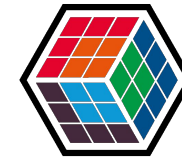
MOTIVAÇÃO



THE
DEVELOPER'S
CONFERENCE



MOTIVAÇÃO



THE
DEVELOPER'S
CONFERENCE





THE
DEVELOPER'S
CONFERENCE

O QUE SÃO EXPRESSÕES REGULARES?

O QUE SÃO EXPRESSÕES REGULARES?



THE
DEVELOPER'S
CONFERENCE

“Método formal de especificar um padrão de texto”



import re

O QUE SÃO EXPRESSÕES REGULARES?



THE
DEVELOPER'S
CONFERENCE

“Hoje tem trilha de Python no TDC”

CARACTERES LITERAIS

O QUE SÃO EXPRESSÕES REGULARES?



THE
DEVELOPER'S
CONFERENCE

“Hoje tem trilha de Python no TDC”

```
re.search  
(r'(?i)(python).*(tdc))
```

CARACTERES LITERAIS

REGRA(SÍMB. + CARAC. FN. ESPECIAIS)

O QUE SÃO EXPRESSÕES REGULARES?



THE
DEVELOPER'S
CONFERENCE

“Hoje tem trilha de Python no TDC”

```
re.search  
(r'(?i)(python).*(tdc))
```

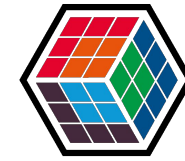
'Python no tdc'

CARACTERES LITERAIS

REGRA(SÍMB. + CARAC. FN.
ESPECIAIS)

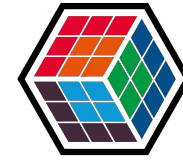
MATCH (CASOU)

PARA QUE SERVEM?



THE
DEVELOPER'S
CONFERENCE

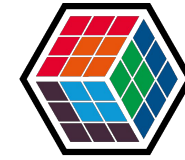
- MUITA / QUALQUER COISA RELACIONADA A BUSCA DE PADRÕES DE CARACTERES
 - ROUTER DE PÁGINAS (EX. DJANGO)
 - DATAS / HORÁRIOS
 - NOMES
 - E-MAILS
 - WEB SCRAPPING
 - CONSULTA BANCOS DE DADOS
 - SUBSTITUIÇÃO DE CARACTERES
 - ETC, ETC



THE
DEVELOPER'S
CONFERENCE

POR ONDE
COMEÇAR?

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

- PRINCIPAIS

. ? * + ^ \$ | [] { } () \

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

- SIMPLIFICANDO



REPRESENTANTES

. [] [^]



QUANTIFICADORES

? * + { }



ÂNCORAS

^ \$ \b

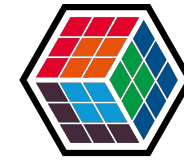


OUTROS

\ | (.) \ 1

METACARACTERES

- REPRESENTANTES

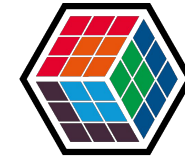


THE
DEVELOPER'S
CONFERENCE



(MÉTODO **FINDALL**: RETORNA UM LISTA
COM TEXTOS CASADOS)

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

- REPRESENTANTES

```
re.findall(r'n.o',  
          'não, nao, neo, nso,  
          nzo,nap') #String
```

#Expressão

```
['não', 'nao', 'neo', 'nso', 'nzo']
```



METACARACTERES

- REPRESENTANTES

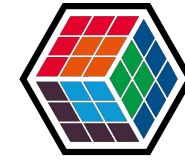


THE
DEVELOPER'S
CONFERENCE



[...]

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

- REPRESENTANTES

```
re.findall (r'[naão] ',  
           'não, nao, neo')
```

```
['n', 'ã', 'o', 'n',  
'a', 'o', 'n', 'o']
```

METACARACTERES

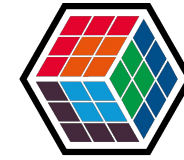


THE
DEVELOPER'S
CONFERENCE

- REPRESENTANTES

```
m = re.search(r'[0-9][0-9]:[0-9][0-9]',  
              'Cheguei as 11:00 horas no TDC hoje')  
m.group()
```

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

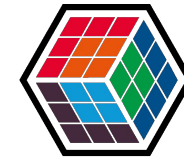
- REPRESENTANTES

```
m = re.search(r' [0-9] [0-9] : [0-9] [0-9] ',  
'Cheguei as 11:00 horas no TDC hoje')
```

```
m.group()
```

(MÉTODO **SEARCH**: TESTA SE CASOU E
RETORNA UM OBJETO COM INFORMAÇÕES
DO MATCH)

METACARACTERES



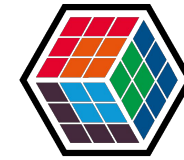
THE
DEVELOPER'S
CONFERENCE

- REPRESENTANTES

```
m = re.search(r' [0-9] [0-9] : [0-9] [0-9] ',  
              'Cheguei as 11:00 horas no TDC hoje')  
m.group()
```

11:00

METACARACTERES



THE
DEVELOPER'S
CONFERENCE

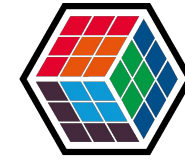
- REPRESENTANTES

```
m = re.findall(r'[^A-z ]', 'Cheguei as  
11:00 horas no TDC hoje')
```

```
['1', '1', ':', '0', '0']
```


