



D04 - Django-Python Training

Things are getting serious.

Vincent Montécot vmonteco@student.42.fr
42 Staff pedago@staff.42.fr

Abstract: Python, Django!

Contents

I	Preamble	2
II	instructions	3
III	Specific Rules of the day	5
IV	Exercise 00	6
V	FY01	7
VI	exercise 02	9
VII	exercise 03	11

Chapter I

Preamble

Here is the list of monsters that you can encounter while exploring a dungeon Earth Fangh:

- all kinds of undead.
- giant spiders.
- orcs and goblins.
- Troll in the underground.
- witches.
- cursed warriors.
- giant rats.
- a bottle of oil.
- toilet paper.
- two sponges.
- ravioli.

Chapter II

Instructions

- Only this page serve as a reference: Do not fi rm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you should assume that the versions of languages and frameworks used are (or later):
 - Python 3
 - HTML5
 - CSS 3
 - Javascript EM6
 - Django 1.9
 - psycpg2 2.6
- Unless otherwise indicated in the subject, the fi les in python each year on **Python one (d01, d02 and d03) must have their end block if `__name__ == '__main__':` in order to insert the entry point in the case of a program, or tests in the case of a module.**
- Unless otherwise indicated in the subject, each year days on **Django will have its own application in the project to make for reasons peda- gogiques.**
- The exercises are precisely ordered from simple to more complex. In no event shall we bear no attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your fi les and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other fi le than those explicitly speci fi ed by the forward drills.
- Unless otherwise specified in the topic you should not include in your rendering:

- The files `__pycache__`.
- Any migration.
- The file created by the command `collectstatic` of `manage.py` (with the way the value of the variable `STATIC_ROOT`).
- The file Python bytecode (files with extension. `.pyc`).
- database files (especially with `sqlite`).
- Any file or folder must or can be created by the normal behavior of the rendering work.

It is recommended that you modify your `.gitignore` in order to avoid accidents.

- When you need to get an accurate output in your programs, it is of course forbidden to a ffi expensive output precalculated instead of performing the exercise correctly.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet /
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that are not otherwise specified in the subject ...
- Please, by Thor and Odin! Re fl échissez dammit!


chapter III

Specific Rules of the day

- The routes on an application must be defined in a file `urls.py` found in the record of this application.
- Any form (derived class `django.forms.Form`) must be in the file `forms.py` of the application to which it is attached.
- Each page has `ffichée` must be properly formatted (presence of a doctype, tags couples `html`, `body`, `head`) Correct management of special characters, not a `ffi weird chage`.
- The server used to this day is the default Django development server included with the utility `manage.py`.
- Only explicitly requested URLs must return an error page. Thus, if only `/ex00` is requested `/ex00foo` should return a 404 error.
- The requested URL must function with and without slash purpose. So if `/ex00` is requested `/ex00` and `/ex00/` must both work.

Chapter IV

Exercise 00

	Exercise: 00
00 Exercise: First static page.	
Render Folder: <i>ex 00</i> / Files to make: requirements.txt, files and folders of your project and your application	
Permitted functions: All the features of Django.	
Remarks: n / A	

You go into this exercise make your first static page with Django. Create virtualenv with python3, install Django create a file requirements.txt containing project dependencies (which is installed in the virtualenv with pip) at the root of the rendering.

Start a project d04.


Start an application ex00.

Make a page that will be available at the URL / ex00 your site. Gather in this page for information on all of the syntax markdown, and give the title to this page " Ex00: Markdown Cheatsheet. ".

The template used should be named index.html.

Chapter V

Exercise 01

	Exercise: 01
Exercise 01: A few more pages.	
Render Folder: ex01 / files to make: requirements.txt, files and folders on your project and your application.	
Permitted functions: All the features of Django.	
Remarks: n / A	

Perform the following pages in a second application ex01:

- **Title:** "Ex01: Django web framework."

URL: / ex01 / django.

Description: Make this page a brief presentation of Django and his historical.

- **Title:** "Ex01: Process a ffi chage a static page."

URL: / ex01 / Display

Description: Describe this page in the process leading to a ffi a chage static web page from a simple template through a view of the request to the response.

- **Title:** "Ex01: Template engine."

URL: / ex01 / templates

Description: Describe this page in the engine operating tem- default flat Django and operation:

- Blocks.
- Curls for ... in.
- Control structures yew.

◦ From a few variables change past by context. The principle of *do not repeat yourself* an integral part of the philosophy of Django. To observe this principle, realize the requested pages using a basic template named `base.html`.

the template `base.html` must include:

- A block `body` in the body.
- A block `style` in the head.
- A block `title` in the head.

Realize also a template `nav.html` containing a navigation bar that lists links to each page of the exercise.

All pages of this exercise should be based on the template `base.html`. The template `nav.html` must be included in all pages.

Also create two files `style1.css` and `style2.css`. The first will define the text color as the *blue*, the other will define as the *red*.


You will need `style1.css` on every page except the "template engine" where you will use `style2.css`.



You should use each one style sheet and only time in all of your templates for this exercise.

Chapter VI

Exercise 02

	Exercise 02 Exercise 02: First
form.	
Render Folder: <i>ex 02</i> / Files to make: requirements.txt, files and folder of your project and your application.	
Permitted functions: All the features of Django.	
Remarks: n / A	

Make a page accessible through the URL / EX02 in a new application EX02.

This page will contain two parts:

- A form with a text field and a button Submit.
- One part to contain the history of inputs. The rule concerning the

form:


- You must not NOT encode the form field hard, but use the class `django.forms.Form` that provides Django to create a form. You pass the ground as context the rendering function of the template.

Here are the rules for the history:

- The history entries will initially be empty.
- Each text by submitting the form you will need:
 - Register input and its time stamp at the end of a file of logs. If this file does not exist it is automatically created. This file must be created in the application folder hand.
 - Add the entry to the history of the page, with the time and date of submission.
- The path to the file logs (name of the file included) must be defined in the file settings.py of the project. Thus, each entry submitted will be present in the historical part of the page and in the file log, preceded by its time stamp.
- Data must also be persistent. If the development server must be restarted for some reason, the data should not be per- due.

Chapter VII

Exercise 03

	Exercise 03 Exercise 03: Fifty
Shades of bic.	
Render Folder: <i>ex 03</i> / Files to make: requirements.txt, files and folders on your project and your application.	
Permitted functions: All the features of Django.	
Remarks: n / A	

Create a last application EX03.

A ffi to a page containing an array of 4 columns and 51 lines (one line to the column names).

This page will be accessible at the URL / EX03.

Each column of the table correspond to a color among *black, red, blue, green*.

The cells of the table must have the following characteristics:

Height: 40 pixels.

Width : 80 pixels.

Background color : a shade of color that matches the column.

The table is a ffi expensive shades of each color to obtain a gradient of 50 lines.

You must observe the following:

- Your template must NOT contain color hard. The different shades must be generated in one view and all be different.
- 4 pairs of tags `<td>` `</td>` are allowed in your template.
- 4 pairs of tags `<th>` `</th>` are allowed in your template.
- Your table should be formatted like this:
 - A couple of tags `<th>` `</th>` by box to the column names.
 - A couple of tags `<tr>` `</tr>` color shade line.
 - A couple of tags `<td>` `</td>` color swatch box.