

flask

A microframework based on Werkzeug. It's extensively documented and follows best practice patterns.

Functions

<code>after_this_request(f)</code>	Executes a function after this request.
<code>copy_current_request_context(f)</code>	A helper function that decorates a function to retain the current request context.
<code>escape(s) -> markup</code> <code>)</code>	Convert the characters <code>&</code> , <code><</code> , <code>></code> , <code>'</code> , and <code>"</code> in string <code>s</code> to HTML-safe sequences.
<code>flash(message[, category])</code>	Flashes a message to the next request.
<code>get_flashed_messages([with_categories, ...])</code>	Pulls all flashed messages from the session and returns them.
<code>get_template_attribute(template_name, attribute)</code>	Loads a macro (or variable) a template exports.
<code>has_app_context()</code>	Works like <code>has_request_context()</code> but for the application context.
<code>has_request_context()</code>	If you have code that wants to test if a request context is there or not this function can be used.
<code>jsonify(*args, **kwargs)</code>	This function wraps <code>dumps()</code> to add a few enhancements that make life easier.
<code>make_response(*args)</code>	Sometimes it is necessary to set additional headers in a view.
<code>redirect(location[, code, Response])</code>	Returns a response object (a WSGI application) that, if called, redirects the client to the target location.
<code>render_template(template_name_or_list, **context)</code>	Renders a template from the template folder with the given context.
<code>render_template_string(source, **context)</code>	Renders a template from the given template source string with the given context.
<code>safe_join(directory, filename)</code>	Safely join <i>directory</i> and <i>filename</i> .
<code>send_file(filename_or_fp[, mimetype, ...])</code>	Sends the contents of a file to the client.
<code>send_from_directory(directory, filename, ...)</code>	Send a file from a given directory with <code>send_file()</code> .

<u>stream_with_context(generator_or_function)</u>	Request contexts disappear when the response is started on the server.
<u>url_for(endpoint, *values)</u>	Generates a URL to the given endpoint with the method provided.

Classes

<u>Blueprint(name, import_name[, ...])</u>	Represents a blueprint.
<u>Config(root_path[, defaults])</u>	Works exactly like a dict but provides ways to fill it from files or special dictionaries.
<u>Flask(import_name[, static_path, ...])</u>	The flask object implements a WSGI application and acts as the central object.
<u>Markup</u>	Marks a string as being safe for inclusion in HTML/XML output without needing to be escaped.
<u>Request(envIRON[, populate_request, shallow])</u>	The request object used by default in Flask.
<u>Response([response, status, headers, ...])</u>	The response object that is used by default in Flask.
<u>Session</u>	alias of <code>SecureCookieSession</code>