

Hangman

[Hangman](#) is a game where a player must guess a word letter by letter. Initially, he gets to see of how many letters the word consists. The player only gets a limited set of chances to guess the word. Every time he guesses a wrong letter, he has one less chance to guess the word. Usually, a drawing of a hangman indicates how many mistakes the player has made, and indirectly, how many chances are left.

Assignment

Write a class `Hangman` that has the method `guessLetter`, among others. Look at the example below to deduct how the class `Hangman` can work. Also, pay attention to the special cases.

Example

```
>>> h = Hangman('BaNana')
>>> print(h)
You have 6 more chances.
.....
>>> h.guessLetter('x')
Wrong: letter x does not occur in the word.
You have 5 more chances.
.....
>>> h.guessLetter('b')
Correct: letter b occurs 1 times in the word.
You have 5 more chances.
B.....
>>> h.guessLetter('n')
Correct: letter n occurs 2 times in the word.
You have 5 more chances.
B.N.n.
>>> h.guessLetter(5)
Traceback (most recent call last):
AssertionError: argument is not a letter
>>> h.guessLetter('kiwi')
Traceback (most recent call last):
AssertionError: argument is not letter
>>> h.guessLetter('e')
Wrong: letter e does not occur in the word.
You have 4 more chances.
B.N.n.
>>> h.guessLetter('n')
Traceback (most recent call last):
AssertionError: letter has already been guessed
>>> h.guessLetter('a')
Correct: letter a occurs 3 times in the word.
Congratulations! You have guessed the word!
BaNana
>>> print(h)
Congratulations! You have guessed the word!
BaNana
>>> h.guessLetter('a')
Sorry, the game is over.
>>> print(h)
Congratulations! You have guessed the word!
```

Example

```
>>> h = Hangman('strawberry', chances=3)
>>> print(h)
You have 3 more chances.
.....
>>> h.guessLetter('a')
Correct: letter a occurs 1 times in the word.
You have 3 more chances.
...a.....
>>> h.guessLetter('b')
Correct: letter b occurs 1 times in the word.
You have 3 more chances.
...a.b....
>>> h.guessLetter('c')
Wrong: letter c does not occur in the word.
You have 2 more chances.
...a.b....
>>> h.guessLetter('y')
Correct: letter y occurs 1 times in the word.
You have 2 more chances.
...a.b...y
>>> h.guessLetter('e')
Correct: letter e occurs 1 times in the word.
You have 2 more chances.
...a.be..y
>>> h.guessLetter('f')
Wrong: letter f does not occur in the word.
You have 1 more chance.
...a.b...y
>>> h.guessLetter('h')
Wrong: letter h does not occur in the word.
Oops, you have been hung.
strawberry
>>> h.guessLetter('h')
Sorry, the game is over.
>>> h.guessLetter('banana')
Sorry, the game is over.
>>> print(h)
Oops, you have been hung.
strawberry
```

[Galgje](#) is een spelletje waarbij een speler letter voor letter de letters van een woord moet raden. Initieel krijgt de speler enkel te zien hoeveel letters het woord telt. De speler krijgt slechts een beperkt aantal beurten om het woord te raden, waarbij er telkens een beurt verloren gaat als de speler om een letter vraagt die niet in het woord voorkomt. Doorgaans wordt op basis van een tekening van een galg aangegeven hoeveel fouten de speler reeds heeft gemaakt, en dus onrechtstreeks ook hoeveel fouten hij nog mag maken.

Opgave

Schrijf een klasse Galgje die onder andere een methode `raadLetter` heeft. Bekijk onderstaand voorbeeld om af te leiden hoe men met de klasse Galgje kan werken. Let daarbij ook op de speciale gevallen.

Voorbeeld

```
>>> g = Galgje('BaNaan')
>>> print(g)
Je hebt nog 6 beurten.
.....
>>> g.raadLetter('x')
Fout: letter x komt niet voor in het woord.
Je hebt nog 5 beurten.
.....
>>> g.raadLetter('b')
Correct: letter b komt 1 keer voor in het woord.
Je hebt nog 5 beurten.
B.....
>>> g.raadLetter('n')
Correct: letter n komt 2 keer voor in het woord.
Je hebt nog 5 beurten.
B.N..n
>>> g.raadLetter(5)
Traceback (most recent call last):
AssertionError: argument is geen letter
>>> g.raadLetter('kiwi')
Traceback (most recent call last):
AssertionError: argument is geen letter
>>> g.raadLetter('e')
Fout: letter e komt niet voor in het woord.
Je hebt nog 4 beurten.
B.N..n
>>> g.raadLetter('n')
Traceback (most recent call last):
AssertionError: letter is al eens geprobeerd
>>> g.raadLetter('a')
Correct: letter a komt 3 keer voor in het woord.
Proficiat! Je hebt het woord geraden!
BaNaan
>>> print(g)
Proficiat! Je hebt het woord geraden!
BaNaan
>>> g.raadLetter('a')
Sorry, het spel is reeds voorbij.
>>> print(g)
Proficiat! Je hebt het woord geraden!
BaNaan
```

Voorbeeld

```
>>> g = Galgje('aardbei', beurten=3)
>>> print(g)
Je hebt nog 3 beurten.
.....
>>> g.raadLetter('a')
Correct: letter a komt 2 keer voor in het woord.
Je hebt nog 3 beurten.
aa.....
>>> g.raadLetter('b')
Correct: letter b komt 1 keer voor in het woord.
Je hebt nog 3 beurten.
aa..b..
>>> g.raadLetter('c')
```

```
Fout: letter c komt niet voor in het woord.
Je hebt nog 2 beurten.
aa..b..
>>> g.raadLetter('d')
Correct: letter d komt 1 keer voor in het woord.
Je hebt nog 2 beurten.
aa.db..
>>> g.raadLetter('e')
Correct: letter e komt 1 keer voor in het woord.
Je hebt nog 2 beurten.
aa.dbe.
>>> g.raadLetter('f')
Fout: letter f komt niet voor in het woord.
Je hebt nog 1 beurt.
aa.dbe.
>>> g.raadLetter('g')
Fout: letter g komt niet voor in het woord.
Ai, je bent opgehangen.
aardbei
>>> g.raadLetter('h')
Sorry, het spel is reeds voorbij.
>>> g.raadLetter('banaan')
Sorry, het spel is reeds voorbij.
>>> print(g)
Ai, je bent opgehangen.
aardbei
```