Team 14 project

Humor Trivia PL
By zackery devers

purpose

We made this for

- Interesting trivia facts to inform
- A good source of humor with sprinkled in funnier trivia responses

background

This project is a Programming language I created in order to educate and humor the user. It is good for lifting spirits while also informing the user of many different types of interesting facts

code

```python
from sly import Lexer, Parser
class MPLLexer(Lexer):
    tokens = {FACT, OPTION}
    ignore = ' \t'
    literals = {':'}
    #DISPLAY = r"DISPLAY"
    FACT = r'fact'
    # Tokens
    OPTION = r'[a-zA-Z_][a-zA-Z0-9_]*'
    @_(r'\n+')
    def newline(self, t):
        self.lineno += t.value.count('\n')
    def error(self, t):
            print("Illegal character '%s'" % t.value[0])
            self.index += 1
class MPLParser(Parser):
    tokens = MPLLexer.tokens
    #print(tokens)
    def __init__(self):
        pass
    @_('FACT ":" OPTION')
    def value(self, p):
        #print(p[2])
        # make the joke joke
        if (p[2] == "one"):
            print("lollipops are sticky")
        elif (p[2] == "two"):
            print("music isnt only heard it is felt")
        elif (p[2] == "three"):
            print("water is clear and pools are blue from the reflection
        elif (p[2] == "four"):
            print("some fungi create zombies then control their minds")
        elif (p[2] == "five"):
            print("oranges werent orange at first")
        elif (p[2] == "six"):
            print("there are no qs in any states name in the us")
        elif (p[2] == "seven"):
            print("a cow-bison hybrid is called a beefoalo")
        elif (p[2] == "eight"):
            print("original apples planted by johny appleseed were meant
        elif (p[2] == "nine"):
            print("scottland has 421 words for snow")
```

```python
            print("a cow-bison hybrid is called a beefoalo")
        elif (p[2] == "eight"):
            print("original apples planted by johny appleseed were meant
        elif (p[2] == "nine"):
            print("scottland has 421 words for snow")
        elif (p[2] == "ten"):
            print("squares can be rectangles but rectangles cant be squar
        elif (p[2] == "eleven"):
            print("the bigger the block of cheese the more holes it has")
        elif (p[2] == "twelve"):
            print("A narwhal's tusk reveals its past living conditions.")
        elif (p[2] == "thirteen"):
            print("The first person convicted of speeding was going eight
        elif (p[2] == "fourteen"):
            print("New car smell is the scent of dozens of chemicals.")
        elif (p[2] == "fifteen"):
            print("The world wastes about 1 billion metric tons of food e
        elif (p[2] == "sixteen"):
            print("The severed head of a sea slug can grow a whole new bo
        elif (p[2] == "seventeen"):
            print("Hair and nails grow faster during pregnancy")
        elif (p[2] == "eighteen"):
            print("The world's smallest reptile was first reported in 202
        elif (p[2] == "nineteen"):
            print("Many feet bones don't harden until you're an adult.")
        elif (p[2] == "twenty"):
            print("this is the last fact on our list")
if __name__ == '__main__':
    lexer = MPLLexer()
    parser = MPLParser()
    #parser.parse(lexer.tokenize("fact : three"))
    while True:
        try:
            text = input('Type -fact : number 1-20- in letters> ')
        except EOFError:
            break
        if text:
            lex = lexer.tokenize(text)
            #for token in lex:
            #   print(token)
            parser.parse(lex)
```

Initializing the begging of code

```
from sly import Lexer, Parser
class MPLLexer(Lexer):
    tokens = {FACT, OPTION}
    ignore = ' \t'
    literals = {':'}
    #DISPLAY = r"DISPLAY"
    FACT = r'fact'
    # Tokens
    OPTION = r'[a-zA-Z_][a-zA-Z0-9_]*'
    @_(r'\n+')
    def newline(self, t):
        self.lineno += t.value.count('\n')
    def error(self, t):
            print("Illegal character '%s'" % t.value[0])
            self.index += 1
```

Defining the PL

```
16  class MPLParser(Parser):
17      tokens = MPLLexer.tokens
18      #print(tokens)
19      def __init__(self):
20          pass
21      @_('FACT ":" OPTION')
```

Library of facts

```
    if (p[2] == "one"):
        print("lollipops are sticky")
    elif (p[2] == "two"):
        print("music isnt only heard it is felt")
    elif (p[2] == "three"):
        print("water is clear and pools are blue from the reflection of the sky.")
    elif (p[2] == "four"):
        print("some fungi create zombies then control their minds")
    elif (p[2] == "five"):
        print("oranges werent orange at first")
    elif (p[2] == "six"):
        print("there are no qs in any states name in the us")
    elif (p[2] == "seven"):
        print("a cow-bison hybrid is called a beefoalo")
    elif (p[2] == "eight"):
        print("original apples planted by johny appleseed were meant for hard apple cider due to being too biter")
    elif (p[2] == "nine"):
        print("scottland has 421 words for snow")
    elif (p[2] == "ten"):
        print("squares can be rectangles but rectangles cant be squares")
    elif (p[2] == "eleven"):
        print("the bigger the block of cheese the more holes it has")
    elif (p[2] == "twelve"):
        print("A narwhal's tusk reveals its past living conditions.")
    elif (p[2] == "thirteen"):
        print("The first person convicted of speeding was going eight mph.")
    elif (p[2] == "fourteen"):
        print("New car smell is the scent of dozens of chemicals.")
    elif (p[2] == "fifteen"):
        print("The world wastes about 1 billion metric tons of food each year.")
    elif (p[2] == "sixteen"):
        print("The severed head of a sea slug can grow a whole new body.")
    elif (p[2] == "seventeen"):
        print("Hair and nails grow faster during pregnancy")
    elif (p[2] == "eighteen"):
        print("The world's smallest reptile was first reported in 2021.")
    elif (p[2] == "nineteen"):
        print("Many feet bones don't harden until you're an adult.")
    elif (p[2] == "twenty"):
        print("this is the last fact on our list")
name   == '__main__':
```

The main

```
if __name__ == '__main__':
    lexer = MPLLexer()
    parser = MPLParser()
    #parser.parse(lexer.tokenize("fact : three"))
    while True:
        try:
            text = input('Type -fact : number 1-20- in letters> ')
        except EOFError:
            break
        if text:
            lex = lexer.tokenize(text)
            #for token in lex:
            #   print(token)
            parser.parse(lex)
```