

Erronka1. Javascript-II

Web garapena bezeroen ingurunean

- 4. Jquery
- 5. Ajax

4. JQuery

Zer da jQuery?

- JQuery Javascripteko liburutegi bat da, arina, azkarra eta ezaugarri asko dituen.
- HTML fitxategiak, gertaerak, animazioak, Ajax eta beste gauza asko kudeatzea modu errazean egiteko aukera ematen du
- API oso erabilgarria da
- Nabigatzaile gehienak ulertzen dute.

Gure web orrietan erabiltzeko, <script> etiketa batekin sartu beharko dugu gure kodigoan:

<https://code.jquery.com>

Adibidez:

```
<body>
```

```
....
```

```
....
```

```
<script src="https://code.jquery.com/jquery-3.6.1.min.js" integrity="sha256-  
o88AwQnZB+VDvE9tvIXrMQaPIFFSUTR+nldQm1LuPXQ=" crossorigin="anonymous">
```

```
</script>
```

```
</body>
```

4. JQuery

Oinarrizko kontzeptu batzuk:

<https://www.arkaitzgarro.com/jquery/capitulo-3.html>

- Edozein gauza egiten hasi aurretik, orria kargatuta dagoela bermatzea oso garrantzitsua da:

```
$(document).ready(function() {  
    ..... });
```

Edo:

```
$(document).ready(prest)
```

```
function prest() {  
    ..... });
```



Fijatu: prest funtzioa parentesi barik
Jarri behar dela

4. JQuery

DOM barruko osagaia ezberdinak aukeratzeko adibide batzuk:

ID erabilita aukeratu osagai bat:

```
$('#myId'); // id-a bakarra izan behar da
```

Klasearen izena erailita aukeratu:

```
$('.myClass');  
$('#div.myClass'); // aurretik osagai mota jartzen bada, arinagoa da.
```

Atributo erabilita aukeratu:

```
$('#input[name=first_name]'); // honela astiro joan daiteke
```

CSS estiloen bezala aukeratu:

```
$('#contents ul.people li');
```

Sasi-hautatzailea erabilita aukeratu:

```
$('#a.external:first'); // <a> motako eta klase "external" duen lehenengoa  
$('#tr:odd'); // taula bateko "tr" bakoitiak  
$('#myForm :input'); // myForm formularioko "input" guztiak  
$('#div:visible'); // ikusgai dauden "div" guztiak  
$('#div:gt(2)'); // "div" guztiak lehenengo hirurak ezezik  
$('#div:animated'); // animatuta dauden "div" guztiak
```

4. JQuery

Funtzio batzuk

Kapak erakusten eta izkututzen:

hide(), show(), toggle()

fadeIn(), fadeOut(), fadeToggle()

slideDown(), slideUp(), slideToggle()

```
<body>
<p id="parrafoa">If you click on the "Hide" button, I will disappear.</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#hide").click(function(){
        $("p").hide();
    });
    $("#show").click(function(){
        $("#parrafoa").show();
    });
});
</script>
</body>
```

4. JQuery

Html eta testua aldatzen:
text(), html(), val(), attr()

```
<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<p>Name: <input type="text" id="box" value="Mickey Mouse"></p>
<p><a href="https://www.w3schools.com" id="w3s">W3Schools.com</a></p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
<button id="btn3">Show Value</button>
<button id="btn4">Show href Value</button>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
    $("#btn3").click(function(){
        alert("Value: " + $("#box").val());
    });
    $("#btn4").click(function(){
        alert($("#w3s").attr("href"));
    });
});
</script>
</body>
```

Adibideak:

elemento = \$("input:text").val()

**\$("#input:button").click(function()
{.....})**

Formularioaren berezko joera ekiditzeko:

```
$("#nireformu").on("submit", function(e)
{
    e.preventDefault();
})
```

4. JQuery

Formulariokin lan egiterakoan:

