

---

# **simpleai Documentation**

*Release 0.8.1*

**Juan Pedro Fisanotti**

**Sep 07, 2017**



---

## Contents

---

|          |                                    |           |
|----------|------------------------------------|-----------|
| <b>1</b> | <b>Simple AI</b>                   | <b>3</b>  |
| <b>2</b> | <b>Installation</b>                | <b>5</b>  |
| <b>3</b> | <b>Examples</b>                    | <b>7</b>  |
| <b>4</b> | <b>More detailed documentation</b> | <b>9</b>  |
| <b>5</b> | <b>Help and discussion</b>         | <b>11</b> |
| <b>6</b> | <b>Authors</b>                     | <b>13</b> |



Contents:

- *Introduction (this page)*
- **Search Problems:**
  - search\_problems
  - local\_search\_problems
  - constraint\_satisfaction\_problems
  - search\_viewers
- **Machine Learning:**
  - classification



Project home: <http://github.com/simpleai-team/simpleai>

This lib implements many of the artificial intelligence algorithms described on the book “Artificial Intelligence, a Modern Approach”, from Stuart Russel and Peter Norvig. We strongly recommend you to read the book, or at least the introductory chapters and the ones related to the components you want to use, because we won’t explain the algorithms here.

This implementation takes some of the ideas from the Norvig’s implementation (the [aima-python](#) lib), but it’s made with a more “pythonic” approach, and more emphasis on creating a stable, modern, and maintainable version. We are testing the majority of the lib, it’s available via pip install, has a standard repo and lib architecture, well documented, respects the python pep8 guidelines, provides only working code (no placeholders for future things), etc. Even the internal code is written with readability in mind, not only the external API.

At this moment, the implementation includes:

- **Search**
  - Traditional search algorithms (not informed and informed)
  - Local Search algorithms
  - Constraint Satisfaction Problems algorithms
  - Interactive execution viewers for search algorithms (web-based and terminal-based)
- **Machine Learning**
  - Statistical Classification





## CHAPTER 2

---

### Installation

---

Just get it:

```
pip install simpleai
```

And if you want to use the interactive search viewers, also install:

```
pip install pydot flask
```

You will need to have pip installed on your system. On linux install the python-pip package, on windows follow [this](#). Also, if you are on linux and not working with a virtualenv, remember to use sudo for both commands (sudo pip install ...).



Simple AI allows you to define problems and look for the solution with different strategies. Another samples are in the `samples` directory, but here is an easy one.

This problem tries to create the string “HELLO WORLD” using the A\* algorithm:

```
from simpleai.search import SearchProblem, astar

GOAL = 'HELLO WORLD'

class HelloProblem(SearchProblem):
    def actions(self, state):
        if len(state) < len(GOAL):
            return list(' ABCDEFGHIJKLMNOPQRSTUVWXYZ')
        else:
            return []

    def result(self, state, action):
        return state + action

    def is_goal(self, state):
        return state == GOAL

    def heuristic(self, state):
        # how far are we from the goal?
        wrong = sum([1 if state[i] != GOAL[i] else 0
                     for i in range(len(state))])
        missing = len(GOAL) - len(state)
        return wrong + missing

problem = HelloProblem(initial_state='')
result = astar(problem)

print(result.state)
print(result.path())
```



## CHAPTER 4

---

More detailed documentation

---

You can read the docs online [here](#). Or for offline access, you can clone the project code repository and read them from the `docs` folder.



## CHAPTER 5

---

Help and discussion

---

Join us at the [Simple AI google group](#).





## CHAPTER 6

---

### Authors

---

- Many people you can find on the [contributors section](#).
- Special acknowledgements to [Machinalis](#) for the time provided to work on this project. Machinalis also works on some other very interesting projects, like [Quepy](#) and [more](#).