

---

---

# Processamento de Linguagem Natural

Aula 01  
Expressões Regulares

---

---

# O que são expressões regulares?

- Cadeias de texto especiais, em uma linguagem formal, para busca/extração de trechos de texto
- As expressões regulares (Regular expression – RE ou ER) podem ser utilizadas para extrair trechos a partir de texto.
- Também conhecidas como RegEx (Regular Expressions)
- Uma ER é uma string textual. *Teste* é uma ER. Assim como `[A-Z]+:\d+`
- Essas strings descrevem padrões para encontrar textos ou posições em um texto

# Para que servem as ER?

- **Encontrar** texto dentro de um corpo textual grande
- **Validar** a conformidade de uma string com um formato desejado
- **Substituir** ou **Inserir** texto em posições demarcadas
- **Quebrar** strings



# Principais aplicações expressões regulares

- Encontrar todos os links em um documento
- Buscar emails, telefones
- Remover/substituir caracteres indesejados
- Normalização de texto (e.g., padronizar o texto convenientemente)
- Divisão em tokens (e.g., divisão em palavras usando os espaços?)
- Radicalização (e.g., lemmatization, stemming)
- Segmentação de frases (e.g., divisão em frases usando a pontuação)



# Padrões básicos de Expressões Regulares

- A ER mais simples é uma cadeia de caracteres

*Regex Belo Horizonte*

A cidade de **Belo Horizonte** foi fundada em 12 de dezembro de 1897

- Importante! Expressões Regulares são *Case-Sensitive*, isso quer dizer que caracteres maiúsculos e minúsculos são vistos como caracteres diferentes.

# Disjunções

- A cadeia de caracteres entre colchetes/parênteses retos (i.e, '[' e ']') especifica uma disjunção de caracteres para a busca.

ER	Padrão	Texto
[abc]	Apenas <b>um</b> caracter a, b ou c	A <b>c</b> idade de Belo Horizonte
[123]	Apenas <b>um</b> dígito 1,2 ou 3	foi fundada em <b>12</b> de dezembro

# Faixa de caracteres

- Ainda utilizando a **Disjunção**, incluímos o caractere traço (-) para configurar a faixa a ser considerada

ER	Padrão	Texto
[a-z]	Apenas <b>uma</b> letra minúscula	A <b>c</b> idade de Belo Horizonte
[A-Z]	Apenas <b>uma</b> letra maiúscula	<b>A</b> cidade de Belo Horizonte
[0-9]	Apenas <b>um</b> dígito	foi fundada em <b>12</b> de dezembro

# Outros padrões

- Negação:  $\wedge$
- Caractere anterior opcional:  $?$
- Caracter *coringa*:  $.$

ER	Padrão	Texto/Matches
[^A]	Não pode ser um A maiúsculo	A <b>c</b> idade de Belo Horizonte
colorid.s?	Depois do <b>d</b> pode ser qualquer caracteres e poderia ter <b>s</b>	<b>colorido, colorida, coloridos, coloridas</b>





# Contadores

Zero ou um a	$a?$
Zero ou mais a	$a^*$
Um ou mais a	$a^+$
Exatamente 3 a	$a\{3\}$
3 ou mais a	$a\{3,\}$
Entre 3 e 6 a	$a\{3,6\}$



# Padrões Especiais

Um espaço	<code>\s</code>
Qualquer caracteres diferente de espaço	<code>\S</code>
Um dígito	<code>\d</code>
Um não-dígito	<code>\D</code>
Uma palavra	<code>\w</code>
Qualque um dos padrões	<code>(a   b   c)</code>

# http://regex101.com

The screenshot displays the regex101.com website interface. The browser address bar shows `https://regex101.com`. The page title is "regular expressions 101". The main content area is divided into several sections:

- SAVE & SHARE**: Includes a "Save Regex" button with a keyboard shortcut of `ctrl+s`.
- FLAVOR**: A list of programming languages with checkboxes: PCRE (PHP) (checked), ECMAScript (JavaScript), Python, and Golang.
- TOOLS**: Includes a "Code Generator" and a "Regex Debugger".
- SPONSOR**: A section for "Flock" with a description: "Simple, intuitive UI and tons of features built specially for Developers. Flock makes collaboration and communication easy."

