Universidad Tecnológica de Tijuana

TSU. en Tecnologías de la Información

en Entornos Virtuales y Negocios Digitales

Made By:

“Humaran Beltran Saul Alejandro“

Teacher:

“Ing. Ray Brunett Parra Galaviz”

Assigment:

“Web Aplications“

Actividad:

“Template Django System“

Cuatrimestre: Third ***Grupo:*** 3-F

Date

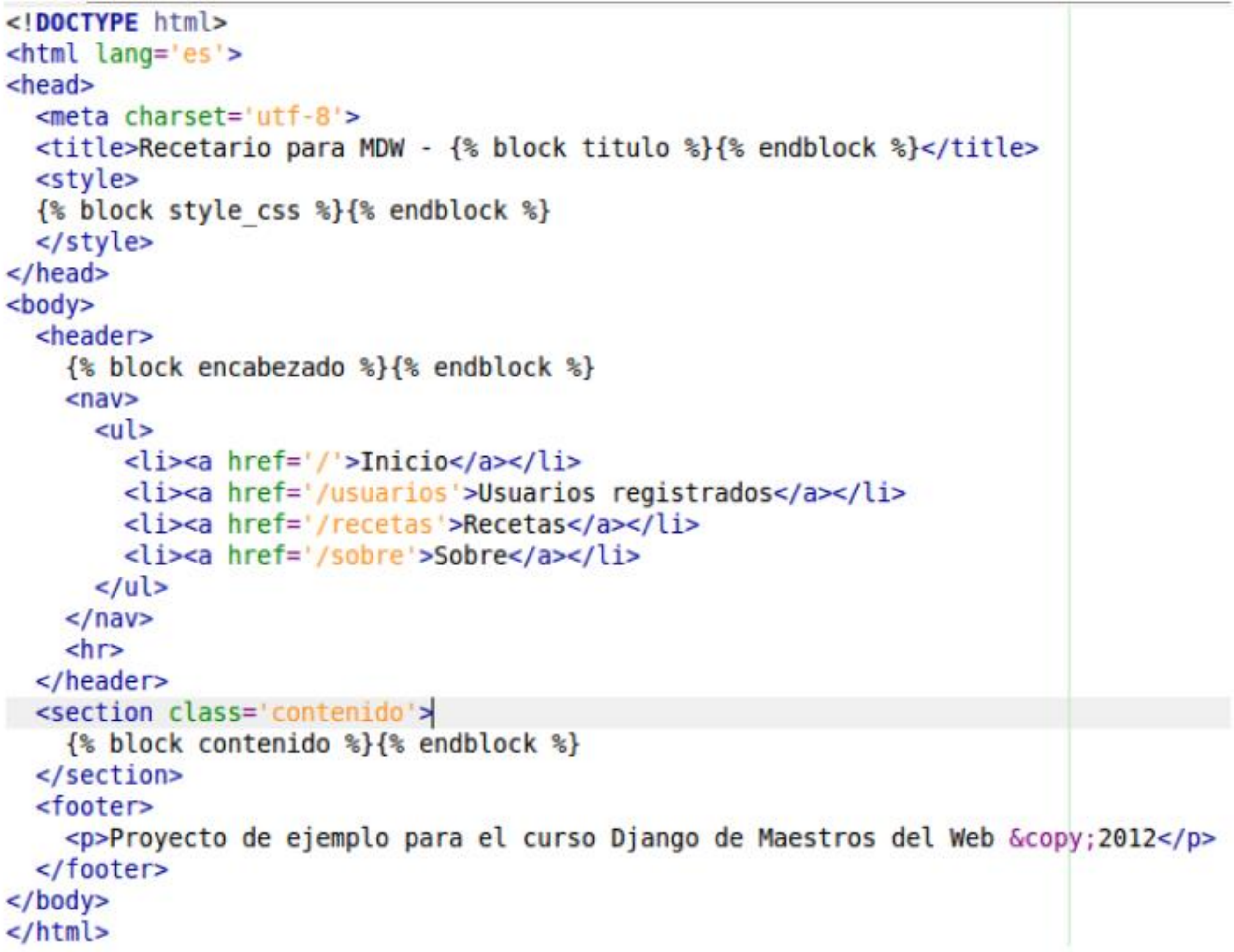
“24/10/2023“

Introduction:

The Django templating system has a part that is used as a framework and allows users to Developers create web pages more efficiently and with a more consistent design.
We will briefly explain what the template system used is about.

Concepts:

• **Templates:** Templates in Django are HTML files with bookmarks and tags specials that allow the insertion of dynamic data. Templates separate logic presentation of application logic, making it easy to maintain and update.
code organization.

