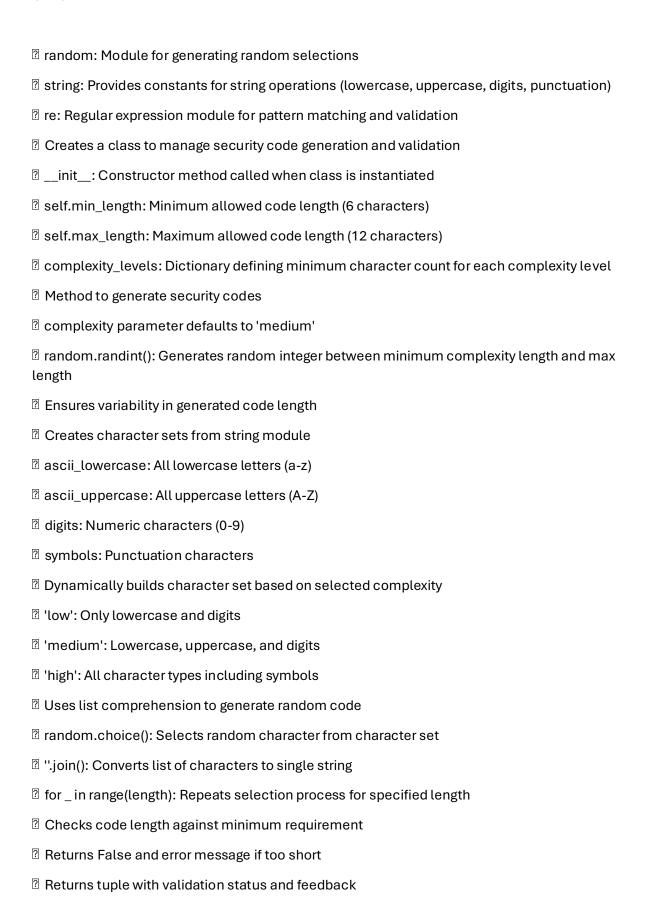
## **CLAUDE AI**



- Dictionary defining regex patterns for different character types
- 2 r'[a-z]': Matches any lowercase letter
- 2 r'[A-Z]': Matches any uppercase letter
- ? r'\d': Matches any digit
- $2 r'[!@#$\%^&*(),.?":{}|<>]': Matches specific symbol characters$
- List comprehension to find missing character types
- 2 checks.items(): Iterates through check dictionary
- 2 re.search(): Searches for pattern in code
- 2 Builds list of character types not present in code
- 1 f any checks failed, generate feedback message
- [2] ', '.join(): Converts list of failed checks to comma-separated string
- Returns validation result and feedback
- Creates instance of SecurityCodeManager
- 2 Provides interactive menu for code generation and validation
- 2 Infinite loop for menu-driven interface
- Displays options to user
- Captures user's menu selection
- 2 Option 1: Generate security code
- Prompts for complexity level
- Defaults to 'medium' if invalid input
- Calls code generation method
- Prints generated code
- 2 Option 2: Validate security code
- Prompts for code input
- ? Calls validation method
- Displays validation result and feedback