

# Files

hangman	Language	files	blank	comment	code
— hangman.db	-----	-----	-----	-----	-----
— hangman.py	HTML	3	12	3	115
— static	Python	1	28	6	68
— bootstrap.min.css	Javascript	3	8	13	30
— bootstrap.min.js	CSS	2	6	6	19
— jquery.min.js					
— main.css					
— main.js					
— templates					
— home.html					
— index.html					
— play.html					
— words.txt					

## hangman.py

```
1  import random
2
3  import flask
4  from flask_sqlalchemy import SQLAlchemy
5
6  app = flask.Flask(__name__)
7
8  # Database
9
10 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///hangman.db'
11 db = SQLAlchemy(app)
12
13 # Model
14
15 def random_pk():
16     return random.randint(1e9, 1e10)
17
18 def random_word():
19     words = [line.strip() for line in open('words.txt') if len(line) > 10]
20     return random.choice(words).upper()
21
22 class Game(db.Model):
23     pk = db.Column(db.Integer, primary_key=True, default=random_pk)
24     word = db.Column(db.String(50), default=random_word)
25     tried = db.Column(db.String(50), default='')
26     player = db.Column(db.String(50))
27
28     def __init__(self, player):
29         self.player = player
30
31     @property
32     def errors(self):
33         return ''.join(set(self.tried) - set(self.word))
34
35     @property
36     def current(self):
37         return ''.join([c if c in self.tried else '_' for c in self.word])
38
39     @property
40     def points(self):
41         return 100 + 2*len(set(self.word)) + len(self.word) - 10*len(self.errors)
```

```

42
43     # Play
44
45     def try_letter(self, letter):
46         if not self.finished and letter not in self.tried:
47             self.tried += letter
48             db.session.commit()
49
50     # Game status
51
52     @property
53     def won(self):
54         return self.current == self.word
55
56     @property
57     def lost(self):
58         return len(self.errors) == 6
59
60     @property
61     def finished(self):
62         return self.won or self.lost
63
64
65     # Controller
66
67     @app.route('/')
68     def home():
69         games = sorted(
70             [game for game in Game.query.all() if game.won],
71             key=lambda game: -game.points)[:10]
72         return flask.render_template('home.html', games=games)
73
74     @app.route('/play')
75     def new_game():
76         player = flask.request.args.get('player')
77         game = Game(player)
78         db.session.add(game)
79         db.session.commit()
80         return flask.redirect(flask.url_for('play', game_id=game.pk))
81
82     @app.route('/play/<game_id>', methods=['GET', 'POST'])
83     def play(game_id):
84         game = Game.query.get_or_404(game_id)
85
86         if flask.request.method == 'POST':
87             letter = flask.request.form['letter'].upper()
88             if len(letter) == 1 and letter.isalpha():
89                 game.try_letter(letter)
90
91         if flask.request.is_xhr:
92             return flask.jsonify(current=game.current,
93                                 errors=game.errors,
94                                 finished=game.finished)
95         else:
96             return flask.render_template('play.html', game=game)
97
98     # Main
99
100     if __name__ == '__main__':
101         app.run(host='0.0.0.0', debug=True)

```

## static/main.css

```
1  th {
2    text-align: center
3  }
4
5  .vspace {
6    margin-top: 25px;
7  }
8
9  .spaced {
10   letter-spacing: 15px;
11 }
12
13 /* Drawing */
14
15 #drawing {
16   fill: none;
17   stroke: black;
18   stroke-width: 3;
19   stroke-linecap: round;
20 }
21
22 #hangman-drawing > * {
23   display: none
24 }
```

## templates/index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7
8      <title>Hangman game</title>
9
10     <!-- Bootstrap -->
11     <link rel="stylesheet"
12         href="{{ url_for('static', filename='bootstrap.min.css') }}">
13     <link rel="stylesheet"
14         href="{{ url_for('static', filename='main.css') }}">
15   </head>
16   <body>
17     <div class="container text-center">
18       <h1>Hangman game</h1>
19       <div class="row vspace">
20         <div class="col-md-6 col-md-offset-3">
21           {% block content %}{% endblock %}
22         </div>
23       </div>
24     </div>
25
26     <script src="{{ url_for('static', filename='jquery.min.js') }}"></script>
27     <script src="{{ url_for('static', filename='bootstrap.min.js') }}"></script>
28
29     {% block bottom %}{% endblock %}
30   </body>
31 </html>
```

## templates/home.html

