

リスト記法変換テスト

2025-09-05 版 発行



第 1 章

リスト記法統一の完全デモ

1.1 はじめに

Re:VIEW Extensions により、従来の複数のリスト記法 (emlist、emlist-num、cmd、source) が統一的な `//list` 記法に変換され、豊富なオプションが利用可能になりました。

1.2 変換例

基本的なコードブロック (emlist → list)

変換前の emlist 記法：

リスト 1.3: Python の Hello World

```
def hello():  
    print("Hello, World!")  
    return True
```

これは内部的に `//list` として処理されます。

行番号付きコード (emlistnum → list with lineno)

変換前の emlistnum 記法が以下のように変換されます：

リスト 1.2: 計算機クラス

```
class Calculator:
    def __init__(self):
        self.result = 0

    def add(self, x, y):
        self.result = x + y
        return self.result

    def multiply(self, x, y):
        self.result = x * y
        return self.result
```

行番号が自動的に付与されます。

コマンド例 (cmd → list with fontsize)

シェルコマンドの表示：

リスト 1.3: パッケージのインストール

```
$ npm install -D textlint
$ npm install -D textlint-plugin-review
$ npm install -D textlint-rule-prh
$ npm run lint
```

fontsize=small により、コマンドらしい表示になります。

設定ファイル (source → list with filename)

ファイル名を表示：

リスト 1.3: データベース設定

```
# config/database.yml
development:
  adapter: postgresql
  database: myapp_development
  host: localhost
  port: 5432
  username: myapp
  password: secret

test:
  adapter: postgresql
  database: myapp_test
  host: localhost
  port: 5432
```

1.3 拡張オプションの活用

特定の開始行番号

100 行目から開始する例：

リスト 1.3: エラーハンドリング部分

```
try:
  result = risky_operation()
  process(result)
except ValueError as e:
```

```
        logger.error(f"Value error: {e}")
        raise
    except Exception as e:
        logger.critical(f"Unexpected error: {e}")
        sys.exit(1)
```

インデント表示

Python のネスト構造を見やすく：

リスト 1.3: ネスト構造の例

```
def outer_function():
    def inner_function():
        def deepest_function():
            return "Deep nested"
        return deepest_function()
    return inner_function()

result = outer_function()
print(result)
```

1.4 変換プロセス

変換は以下のコマンドで実行されます：

リスト 1.3: 前処理コマンド

```
# Re:VIEW ファイルの前処理
$ node packages/review-macro-shims/bin/review-preprocess.js \
```

```
"articles/**/*.re" \  
-o output \  
--stats  
  
# 変換後のビルド  
$ cd output  
$ review-pdfmaker config.yml
```

1.5 まとめ

Re:VIEW Extensions のリスト記法統一により：

- コードブロックの記法が `//list` に統一される
- 豊富なオプションによる表現力の向上
- 既存文書の自動移行が可能
- PDF/HTML/ePub で一貫した出力

これにより、技術文書の品質と保守性が向上します。