The central area contains the **REGULAR EXPRESSION** input field with the text `/ colorid.s? / gm` and a status bar indicating "1 match, 10 steps (~0ms)". Below it is the **TEST STRING** input field containing the text "coloridos".

The **EXPLANATION** section on the right provides a detailed breakdown of the regex components:

- / colorid.s? / gm**:
  - `colorid` matches the characters `colorid` literally (case sensitive).
  - `.` matches any character (except for line terminators).
  - `s?` matches the character `s` literally (case sensitive).
  - Quantifier** — Matches between **zero** and **one** times, as many times as possible, giving back as needed (*greedy*).
  - Global pattern flags**:
    - `g` **modifier**: global. All matches (don't return after first match).

The **MATCH INFORMATION** section shows "Match 1" with a "Full match" of "0-9 coloridos".

The **QUICK REFERENCE** section at the bottom right provides a search bar and a list of tokens with their corresponding regex symbols:

Token	Symbol
Between 3 and 6 of a	<code>a{3,6}</code>
Start of string	<code>^</code>
End of string	<code>\$</code>
A word boundary	<code>\b</code>
Non-word boundary	<code>\B</code>

# Falso Positivo & Falso Negativo

- Falso Positivos
  - Identificar cadeias de caracteres que não deveriam ser identificadas
  - Exemplo: Humanidade, idade, humana
- Falso Negativos
  - Não identificar cadeias de caracteres que deveriam ser identificadas

# Falso Positivo & Falso Negativo

- Devemos tratar esses dois problemas de identificação de padrões que acontecem com frequência nas implementações de sistemas de processamento de fala e linguagem
- Reduzir a taxa geral de problemas desse tipo nas aplicações envolvem dois esforços contrários:
  - Aumentar a precisão → Minimiza Falso Positivos
  - Aumentar a cobertura → Minimiza Falso Negativos

# Expressões Regulares

São utilizadas, frequentemente, como primeiro passo para qualquer processamento de texto.

O resultado das ERs é utilizado em classificadores de técnicas de aprendizado de máquina.

# Expressões Regulares em Python

- Expressões regulares são suportadas pelo módulo **re**
- Algumas funções mais utilizadas são:
  - **split** : Quebra a string no *Regex*
  - **findall**: busca todos os padrões em uma string
  - **search**: busca um padrão
  - **match**: identifica uma string inteira ou substring baseado em um padrão

# ELIZA - Exemplo Prático

- Foi um dos primeiros sistemas de PLN / MIT:1964-1966
  - Weizenbaum, Joseph. (1966). ELIZA – A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45.
- Realizava uma conversa bem limitada e superficial com o usuário, imitando respostas.
- Basicamente são procurados padrões textuais na conversa. A resposta de Eliza seria uma substituição desses padrões, com a finalidade de criar uma ilusão de entendimento da conversa.





# ELIZA - Exemplo Prático

- No sistema são procuradas frases como por exemplo:
  - “Você está X”
  - Sendo substituídas “O que faz pensar que eu estou X?”
- Simulação de um terapeuta:
  - As respostas são um espelho do pensamento do paciente.
  - Na tentativa de os mesmos pacientes descubram os motivos pelos quais eles gostariam de mudar.

# ELIZA - Exemplo Prático

- Exemplos de diálogos
  - *Pessoa*: Os pais são todos iguais!!!
  - *Eliza*: DE QUE FORMA?
  - *Pessoa*: Eles estão sempre nos incomodando sobre algo
  - *Eliza*: VOCÊ PODE PENSAR EM UM EXEMPLO ESPECÍFICO?
  - *Pessoa*: Olha só, meus pais me fizeram vir aqui
  - *Eliza*: SEUS PAIS FIZERAM VIR AQUI
  - *Pessoa*: Eles falam que estou deprimido há muito tempo
  - *Eliza*: LAMENTO OUVIR QUE VOCÊ ESTÁ DEPRIMIDO