METACARACTERES

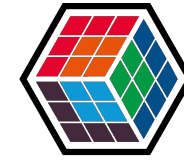
- QUANTIFICADORES



THE
DEVELOPER'S
CONFERENCE



METACARACTERES

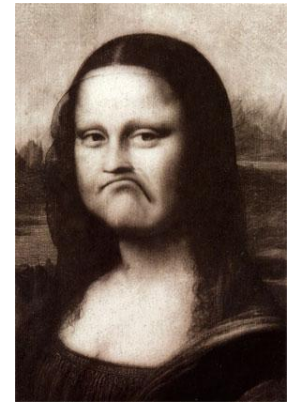


THE
DEVELOPER'S
CONFERENCE

- QUANTIFICADORES

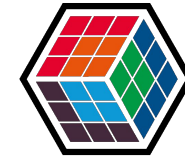
```
re.findall (r'fala[r!]?',  
'falar, fala!, fala')
```

```
['falar', 'fala!', 'fala']
```

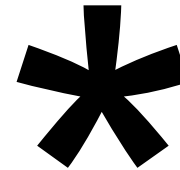


METACARACTERES

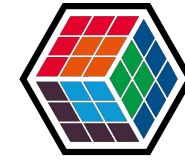
- QUANTIFICADORES



THE
DEVELOPER'S
CONFERENCE



METACARACTERES



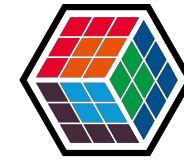
THE
DEVELOPER'S
CONFERENCE

- QUANTIFICADORES

```
re.findall (r 'Ah*!' ,  
'Ah! , Ahhh! , Ahhhh! , Ahhhhhhhhhh! ' )
```

```
[ 'Ah! ' , 'Ahhh! ' , 'Ahhhh! ' ,  
'Ahhhhhhhhhhh! ' ]
```

METACARACTERES



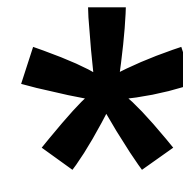
THE
DEVELOPER'S
CONFERENCE

- QUANTIFICADORES

```
re.findall (r'Ah.*!', 'Ah!,  
Ahhh!, Ahhhh!,A(h * 1000)!')
```



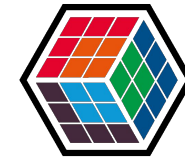
```
'Ah!, Ahhh!,  
Ahhhh!,A(h * 1000)!'
```



METACARACTERES

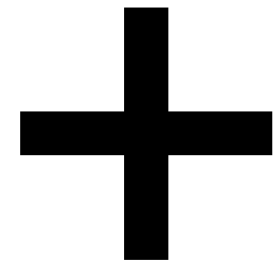
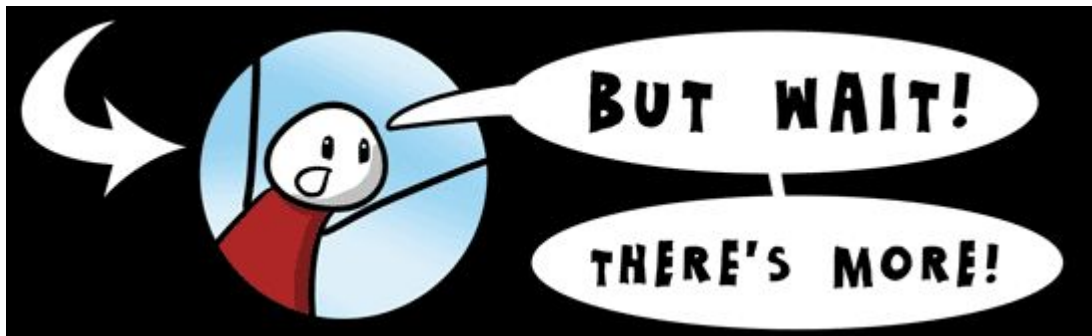
- QUANTIFICADORES

```
if re.search(r'Ah+!', 'A!'):  
    print ('Casou')  
else:  
    print ('Não Casou')
```



THE
DEVELOPER'S
CONFERENCE

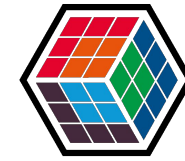
Não Casou



METACARACTERES

- QUANTIFICADORES

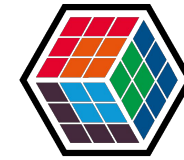
{ }



THE
DEVELOPER'S
CONFERENCE



METACARACTERES



THE
DEVELOPER'S
CONFERENCE

- QUANTIFICADORES

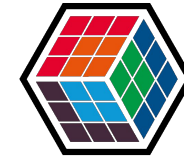
```
re.findall (r'Ah{1,3}!', 'Ah!, Ahhh!,  
Ahhhh!, Ahhhhhhhhhh!')
```

['Ah!', 'Ahhh!']



METACARACTERES

- ANCORAS

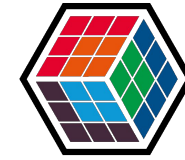


THE
DEVELOPER'S
CONFERENCE



METACARACTERES

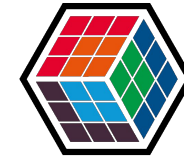
- ANCORAS



THE
DEVELOPER'S
CONFERENCE

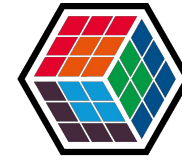


= fome!



THE
DEVELOPER'S
CONFERENCE

NOTEBOOK



THE
DEVELOPER'S
CONFERENCE

OBRIGADO!

AMOMMENDES@GMAIL.COM