Material de Apoio

Regex

O que é?

Regex, ou Expressões Regulares, é uma linguagem formal usada para especificar padrões de texto. Esses padrões são utilizados para busca, correspondência e manipulação de strings. As expressões regulares são poderosas ferramentas amplamente utilizadas em linguagens de programação, editores de texto e ferramentas de linha de comando.

Exemplos de Uso

1. **Validação de Formatos:** Verificar se um email ou número de telefone está no formato correto.
2. **Busca e Substituição:** Encontrar e substituir trechos de texto em arquivos.
3. **Extração de Dados:** Extrair dados específicos de grandes blocos de texto.

Como funciona?

O motor de regex é o componente que interpreta e executa a expressão regular. Ele percorre o texto de entrada e tenta encontrar correspondências com o padrão especificado. Existem dois principais tipos de motores de regex: os baseados em NFA (Non-deterministic Finite Automaton) e os baseados em DFA (Deterministic Finite Automato

Processamento de Correspondência

1. **Leitura do Padrão**: O motor de regex começa lendo o padrão especificado.
2. **Percorrendo o Texto:** O motor percorre o texto de entrada caractere por caractere.
3. **Tentativa de Correspondência**: Para cada posição no texto, o motor tenta encontrar uma correspondência com o padrão.
4. **Retorno de Resultados:** Se uma correspondência for encontrada, o motor retorna a posição da correspondência e, opcionalmente, os grupos capturados. Se nenhuma correspondência for encontrada, o motor avança para a próxima posição no texto e tenta novamente.

Estrutura do Regex

Metacaracteres

Metacaracteres são caracteres com significados especiais em expressões regulares. Alguns dos metacaracteres mais comuns incluem:

- .: Corresponde a qualquer caractere (exceto nova linha).
- ^: Indica o início da linha.
- $: Indica o fim da linha.
- *: Corresponde a zero ou mais ocorrências do caractere anterior.
- +: Corresponde a uma ou mais ocorrências do caractere anterior.
- ?: Corresponde a zero ou uma ocorrência do caractere anterior.
- \: Utilizado para escapar metacaracteres.

Estrutura do Regex

Classes de Caracteres

Classes de caracteres permitem especificar um conjunto de caracteres dentro de colchetes []. Por exemplo, [abc] corresponde a qualquer caractere 'a', 'b' ou 'c'.

- \d: Corresponde a qualquer dígito (equivalente a [0-9]).
- \w: Corresponde a qualquer caractere alfanumérico (equivalente a [a-zA-Z0-9_]).
- \s: Corresponde a qualquer espaço em branco.

Quantificadores

Quantificadores especificam quantas vezes um elemento deve ocorrer:

- {n}: Exatamente n vezes.
- {n,}: Pelo menos n vezes.
- {n,m}: Entre n e m vezes.

Grupos e Capturas

Parênteses () são usados para agrupar partes de uma expressão e capturar as correspondências. Por exemplo, (abc)+ corresponde a uma ou mais ocorrências da sequência 'abc'.

Exemplos práticos

Validação de Email:

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

Validação de Número de Telefone (Formato Brasileiro):

```
^\(?\d{2}\)?[\s-]?\d{4,5}-?\d{4}$
```

Busca de URL:

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

Extração de Datas (formato DD/MM/YYYY):

```
\b\d{2}\/\d{2}\/\d{4}\b
```

Dicas e Boas práticas

1. **Teste suas Expressões**: Use ferramentas online para testar e validar suas expressões regulares.
2. **Comente suas Regex**: Adicione comentários explicativos em expressões complexas para melhorar a legibilidade.
3. Use Grupos Nomeados: Facilite a leitura e manutenção utilizando grupos nomeados quando possível.

Compreender e utilizar expressões regulares pode parecer desafiador no início, mas com prática, elas se tornam uma ferramenta indispensável para manipulação de texto em diversas áreas da programação e análise de dados.

E aí, curtiu?

Esperamos que esse resumo tenha enriquecido sua perspectiva estratégica para enfrentar os desafios.

Salve esse PDF para consultar sempre que precisar.

DNC