Name atributo erabiliz (getElementsbyName ordezkatzeko)

```
<input type="radio" name="sexua" value="Gizonezkoa" checked> Gizonezkoa  
<input type="radio" name="sexua" value="Emakumezkoa"> Emakumezkoa
```

```
var radios = $("[name='sexua']");  
for (i = 0, i < radios.length; i++)  
{  
    .....  
    radios[i].checked  
}
```

getElementsbyTagName ordezkatzeko:

```
$("#div").each(function() {  
    .....  
})  
  
var paragrafoak = $("p");  
for (i = 0; i < paragrafoak.length; i++)  
{  
    .....  
    paragrafoak[i]  
}
```

4. JQuery

Formulariokin lan egiterakoan:

Checkbox eta Radio osagaiak “checked” dauden jakiteko:

```
If ($("#osagaia1").is(":checked"))  
{  
.....  
}
```

Select baten aukeragaitik galdetzeko:

```
if ($("#herria option:selected").text() == "Aukeratu")  
{  
.....  
}
```

Edo:

```
if ($("#herria option:selected").val() == 0  
{  
.....  
}
```


4. JQuery

Bootstrap bidez formularioak balidatzeko:

1. HTML formularioen balidazioa CSS bidez egiten da: :invalid eta :valid erabiliz <input>, <select>, eta <textarea> osagaietan.
1. Formularioan .needs-validation eta .was-validated klasea erabiltzen da.
2. Javascriptegaz balidatzeko metodoak: [constraint validation API](#)

Informazio osoa, esteka honetan:

<https://getbootstrap.com/docs/4.5/components/forms/#validation>

4. JQuery

Adibidez formulario hau:

```
<form class="needs-validation" novalidate>
  <div class="form-row">
    <div class="col-md-6 mb-3">
      <label for="validationCustom01">First name</label>
      <input type="text" class="form-control" id="validationCustom01" value="Mark" required>
      <div class="valid-feedback">
        Looks good!
      </div>
    </div>
    <div class="col-md-6 mb-3">
      <label for="validationCustom02">Last name</label>
      <input type="text" class="form-control" id="validationCustom02" value="Otto" required>
      <div class="valid-feedback">
        Looks good!
      </div>
    </div>
  </div>
  <button class="btn btn-primary" type="submit">Submit form</button>
</form>
```

4. JQuery

Eta script hau:

```
<script>

$(document).ready(function(){
    $('.needs-validation').on('submit', function(e) {
        if (!this.checkValidity()) {
            e.preventDefault();
            e.stopPropagation();
        }
        $(this).addClass('was-validated');
    });
});

</script>
```

4. JQuery

**Bukleak egiten, css aldatzen:
each, this, css, append**

```
<body>
<div id="bat">
<p>Primer párrafo</p>
<p>Otro párrafo</p>
<p>Tecer párrafo</p>
<p>Uno más</p>
<p>y se acabo</p>
</div>
<input type="submit" value="Nuevo">

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>
<script>
    $(document).ready(function(){
        $("p").each(function(i){
            if(i%2==0){
                $(this).css("background-color", "#eee");
            }else{
                $(this).css("background-color", "#ccc");
            }
        });
        $("input:submit").click(function() {
            $("#bat").append("<p>Otro mas</p>");
        });
    });
</script>
</body>
```

Adibideak:

`$("#ul").append(" ")`

```
<body>
<div>red</div>
<div>blue</div>
<div>red</div>
<div>white</div>
<div>red</div>
<div>green</div>
<div>orange</div>
<div>red</div>
<div>nada</div>
<div>red</div>
<div>blue</div>
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></
script>
<script>
    $(document).ready(function(){
        $("div").each(function(i){
            elemento = $(this);
            if(elemento.html() == "white")
                return true;
            if(elemento.html() == "nada")
                return false;
            elemento.css("color", elemento.html());
        });
    });
</script>
</body>
```

4. JQuery

33. Formulario bat egin saskira erosketak gehitzeko eta erosketa egiteko: (each, append, css, val) (bootstrap-eko balidazioa erabili zerbait idazten dela bermatzeko)

Saskia egiteko aplikazioa. Idatzi zure erosketak:

