

1.

```
1 import string
2
3 def isWordGuessed(secretWord, lettersGuessed):
4     for char in secretWord:
5         if char not in lettersGuessed:
6             return False
7     return True
8
9 def getGuessedWord(secretWord, letterGuessed):
10    strg = ''
11    for char in secretWord:
12        if char in letterGuessed:
13            strg+=char
14        else:
15            strg+=('_')
16    return strg
17
18 def getAvailableLetters(lettersGuessed):
19    strg = string.ascii_lowercase
20    for char in lettersGuessed:
21        if char in strg:
22            strg = strg.replace(char, '')
23    return strg
24
25 def Hangaroo(secretWord):
26    intro = str(len(secretWord))
27    lettersGuessed = []
28    guess = str
29    mistakesMade = 8
30    wordGuessed = False
31
32    print("Welcome to Hangaroo!")
33    print("I am thinking of a word " + intro + " letters long.")
34    print("-----")
35
36    while mistakesMade > 0 and mistakesMade <= 8 and wordGuessed is False:
37        if secretWord == getGuessedWord(secretWord, lettersGuessed):
```

Variable explorer

Name	Type	Size	Value
------	------	------	-------

IPython console

```
In [72]: isWordGuessed("chocolate",["c","h","a"])
Out[72]: False

In [73]: isWordGuessed("apple",["a","p","l","e"])
Out[73]: True

In [74]:
```

iswordguessed('apple', ['a', 'p', 'l', 'e'])

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 19 Column: 34 Memory: 81 %

2.

```
1 import string
2
3 def isWordGuessed(secretWord, lettersGuessed):
4     for char in secretWord:
5         if char not in lettersGuessed:
6             return False
7     return True
8
9 def getGuessedWord(secretWord, letterGuessed):
10    strg = ''
11    for char in secretWord:
12        if char in letterGuessed:
13            strg+=char
14        else:
15            strg+=('_')
16    return strg
17
18 def getAvailableLetters(lettersGuessed):
19    strg = string.ascii_lowercase
20    for char in lettersGuessed:
21        if char in strg:
22            strg = strg.replace(char, '')
23    return strg
24
25 def Hangaroo(secretWord):
26    intro = str(len(secretWord))
27    lettersGuessed = []
28    guess = str
29    mistakesMade = 8
30    wordGuessed = False
31
32    print("Welcome to Hangaroo!")
33    print("I am thinking of a word " + intro + " letters long.")
34    print("-----")
35
36    while mistakesMade > 0 and mistakesMade <= 8 and wordGuessed is False:
37        if secretWord == getGuessedWord(secretWord, lettersGuessed):
```

