Hangman Game

Making Game with Python (1)

Zhihong (John) Zeng & Andrew Zeng

Class 1

- Review and test
- Check a letter in a string
- Hangman game flowchart
- choose_word
- is_word_guessed
- get_guessed_word
- guess_loop
- Hangman game

Review

- import time module
- Escape character and multiline string
- User-defined function
- While loop
- Boolean operators: and, or , not

Test

```
# find the bugs
A = ‘It’s a test’

# what will be printed
print(‘Welcome\nToday is Sunday’)

def addition(x, y=0):
        ans = x + y
        return ans

print(addition(1, 2))
print(addition(1))
```

Test

```
# find the bugs
A = 'It's a test'

# what will be printed
print('Welcome\nToday is Sunday')

def addition(x, y=0):
        ans = x + y
        return ans

print(addition(1, 2))
print(addition(1))
```

```
# correction
A = 'It\'s a test'
A = "It's a test"

Welcome
Today is Sunday


3
1
```

Test

```
# what will be printed

x = 5
while x > 0:
        print(x)
        x -= 1  # x = x-1



x = 5
while x:  #  Boolean(x) = False if x=0, x=” or x =[]. Otherwise True
        print(x)
        x -= 1
```

Test

```
# what will be printed

x = 5
while x > 0:
        print(x)
        x -= 1



x = 5
while x:
        print(x)
        x -= 1
```

0==False

```
# what will be printed

5
4
3
2
1

5
4
3
2
1
```

Check a item in a string

A = ‘abcd’

print(‘c’ in A) → True

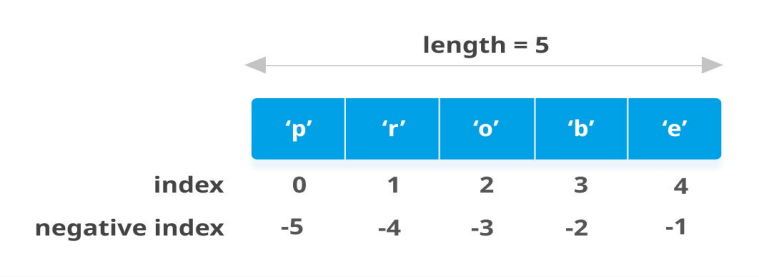
print(‘e’ in A) → False

print(‘e’ not in A) → True

list

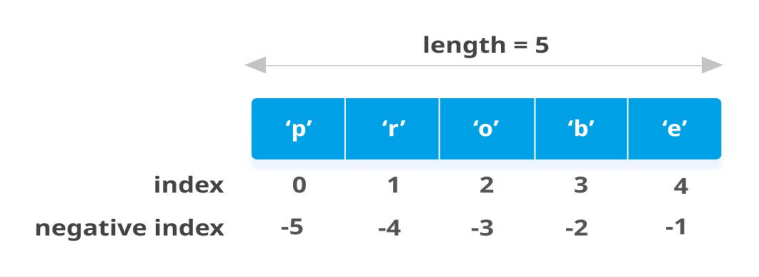
- A list is a collection which is ordered and changeable
- Syntax:
  - Square brackets: […]
- Create a list
  - mylist = [1, 2, 5, 6]
  - mylist = [‘a’, ‘b’, ‘apple’]
  - mylist = [‘a’, 1, ‘apple’]
  - mylist = [[1,2], [3,4]]
- Get the size of list
  - len(mylist)

list: access items



- mylist = [‘p’, ‘r’, ‘o’, ‘b’, ‘e’]
- print(mylist[0]) ‘p’
- print(mylist[2]) ‘o’
- print(mylist[-1]) ‘e’
- print(mylist[0:2]) [‘p’, ‘r’]

list: change item



- mylist = [‘p’, ‘r’, ‘o’, ‘b’, ‘e’]
- mylist[0] = ‘a’
- print(mylist) → [‘a’, ‘r’, ‘o’, ‘b’, ‘e’]
- mylist.append(‘t’)
- print(mylist) → [‘a’, ‘r’, ‘o’, ‘b’, ‘e’, ‘t’]

list: for loop

```
a = list(range(6))

print(a)

for x in a:

        print(x)
```

list and string

- str1 = ‘probe’
- list1 = list(str1)
- str1_1 = ‘’.join(list1)

[‘p’, ‘r’, ‘o’, ‘b’, ‘e’]
‘probe’