```
1 {% extends 'index.html' %}
2
3 {% block content %}
4     <form action="{% url_for('new_game') %}" class="form-inline">
5         <div class="form-group vspace">
6             <input name="player" class="input-lg" required="required"
7                 placeholder="Your name">
8             <button class="btn btn-primary btn-lg" type="submit">Play!</button>
9         </div>
10    </form>
11
12    <div class="panel panel-default vspace">
13        <div class="panel-heading">Top 10</div>
14        <table class="table">
15            <thead>
16                <tr>
17                    <th>#</th>
18                    <th>Player</th>
19                    <th>Word</th>
20                    <th>Errors</th>
21                    <th>Points</th>
22            </thead>
23            <tbody>
24                {% for game in games %}
25                    <tr>
26                        <td>{{ loop.index }}</td>
27                        <td>{{ game.player }}</td>
28                        <td class="text-success">{{ game.current }}</td>
29                        <td class="text-danger spaced">{{ game.errors }}</td>
30                        <td>{{ game.points }}</td>
31                    </tr>
32                {% endfor %}
33            </tbody>
34        </table>
35    </div>
36{% endblock %}
```

## templates/play.html

```
1 {% extends 'index.html' %}
2
3 {% block content %}
4     <svg width="250" height="230" id="drawing">
5         <!-- Structure -->
6         <line y1="220" y2="220" x1="10" x2="120" />
7         <line y1="10" y2="220" x1="65" x2="65" />
8         <line y1="10" y2="10" x1="65" x2="175" />
9         <line y1="10" y2="40" x1="175" x2="175" />
10
11        <!-- Hangman -->
12        <g id="hangman-drawing">
13            <circle cx="175" cy="55" r="15" />
14            <ellipse cx="175" cy="115" rx="20" ry="45" />
15            <line y1="150" y2="190" x1="162" x2="140" />
16            <line y1="150" y2="190" x1="188" x2="210" />
17            <line y1="80" y2="40" x1="162" x2="140" />
18            <line y1="80" y2="40" x1="188" x2="210" />
19        </g>
20    </svg>
```

```

21
22 {% if game.finished %}
23     <h1>
24         {% for letter in game.word %}
25             <span class="
26                 text-{% if letter in game.tries %}success{% else %}danger{% endif %}">
27                 {{ letter }}
28             </span>
29         {% endfor %}
30     </h1>
31     <h3>
32         {% if game.won %} You won! {% else %} You lost :-( {% endif %}
33     </h3>
34     <p>
35         <a class="btn btn-default" href="{{ url_for('home') }}">Home</a>
36         <a class="btn btn-primary"
37             href="{{ url_for('new_game') }}"?player={{ game.player }}">
38             Play again
39         </a>
40     </p>
41 {% else %}
42     <h1 id="current" class="spaced">{{ game.current }}</h1>
43 {% endif %}
44
45 <h2 class="vspace" id="errors">
46     Errors ({{ game.errors|length }}/6):
47     <span class="text-danger spaced">{{ game.errors }}</span>
48 </h2>
49
50 {% if not game.finished %}
51     <form method="post" id="letter-form" class="form-inline vspace">
52         <div class="form-group">
53             <input class="input-lg text-uppercase" maxlength="1" size="1"
54                 name="letter" autocomplete="off" autofocus>
55         </div>
56         <button type="submit" class="btn btn-default btn-lg">Try letter</button>
57     </form>
58 {% endif %}
59 {% endblock %}
60
61 {% block bottom %}
62     <script src="{{ url_for('static', filename='main.js') }}"></script>
63     <script>
64         updateDrawing('{{ game.errors }}');
65     </script>
66 {% endblock %}

```

## static/main.js

```

1  /* Submit letter */
2
3  $('#letter-form').submit(function(e) {
4      var data = $('#letter-form').serialize();
5
6      /* Empty input */
7      $('#letter-form input').val('');
8
9      $.ajax({
10         type: "POST",
11         url: '',
12         data: data,

```

```

13     success: function(data) {
14         /* Refresh if finished */
15         if (data.finished) {
16             location.reload();
17         }
18         else {
19             /* Update current */
20             $('#current').text(data.current);
21
22             /* Update errors */
23             $('#errors').html(
24                 'Errors (' + data.errors.length + '/6): ' +
25                 '<span class="text-danger spaced">' + data.errors + '</span>');
26
27             /* Update drawing */
28             updateDrawing(data.errors);
29         }
30     }
31 });
32 e.preventDefault();
33 });
34
35 function updateDrawing(errors) {
36     $('#hangman-drawing').children().slice(0, errors.length).show();
37 }

```

## README.md

### # Hangman

Simple hangman game implemented with Flask

### # Installation

#### ## Option 1: Ubuntu packages

```
sudo apt-get install python-flask python-flask-sqlalchemy
```

#### ## Option 2: pip

[Install pip](<https://pip.pypa.io/en/stable/installing/>), then:

```
pip install Flask Flask-SQLAlchemy
```

### # Run

```
python hangman.py
```

Create database with:

```
python -c 'from hangman import db; db.create_all()'
```

### # Links

- \* Hangman github repository: <https://github.com/vlopezferrando/hangman>
- \* Slides: <https://slides.com/victorlrf/flask>
- \* Flask: <http://flask.pocoo.org>
- \* Jinja2: <http://jinja.pocoo.org/docs/dev/>
- \* Bootstrap: <http://getbootstrap.com>
- \* JQuery: <https://jquery.com>