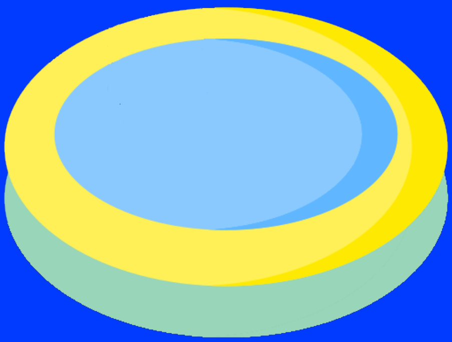
{EPITECH}

WADING POOL

< DAY 07 />



WADING POOL

Today, there is no challenge, because the entire day is already a challenge!
That's why you have multiple game proposals.

If you need a little break in your day, we suggest you to play:

- ✓ lightbot ;
- ✓ music lab ;
- ✓ css diner ;
- ✓ css grid garden
- ✓ flexbox defense ;
- ✓ flexbox froggy.

We also encourage you to resume your game(s) later.

Today's objective is to get some fun building (and playing) the guessing game hangman.

Objective and rules of OUR game

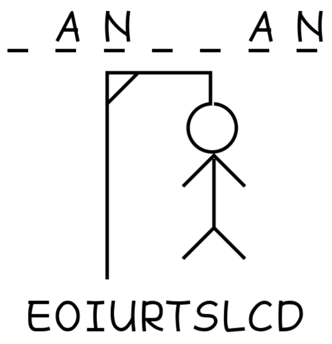
Players have to guess a word with a minimum of penalties.
Each turn, a player suggests a letter:

- ✓ if the word contains it, each occurrence of the letter is revealed;
- ✓ if the word does not contain it, the player gets 1 penalty.

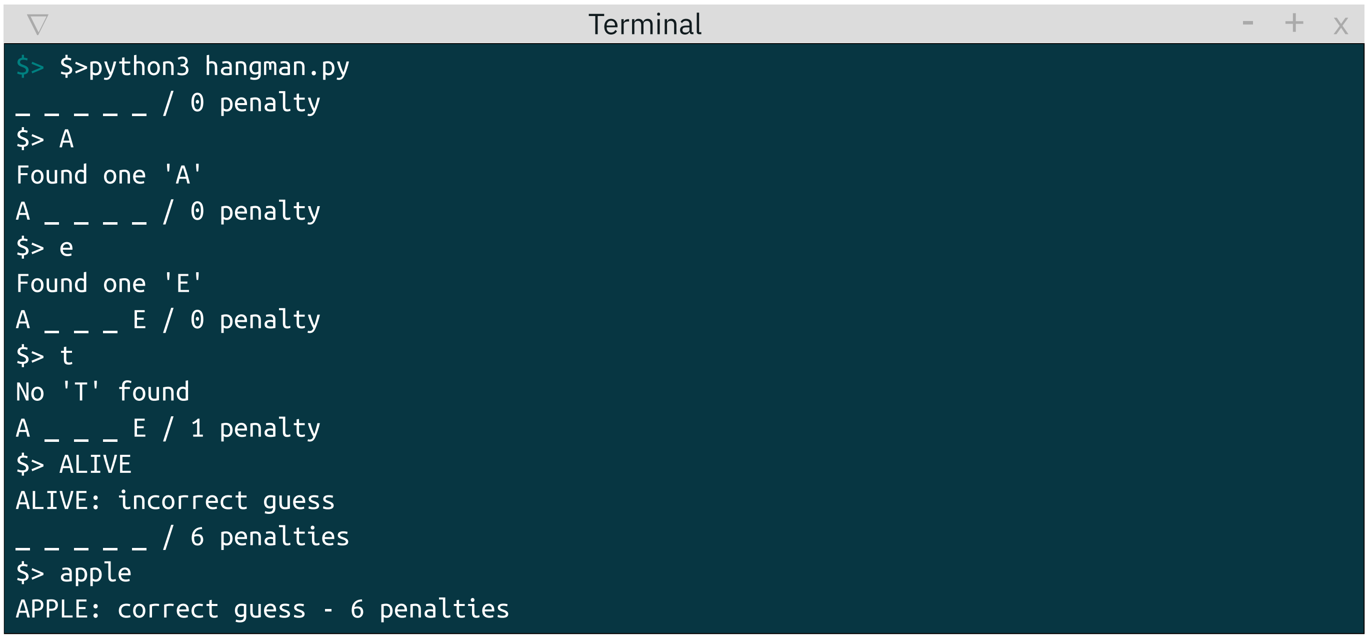
At any time, the player can propose a full word:

- ✓ if the word is the one to be guessed, the player wins:
- ✓ else, the player gets 5 penalties.

If the number of penalties exceeds 12, the player looses.



Here is a game example:


Terminal

```
$> $>python3 hangman.py
_ _ _ _ _ / 0 penalty
$> A
Found one 'A'
A _ _ _ _ / 0 penalty
$> e
Found one 'E'
A _ _ _ E / 0 penalty
$> t
No 'T' found
A _ _ _ E / 1 penalty
$> ALIVE
ALIVE: incorrect guess
_ _ _ _ _ / 6 penalties
$> apple
APPLE: correct guess - 6 penalties
```

No graphical interface here, the game must be playable in a terminal.
Feel free to adapt the display of the game as you fancy.

First brick

Write a function that takes an integer as parameter and prints ”You loose!” if the parameter is greater or equal than 12.

Next brick

Write a snippet of code to return a **random** item from the set `{1, 2, 3, 4, 5, 6}`.
Encapsulate this piece of code inside a function.

First package

Install the english-words-py package. Then, create a program that:

✓ imports the `english_word_lower_set` from this package ;
✓ prints a random word from this set.

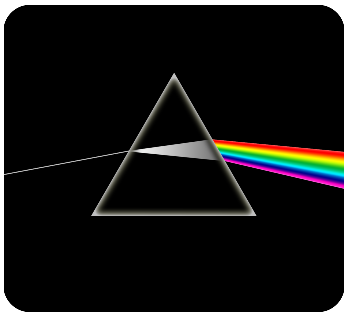
The well named **PIP** (Package Installer for Python) is the usual tool for this job.

Another brick (in the wall)

Write a function that:

✓ takes a string as parameter ;
✓ finds the length `n` of this string ;
✓ prints a string made of `n` pairs (underscore space).

For instance, the input `funny` provides the output _ _ _ _ _.



Pseudocode

Before coding a program, a good way to start is to define the pseudocode: it will describe the steps your program will follow.

Write down the pseudocode for your hangman game.
Then, test and challenge it with a fellow comrade.

If you're stuck, look for some inspiration on the web or in the class.

Implementation

You've define some few basic functions and describe the main steps.
It is now time to dive in. Implement the entire game.
Do as you please to reach this goal.

Never remain stuck, but ask your fellow comrades or your local pedago or the web.

As usual, we warmly recommend you to try a no-code tool, such as Flowgorithm, that generates code from graphical flowcharts.

Give us more

Make your game customizable, by adding few arguments to its program, such as:

- ✓ the number of penalties ;
- ✓ the length of the unknown word ;
- ✓ the theme of the unknown word ;
- ✓ a file containing a word list to pick in ;
- ✓ a time limit ;
- ✓ ... anything you fancy ...

A nice tool to parse Python's arguments is argparse.

v 2