Saskira gehitzekoa:

gehitu botoian klik egiterakoan

Erosketa egin botoian klik egiterakoan

Saskia egiteko aplikazioa. Idatzi zure erosketak:

Saskira gehitzekoa:

- esnea
- ogia

Saskia egiteko aplikazioa. Idatzi zure erosketak:

Saskira gehitzekoa:

Esta página dice:
Zure erosketa listo dago

- esnea
- patata
- arraultzak
- ogia

4. JQuery

Efektoak sortzen: animate

```
<span style="outline:1px solid black" id="capa0">Capa donde cambiaremos la propiedad outlineWidth</span><br>
<input type="button" id="boton0" value="Cambiar a 15px el grosor del borde exterior del elemento">
<br>
<br>
<span style="border:1px solid black" id="capa1">Capa donde cambiaremos la propiedad borderWidth</span><br>
<input type="button" id="boton1" value="Cambiar a 10px el grosor del borde del elemento en 3 segundos">
<br>
<br>
<span style="border:1px solid black" id="capa2">Capa donde cambiaremos la propiedad borderBottomWidth</span><br>
<input type="button" id="boton2" value="Cambiar a 20px el grosor del borde inferior del elemento en 5 segundos y callback">
<br>
<br>
<span style="border:1px solid black" id="capa3">Capa donde cambiaremos la propiedad borderTopWidth</span><br>
<input type="button" id="boton3" value="Cambiar a 15px el grosor del borde superior del elemento y callback">
<br>
<br>
<span style="border:1px solid black" id="capa4">Capa donde cambiaremos la propiedad borderLeftWidth y borderRightWidth</span><br>
<input type="button" id="boton4" value="Cambiar a 10px el grosor del borde izquierdo y a 20px el derecho">
```

```
$(document).ready(function(){
$("#boton0").click(function(){
    $("#capa0").animate({outlineWidth:"15px"});
});
$("#boton1").click(function(){
    $("#capa1").animate({borderWidth:"10px"},3000);
});
$("#boton2").click(function(){
    $("#capa2").animate({borderBottomWidth:"20px"},5000,function() {alert ('Efecto terminado!');});
});
$("#boton3").click(function(){
    $("#capa3").animate({borderTopWidth:"15px"},function() {alert ('Efecto terminado!');});
});
$("#boton4").click(function(){
    $("#capa4").animate({borderLeftWidth:"10px", borderRightWidth:"20px"});
});
});
```

4. JQuery

Gertaera bat harrapatzen denean, funtzio batean tratatu daiteke, (“e” aldagaia erabiliz event laburtzeko edo beste izen bat).

Harrapatu dugun gertaera tekla bat zapaltzea bada, tekla hori zein izango dan jakin daiteke. Adibidez, geziak: “ArrowRight, ArrowLeft, ArrowUp, ArrowDown”

```
$(document).ready(function() {  
  
$(document).on('keydown',function(e) {  
    switch(e.key) {  
        // left arrow pressed  
        case "ArrowLeft":  
            .....  
            break;  
        case "ArrowRight":  
            .....  
    }  
})  
}
```

4. JQuery

34. Animazioa duen ariketa bat egin (animate erabiliz). Espazioko kohete bat 10 pixel mugitzen da gezi teklak erabiltzen ditugunean, ezkerrera, eskuinera, gora eta behera.
adibidez: `.animate({top:"+=10px"}, 1000)`

api.jquery.com orriaren adibideak ikusi: <https://api.jquery.com/animate/#animate-properties-duration-easing-complete>

Irudi bi behar dira, atzealdean dagoen "spacestars.jpg" eta kohetea bera "rocket.png". Hasieran, irudia erdian eta goian dago (position:relative; left:50%;top:0;)

Animatu: Sakatu gezia eta mugitu egingo da!



Diagonalean mugitzeko prestatu. Bi mugimendu izan daitezke, bat, gora eta ezkerretara, eta bestea bera eta eskuinetara.

Botoi bat prestatu hasierako lekura bueltatzeko.