- str2 = ‘apple orange’
- list2 = str2.split(‘ ‘)
- str2_0 = ‘,’.join(list2)

[‘apple’, ‘orange’]
‘apple,orange’

Check a item in a list or string

A = [1, 2, 3, 4]

print(2 in A)  True

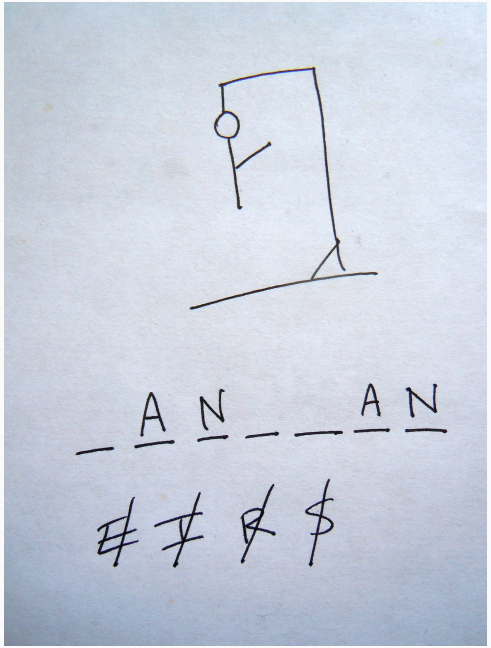
A=[‘a’, ‘b’, ‘c’, ‘d’]

print(‘c’ in A)  True

A = ‘abcd’

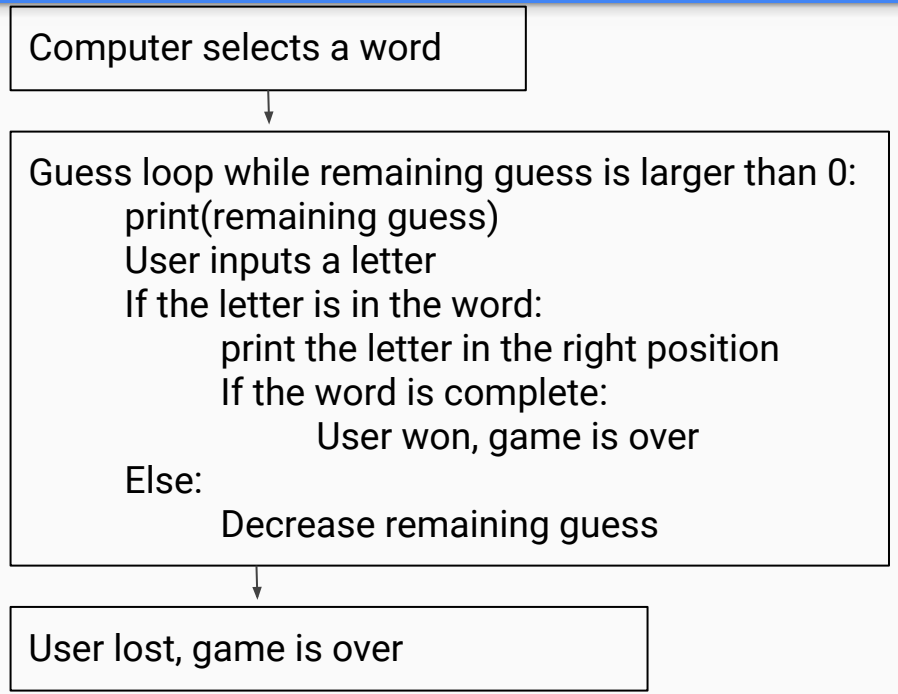
print(‘c’ in A)  True

Hangman Game



demo

Hangman Game Flowchart



User-defined functions

```
import random

def choose_word():
    wordlist = 'ant bear cat dog beer'.split()
    print(wordlist)
    w = random.choice(wordlist)
    return w

def is_word_guessed(secrete_word, letters_guessed):
    for x in secrete_word:
        if x not in letters_guessed:
            return False
    return True

def get_guessed_word(secrete_word, letter_guessed):
    word = ''
    for x in secrete_word:
        if x in letter_guessed:
            word += x
        else:
            word += '_'
    return word
```

Function test

```
from hangman_2020 import *

print(choose_word())
print(choose_word())

ans = is_word_guessed('beer', 'bre')
assert ans==True, 'fail'

ans = is_word_guessed('beer', 'br')
assert ans == False, 'fail'

ans = get_guessed_word('hangman', 'an')
assert ans == '_an__an', 'fail in get_guessed_word'
```

