1.下载Django

Pip install django ==2.4

2.创建Django项目

Python –m django startproject 项目名

3.执行迁移

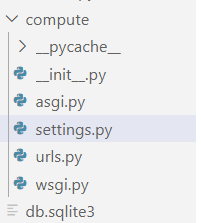
python manage.py migrate

4.创建app

python manage.py startapp App名

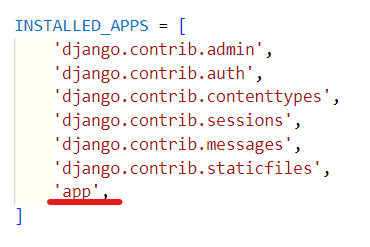
5.修改参数

在该文件中进行：

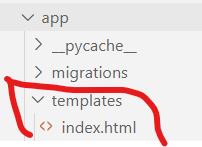


修改值如下：

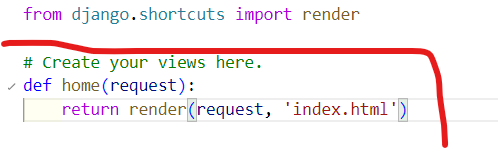




6.创建templates



7.编辑views.py文件



代码：

def home(request):

    return render(request, 'index.html')

8.编辑urls.py文件



代码：

from django.contrib import admin

from django.urls import path

from app.views import home

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', home, name='home'),

]

9.启动服务

python manage.py runserver

10.计算器项目代码：

Index.html:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Online Computer</title>

    {% load static %}

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <link rel="stylesheet" href="{% static 'css/style.css' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

    <div class="container-fluid">

    <div class="row">

    <div class="col-xs-1 col-sm-4"></div>

    <div id="Computer" class="col-xs-10 col-sm-6">

        <input type="text" id="txt\_code" name="txt\_code" value="" class="form-control input\_show" placeholder="computing process" disabled />

        <input type="text" id="txt\_result" name="txt\_result" value="" class="form-control input\_show" placeholder="result" disabled />

        </br>

    <div>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_7()">7</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_8()">8</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_9()">9</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_div()">/</button>

        </br>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_4()">4</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_5()">5</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_6()">6</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_mul()">\*</button>

        </br>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_1()">1</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_2()">2</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_3()">3</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_sub()">-</button>

        </br>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_0()">0</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_00()">00</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_dot()">.</button>

        <button type="button" class="btn btn-default btn\_num" onclick="fun\_add()">+</button>

    </div>

    <div>

    </br>

    <button type="button" class="btn btn-success btn-lg btn\_clear" id="lgbut\_clear" onclick="fun\_clear()">

        Clear

    </button>

    <button type="button" class="btn btn-primary btn-lg" id="lgbut\_compute">

        Compute

    </button>

    </div>

    <div class="col-xs-1 col-sm-2"></div>

    </div>

</div>

</div>

<div class="extendContent"></div>

</body>

<script>

var x=document.getElementById("txt\_code");

var y=document.getElementById("txt\_result");

function fun\_7(){

    x.value +='7';

}

function fun\_8(){

    x.value +='8';

}

function fun\_9(){

    x.value +='9';

}

function fun\_4(){

    x.value +='4';

}

function fun\_5(){

    x.value +='5';

}

function fun\_6(){

    x.value +='6';

}

function fun\_1(){

    x.value +='1';

}

function fun\_2(){

    x.value +='2';

}

function fun\_3(){

    x.value +='3';

}

function fun\_0(){

    x.value +='0';

}

function fun\_00(){

    x.value +='00';

}

function fun\_div(){

    x.value +='/';

}

function fun\_sub(){

    x.value +='-';

}

function fun\_add(){

    x.value +='+';

}

function fun\_mul(){

    x.value +='\*';

}

function fun\_dot(){

    x.value +='.';

}

function fun\_clear(){

    x.value ='';

    y.value='';

}

</script>

<script>

    function showResult(data){

        var y=document.getElementById('txt\_result');

        y.value=data['result'];

    }

</script>

<script>

    $('#lgbut\_compute').click(function(){

        $.ajax({

            url:'/compute/',

            type:'POST',

            data:{

                'code':$('#txt\_code').val(),

                'csrfmiddlewaretoken': '{{ csrf\_token }}'

            },

            dataType:'json',

            success:showResult

        })

    })

</script>

</html>

Views.py:

from django.shortcuts import render

# Create your views here.

def home(request):

    return render(request, 'index.html')

import subprocess

from django.views.decorators.http import require\_POST

from django.http import JsonResponse

from django.views.decorators.csrf import csrf\_exempt

def run\_code(code):

    try:

        code='print('+ code +')'

        output=subprocess.check\_output(['python','-c',code],universal\_newlines=True,stderr=subprocess.STDOUT,timeout=30)

    except subprocess.CalledProcessError as e:

        output='input error!'

    return output

@csrf\_exempt

@require\_POST

def compute(request):

    code=request.POST.get('code')

    result=run\_code(code)

    return JsonResponse(data={'result':result})

urls.py：

"""compute URL Configuration

The urlpatterns list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/4.1/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path

from app.views import home

from app.views import compute

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', home, name='home'),

    path('compute/', compute, name='compute')

]