<https://codepen.io/seeker5084/pen/VMQGwX>

<https://thoughtbot.com/blog/transitions-and-transforms> WAG 2022-2023 Ikasturtea

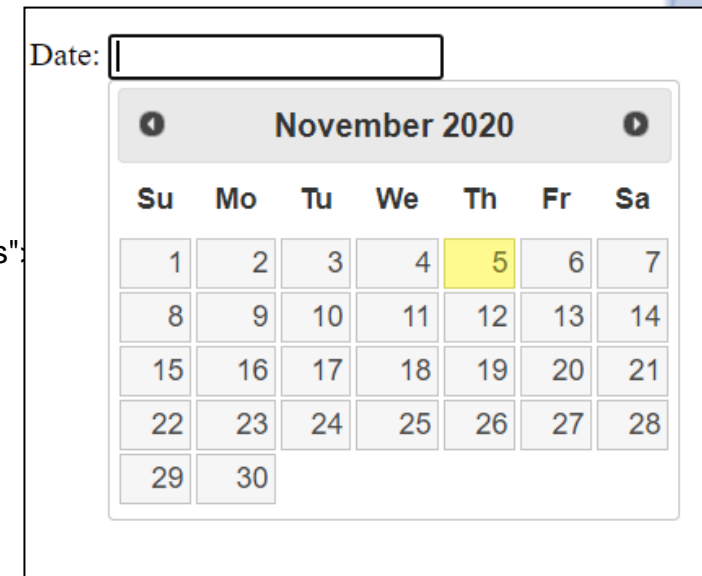
4. Jquery

Eta beste adibide bat ikusiko dugu: datak aukeratzeko osagai bat (datepicker)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Datepicker - Default functionality</title>
  <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
</head>
<body>