Variable explorer

Name	Type	Size	Value
------	------	------	-------

IPython console

```
In [72]: isWordGuessed("chocolate",["c","h","a"])
Out[72]: False

In [73]: isWordGuessed("apple",["a","p","l","e"])
Out[73]: True

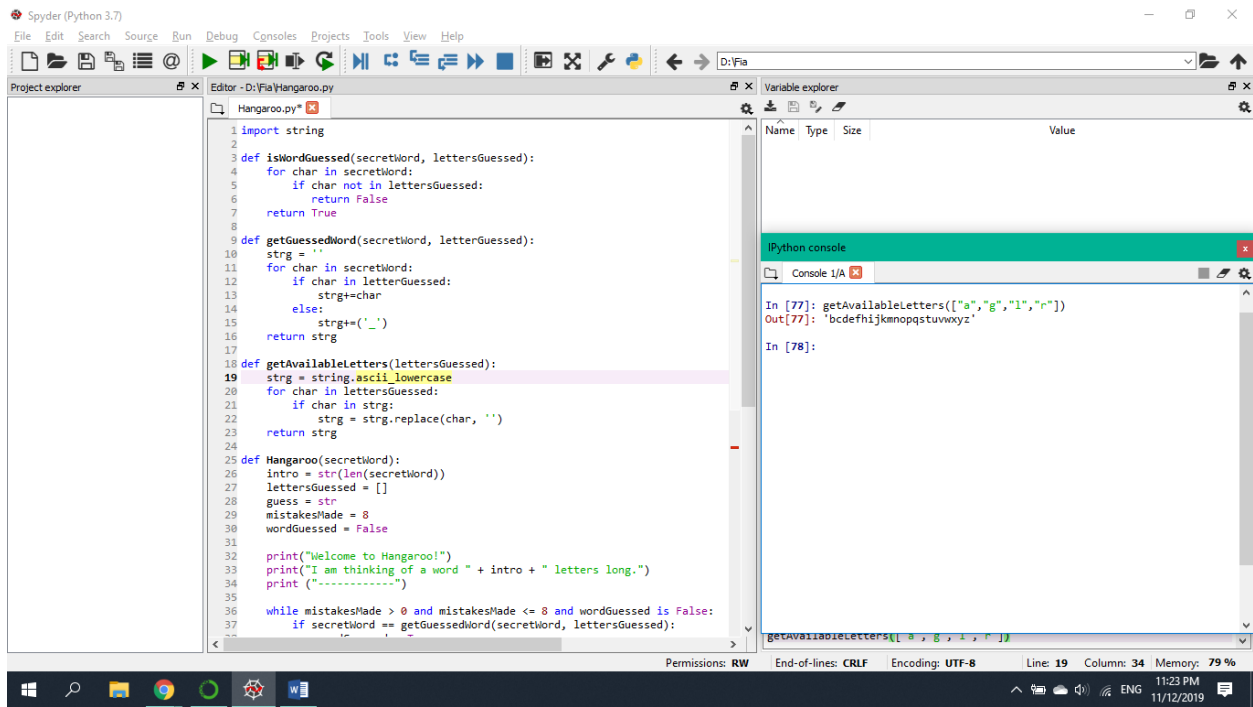
In [74]: getGuessedWord("mango",["m","o"])
Out[74]: 'm__o'

In [75]:
```

getguessedword('mango', ['m', 'o'])

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 19 Column: 34 Memory: 79 %

3.



The image shows the Spyder Python IDE interface. The main editor window displays the code for a Hangaroo game. The code includes functions for checking if a word is guessed, building the guessed word string, and getting available letters. The console shows the output of the getAvailableLetters function.

```
1 import string
2
3 def isWordGuessed(secretWord, lettersGuessed):
4     for char in secretWord:
5         if char not in lettersGuessed:
6             return False
7     return True
8
9 def getGuessedWord(secretWord, letterGuessed):
10    strg = ''
11    for char in secretWord:
12        if char in letterGuessed:
13            strg+=char
14        else:
15            strg+=('_' )
16    return strg
17
18 def getAvailableLetters(lettersGuessed):
19    strg = string.ascii_lowercase
20    for char in lettersGuessed:
21        if char in strg:
22            strg = strg.replace(char, '')
23    return strg
24
25 def Hangaroo(secretWord):
26    intro = str(len(secretWord))
27    lettersGuessed = []
28    guess = str
29    mistakesMade = 8
30    wordGuessed = False
31
32    print("Welcome to Hangaroo!")
33    print("I am thinking of a word " + intro + " letters long.")
34    print("-----")
35
36    while mistakesMade > 0 and mistakesMade <= 8 and wordGuessed is False:
37        if secretWord == getGuessedWord(secretWord, lettersGuessed):
```

Variable explorer:

Name	Type	Size	Value
------	------	------	-------

Python console:

```
In [77]: getAvailableLetters(["a","g","l","r"])
Out[77]: 'bcdefhijkmnopqstuvwxzy'

In [78]:
```

getAvailableLetters(['a', 'g', 'l', 'r'])

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 19 Column: 34 Memory: 79 %

11:23 PM 11/12/2019