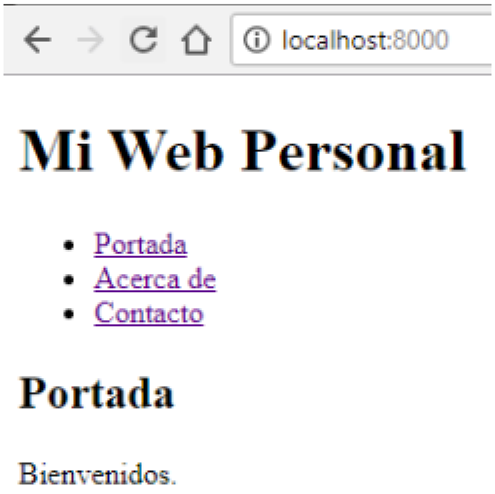


```
<!DOCTYPE html>
<html lang='es'>
<head>
  <meta charset='utf-8'>
  <title>Recetario para MDW - {% block titulo %}{% endblock %}</title>
  <style>
  {% block style_css %}{% endblock %}
  </style>
</head>
<body>
  <header>
    {% block encabezado %}{% endblock %}
    <nav>
      <ul>
        <li><a href='/'>Inicio</a></li>
        <li><a href='/usuarios'>Usuarios registrados</a></li>
        <li><a href='/recetas'>Recetas</a></li>
        <li><a href='/sobre'>Sobre</a></li>
      </ul>
    </nav>
    <hr>
  </header>
  <section class='contenido'>
    {% block contenido %}{% endblock %}
  </section>
  <footer>
    <p>Proyecto de ejemplo para el curso Django de Maestros del Web &copy;2012</p>
  </footer>
</body>
</html>
```

• **Django templating system:** Django provides a built-in templating system which includes a templating engine and special syntax for embedding dynamic data on web pages. This template system is responsible for processing the templates and generate the final content that is shown to the user.

• **Template syntax:** Django templates use a special syntax to embed dynamic data and perform logical operations, such as loops and conditionals. Some examples of template tags include
- {% for %}
- {% if %}
-{{variable}}

• **Template inheritance:** Django allows the creation of base templates that can be inherited by other templates. This makes it easy to create a design consisting of a website and reusing common elements such as headers and footers.



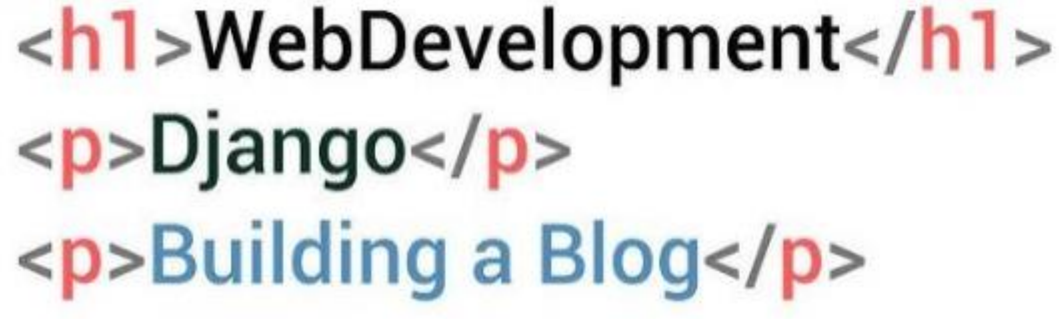
Mi Web Personal

- Portada
- Acerca de
- Contacto

Portada

Bienvenidos.

• **Context:** The context is a set of data that is passed to a template and that contains the information necessary to generate the page. Data in context can come from the database, application logic, or other sources.



```
<h1>WebDevelopment</h1>
<p>Django</p>
<p>Building a Blog</p>
```

• **Data filtering:** Django provides filters that allow you to modify and format data before displaying them in the template. For example, it is possible to apply filters to format dates, numbers or text strings.

• **Template inclusion:** Django templates can include other templates, which facilitates modularity and code reuse.

• **Template Tags:** Django provides a set of template tags that
They allow logical operations to be performed within templates. Some common labels include
- {% for %} for loops,
- {% if %} for conditionals
- {% block %} to define blocks that can be replaced in child templates.

• **Filters**: Template filters are functions that are applied to variables within the templates to modify or format data. Django includes a variety of filters built-ins, such as date to format dates and length to obtain the length of a list.

• Nested Templates: You can nest templates inside other templates. This means that you can divide the presentation of a web page into reusable fragments and combine them to build more complex pages.

• **Template Upload:** You can organize your templates in directories and upload them easily.
efficient in your views using the directive
- {% extends %}.

• **Python integration:** Although the templates are designed to separate the logic
presentation of the application logic, Django provides a way to execute
Python code in templates via tags like
- {% if %} and {% for %}.

• **Security:** Django includes security measures in its template system to
protect against common attacks, such as script injection (XSS).

Conclusion:

Django's templating system is highly versatile and is one of the reasons why
Django is widely used in web development. Provides a clean focus
and organized for the generation of dynamic content in your web applications, which
makes it easy to create attractive and functional websites