<p>Date: <input type="text" id="datepicker"></p>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
  src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"
  integrity="sha256-VazP97ZCwtekAsvgPBSUwPFKdrwD3unUfSGVYrahUqU="
  crossorigin="anonymous"></script>
<script>
  $( function() {
    $( "#datepicker" ).datepicker();
  } );
</script>
</body>
</html>
```



<https://code.jquery.com/ui/>

4. JQuery

Eta beste adibide bat openstreetmap erabiltzen (google maps api-a ez da dohainik):

<https://leafletjs.com/> (hemen hasierako adibide bat: <https://leafletjs.com/examples/quick-start/>)

MapBox access token bat erabiltzen du. Interesantea da bakoitzak berena sortzea, baina aprobak egiteko erabil daiteke:

accessToken:

'pk.eyJ1IjoibWFwYm94IiwiaWUiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriSLbB6B5aw'

Routing-a egiteko informazioa:

<https://www.liedman.net/leaflet-routing-machine/>

<https://stackoverflow.com/questions/60542919/calculate-time-between-two-points-in-leaflet-routing-machine-depending-on-transp>

****Hau probatu:**

<https://gis.stackexchange.com/questions/193235/leaflet-routing-machine-how-to-dynamically-change-router-settings/193251#193251>

5. Ajax



Zer da Ajax?

AJAX: **A** sinkrono **J**avaScript **A**nd **X**ML

Web orrialdeak asinkronoki eguneratzeko aukera ematen duen teknika da, eta horrek esan nahi du nabigatzaileak ez duela orrialde osoa berriro kargatu behar orriaren zati bat aldatzeko. AJAXek zerbitzariari eta zerbitzaritik eguneratutako informazioa soilik kargatzen arduratzen da

Aurretik, botoi bat sakatzen genuenean, mezua zerbitzarira bidaliko zen eta erantzuna itzuliko zen. Ezin genuen orrialdeko beste elementu batzuekin elkarreragin, erantzuna jasotzen zen arte eta orrialde hori eguneratzen zen arte.

Hori atzerapen bat zen eta web bisitariaren esperientzia negatiboki eragin zezakeen. Horregatik, hori ekiditeko, AJAX asmatu zen.

Hasiera baten XML formatua erabiltzen zen soilik, baina beranduago JSON formatua erabiltzeko prestatu zen ere.

5. Ajax

Ajax bidez **http eskaerak** egiteko hainbat modu daude, baina nagusiak **XMLHttpRequest** eta **fetch** dira (nabigatzailean lehenetsita daudenak).

Fetch barriagoa da baina guk hurrengo erronketan erabiliko dugu teknika hori, beraz, XMLHttpRequest erabiliko dugu. Baina ez dugu idatziko zuzenean, **JQUERY**-k ematen digun aukerak erabiliko ditugu.

Nola funtzionatzen du AJAX-ek?

- Gertaera (Ebentoa) bat dago orri batean, adibidez, hasierako orria kargatzen dela, formulario bat bidaltzen dela, esteka edo botoi batean klik egiten dela, etab.
- XMLHttpRequest objektu bat sortzen da eta eskaria zerbitzariari bidaltzen dio.
- Zerbitzariak eskariari erantzun egiten dio.
- Erantzuna harrapatu egiten da eta zerbitzariak beste erantzun bat bueltatzen du eskatu diren datuekin.
- Posibilitate ezberdin asko daude GET eta POST eskaerak erabili nahi izateko modu asinkronoan, datuak atzera eta aurrera mugitzeko zerbitzarira eta zerbitzaritik. Gainera, teknika honek, web orriak dinamikoagoak eta arinago kargatzekoak sortzea ahalbidetzen du.

5. Ajax

Ajax exekutatzeko zerbitzaria behar da. Editore ezberdinak aukera ematen dute hau egiteko:

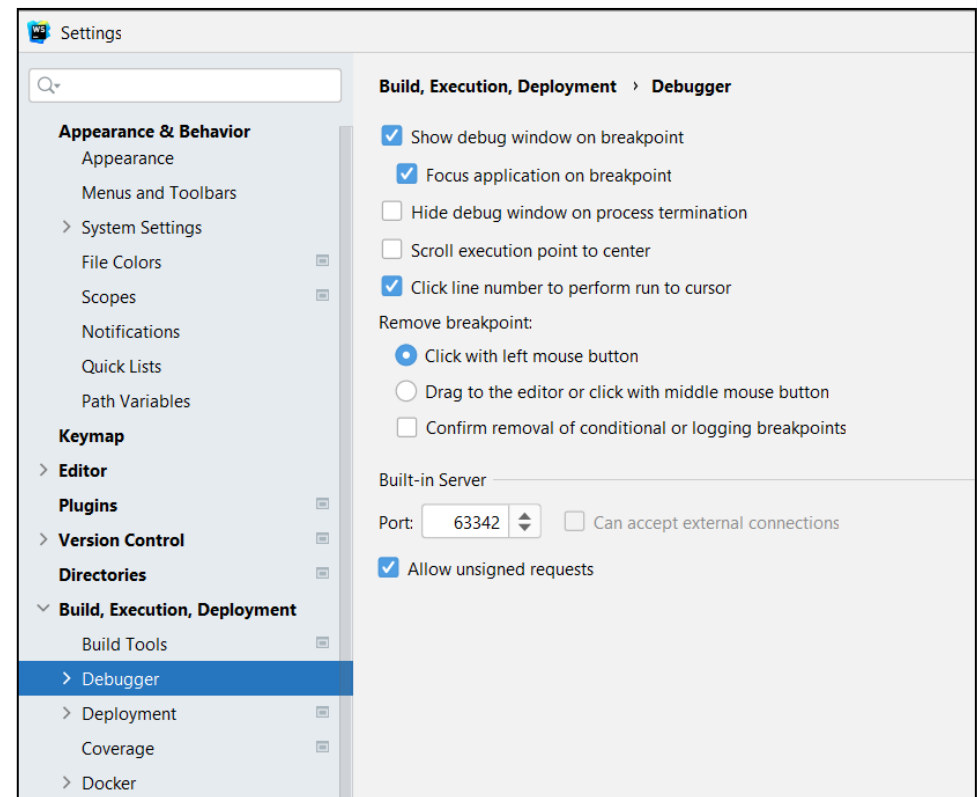
- Visual Studio Code

Live Server luzapena instalatu – «**Open with Live Server**»

- WebStorm

Settings

- Build, Execution, Deployment
- Debugger



5. Ajax – adibide sinple bat

index.html

