

Jesse Hicklen

CS-414

Assignment 3 documentation

September 17, 2025

1. EBNF grammar

(start symbol = <command>):

```
<command> ::= <ls-command> | <cd-command> | <cat-command> | <print-command> | <exec-command>
```

```
<ls-command> ::= "ls" [ <path> ]
```

```
<cd-command> ::= "cd" [ <path> ]
```

```
<cat-command> ::= "cat" <filename>
```

```
<print-command> ::= "print" <filename>
```

```
<exec-command> ::= "exec" <filename>
```

```
<path> ::= "\" | <folder> { "\" <folder> }
```

```
<folder> ::= <folder-name>
```

```
<folder-name> ::= <letter> { <letter> } -- 1..8 letters (A-Z, a-z)
```

```
<filename> ::= <name8> "." <ext3>
```

```
<name8> ::= <letter> <letter> <letter> <letter> <letter> <letter> <letter> <letter>
```

```
<ext3> ::= <letter> <letter> <letter>
```

```
<letter> ::= "A".."Z" | "a".."z"
```

Tokenizer and parser working when run by main:

```
> python -u "c:\cs414.Hicklen\Assignment03\HicklenParserMain.py"
```

```
Input: ls
```

```
Tokens: [Token('COMMAND', 'ls'), Token('EOF', '')]
```

```
AST: LSNode(path=None)
```

```
Input: ls \
```

```
Tokens: [Token('COMMAND', 'ls'), Token('BACKSLASH', '\\'), Token('EOF', '')]
```

```
AST: LSNode(path=[])
```

```
Input: ls folder
```

```
Tokens: [Token('COMMAND', 'ls'), Token('IDENT', 'folder'), Token('EOF', '')]
```

```
AST: LSNode(path=['folder'])
```

```
Input: cd
```

```
Tokens: [Token('COMMAND', 'cd'), Token('EOF', '')]
```

```
AST: CDNode(path=[])
```

```
Input: cd mydocs
```

```
Tokens: [Token('COMMAND', 'cd'), Token('IDENT', 'mydocs'), Token('EOF', '')]
```

```
AST: CDNode(path=['mydocs'])
```

```
Input: cat ABCDEFGH.TXT
```

```
Tokens: [Token('COMMAND', 'cat'), Token('FILENAME', 'ABCDEFGH.TXT'), Token('EOF', '')]
```

```
AST: CatNode(filename='ABCDEFGH.TXT')
```

```
Input: print ABCDEFGH.PRN
```

```
Tokens: [Token('COMMAND', 'print'), Token('FILENAME', 'ABCDEFGH.PRN'), Token('EOF', '')]
```

```
AST: PrintNode(filename='ABCDEFGH.PRN')
```

```
Input: exec ABCDEFGH.EXE
```

```
Tokens: [Token('COMMAND', 'exec'), Token('FILENAME', 'ABCDEFGH.EXE'), Token('EOF', '')]
```

```
AST: ExecNode(filename='ABCDEFGH.EXE')
```

```
Input: ls folder\sub
```

```
Tokens: [Token('COMMAND', 'ls'), Token('IDENT', 'folder'), Token('BACKSLASH', '\\'), Token('IDENT', 'sub'), Token('EOF', '')]
```

```
AST: LSNode(path=['folder', 'sub'])
```

```
Input: exec badname.exe
```

```
Tokens: [Token('COMMAND', 'exec'), Token('IDENT', 'badname'), Token('DOT', '.'), Token('IDENT', 'exe'), Token('EOF', '')]
```

```
ParseError: exec requires an 8.3 filename
```

```
PS C:\cs414.Hicklen>
```

Acknowledgement

I used AI assistance (ChatGPT) to help me better understand the structure and requirements of this assignment. Specifically, it guided me in breaking the project into modular parts (grammar, tokenizer, parser, and main), clarified how to correctly set up Python imports when working across multiple files, and helped troubleshoot issues with running Python from PowerShell and VS Code. The AI also helped me verify that my tokenizer and parser outputs aligned with the EBNF grammar and confirmed that my implementation met the assignment requirements such as enforcing DOS 8.3 filenames, folder name length limits, AST construction, and error handling. All coding and design decisions, as well as the final implementation, are my own.