1. **Module Introduction :**
   Static Files → The files that won't change, which is passed from the main project. Ex image, video.
   Media Files → The files that will change **Dynamically** from backend side. E.x Profile Pic

2. **Working with Static Files inside Project :**
   Static Files → Unchangeable Files
3. **Working with Static Files inside Application :**
   We must configure in settings.py for project static,
   but it's not necessary for application static, without configuring static in project/settings.py there will be error shown

4. **Static Files Vs Media Files Configuration :**
   Dynamic Files → Media Files (From Backend Side)

5. **Adding Bootstrap to Django Project :**
6. **Using Url tag in Django Project :**

7. **Introduction to Template Inheritance : DRY Principle**
   DRY → Don't Repeat Yourself

Template Inheritance/ Template Extending

Template inheritance allows us to build a base "skeleton" template that contains all the common elements of our site and defines blocks that child templates can override.

To inherit template, use the extends tag.

The base.html block tags will be recognized by the template engine, which will then replace those blocks with the contents of the child template.

As many levels of inheritance as necessary may be used.

extends Tag

Syntax:-

{% extends 'parent_template_name' %}

{% extends variable %}

Example:-

{% extends "./base1.html" %}

{% extends "../base2.html" %}

{% extends "./my/base3.html" %}

{% extends somthing %}

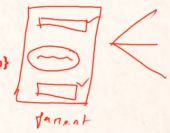
block Tag

{% block %} – The block tag is used to for overriding specific parts of a template.

Syntax:-

{% block blockname %}....{% endblock %}

{% block blockname %}....{% endblock blockname %}



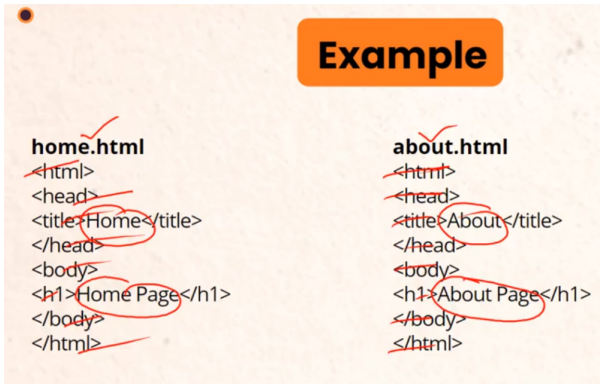
Example:-

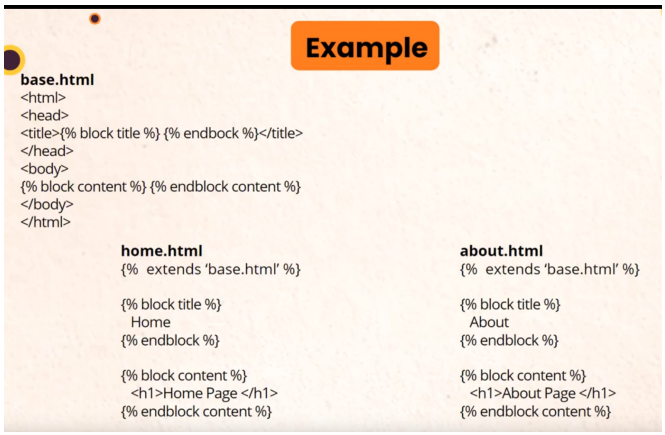
{% block title %} ..... {% endblock %}

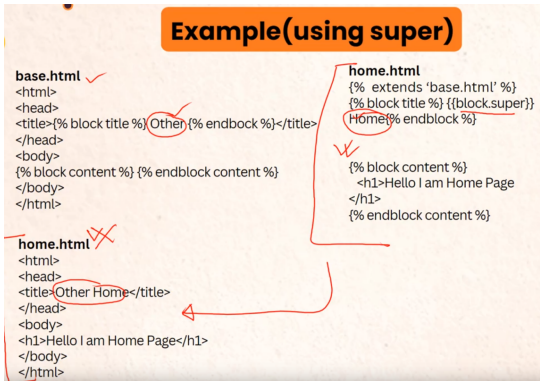
{% block content %}..... {% endblock content %}

Rules

- {% extends %} must be the first template tag in that template. Template inheritance won't work, otherwise.
- More {% block %} tags in our base templates are better.
- We Can't define multiple block tags with the same name in the same template.
- If We need to get the content of the block from the parent template, the {{ block.super }} variable will do the trick.







8. **Template Inheritance : Code**
9. **Creating Custom Template and Template Tags :**

<u>Custom Template Filter</u> :

i) write def funcion in the tagfile
ii) pass the (arg, value)
iii) register.filter("assign_name", function_name)
iv) now load tagfile and call where you want to use **that filter.**

Custom Template Tags :

a. Folder convert into Module (default name “templatetags”)
b. __init__.py
c. Import
d. tagfile.py
e. write the function return by dictionary
f. variable = where to show by calling get_template
g. make a html file
h. register.inclusion_tag(variable)(the function)
i. load tagfile in that html file
j. work what you want to do by DTL and use the values where you need
k. now {% load tagfile %} and use which html page you want just by calling the {% function name %}.

—----------END—----------