Guess loop and other

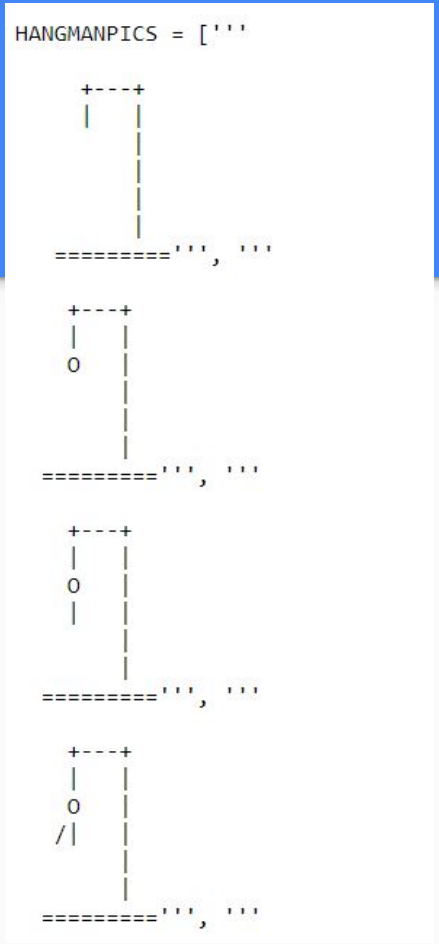
```
def guess_loop(secrete_word, max_guess):
    guessed = ''
    while max_guess > 0:
        print(f'You have {max_guess} guesses left')
        letter = input('Please guess a letter: ').lower()
        if letter in secrete_word:
            if letter in guessed:
                print('That letter has already been guessed')
                continue
            else:
                guessed += letter
                guessed_word = get_guessed_word(secrete_word, guessed)
                print(f'Good guess: {guessed_word}')
                if is_word_guessed(secrete_word, guessed):
                    print('Congrats, You won\n')
                    return
        else:
            guessed_word = get_guessed_word(secrete_word, guessed)
            print(f'Oops. That letter is not in my word: {guessed_word}')
            max_guess -= 1

    print('Sorry, you ran out of guesses.')

def hangman(max_guess):
    secrete_word = choose_word()
    print("Welcome the game")
    guess_loop(secrete_word, max_guess)

if __name__ == '__main__':
    hangman(4)
```

Function test

```
from hangman_2020 import *

print(choose_word())
print(choose_word())

ans = is_word_guessed('beer', 'bre')
assert ans==True, 'fail'

ans = is_word_guessed('ant', 'an')
assert ans == False, 'fail'

ans = get_guessed_word('banana', 'a')
assert ans == '_a_a_a', 'fail in get_guessed_word'

guess_loop('dog', 3)
```

Add visualization:

- HANGMANPICS
- Hangman game with HANGMANPICS



HANGMANPICS

- HANGMANPICS is a list of multi-line string
- from hangman_pictures import HANGMANPICS

(github.com/zhihongzeng2002/pythongame/tree/master/1: hangman_pictures.py)

Code change

```
from hangman_pictures import HANGMANPICS

def guess_loop_2(secrete_word): ###
    max_guess = len(HANGMANPICS) ###
    guessed = ''
    while max_guess > 1:  ###
        print(f'You have {max_guess-1} guessed left') ###
        print(HANGMANPICS[-max_guess])  ###
        letter = input('Please guess a letter: ').lower()
        if letter in secrete_word:
            if letter in guessed:
                print('That letter has already been guessed')
            else:
                guessed += letter
                guessed_word = get_guessed_word(secrete_word, guessed)
                print(f'Good guess: {guessed_word}')

                if is_word_guessed(secrete_word, guessed):
                    print('Congrats, you won.\n')
                    return

        else:
            guessed_word = get_guessed_word(secrete_word, guessed)
            print(f'Oops. That letter is not in my word: {guessed_word}')
            max_guess -= 1
    print(HANGMANPICS[-1])  ###
    print(f'Sorry, you ran out of guesses. My word is {secrete_word}')


def hangman(): ###
    secrete_word = choose_word()
    print('Welcome the game')
    guess_loop_2(secrete_word)   ###

if __name__ == '__main__':
    hangman()     ###
```