```
<html>
<head>
  <title>Ajax Simple</title>
</head>
<body>

<a href="#" id="esteka">Klik egin!</a>
<br>
<div id="kapa"></div>

<script src="jquery-1.3.2.min.js" type="text/javascript"></script>

<script src="scripta.js" type="text/javascript"></script>
</body>
</html>
```

scripta.js

```
$(document).ready(function(){
  $("#esteka").click(function(evento){
    evento.preventDefault();
    $("#kapa").load("edukia-ajax.html");
  });
});
```

edukia-ajax.html (artxibo hau karpeta berdinean)

```
<p>
  Hau ajax eta jquery proba bat da
</p>
```

5. Ajax- adibidea json-a irakurtzen

index.html

```
<html>
<head>
  <title>Ajax Simple</title>

  <style>
    #ongietorri {
      display: none;
    }
  </style>
</head>
<body>

<button id="ajaxButton">jQuery .ajax method</button>
<h3 id="ongietorri">

Hola <span id="izena"></span>. Zure pasahitza hau da: <span
id="pasahitza"></span>

</h3>
<script src="jquery-1.3.2.min.js" type="text/javascript"></script>

<script src= "scripta.js" type="text/javascript"></script>
</body>
</html>
```

scripta.js

```
$("#ajaxButton").click(function () {
  /* estilo honetako url-a erabiliko zen
  /*var url = 'http://localhost:8888/login';*/

  var url = 'edukia-ajax.json';

  $.ajax({
    type:"GET",
    url:url,
    success: function (data) {
      $("#izena").html(data.user_name);
      $("#pasahitza").html(data.password);
      $("#ongietorri").css("display","block");
    }
  });
});
edukia-ajax.json (artxibo hau karpeta
berdinean)

{"user_name":"jerry",
 "password":"888888"}
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Django proiektu bat sortuko dugu probak egiteko.

Lehenengo ingurune birtuala sortuko dugu: **Django_Birtuala**

Python instalatuta daukagula baieztatu:

```
$ pip3 --version
```

Virtualenv instalatu:

```
$ pip3 install virtualenvwrapper (edo virtualenvwrapper-win)
```

Nahi dugun karpetan kokatuko gara eta ingurune birtuala sortuko dugu:

```
$ virtualenv Django_Birtuala
```

Aktibatzeke, direktorio horretara sartu eta aktibatu:

```
$ cd Django_Birtuala
```

```
$ Scripts\activate
```

```
(Django_Birtuala) E:\Aprobak_Django\NireDjango\Django_Birtuala>
```

Desaktibatzeke:

```
$ Scripts\deactivate
```


5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Lana aurrezteko Django proiektua kopiatuko dugu. Ingurune birtual batean exekutatzeko dugulez, lehenengo pausua betekizunak prestatzea izango da. Ingurune berria instalatu (aurretik sortuta eta aktibatuta egon behar da)

```
$ pip install -r /path/to/requirements.txt
```

Hau eginda, proiektuko karpeta kopiatuko dugu bere tokian, karpeta sartu eta ingurune berria prest dago martxan jartzeko:

```
$ cd personal_portfolio
```

```
$ python manage.py runserver
```

Martxan dagoen ingurunean betekizunak zehaztuko ditugu eta gero kopiatzen dugun tokian betekizun horiek instalatu.

Hasierako ingurunean betekizunak sortzeko:

```
$ pip freeze > requirements.txt
```

```
(Django_Birtuala) E:\Aprobak_Django\NireDjango\Django_Birtuala\personal_portfolio>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 01, 2021 - 18:55:58
Django version 3.1.4, using settings 'personal_portfolio.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Arazo posibleak;

«**execution policy**» arazoa terminaletik pip komandoak exekutatzeko: Windows Power Shell zabaldu administradore bezala eta hau idatzi: «**Set-ExecutionPolicy Unrestricted**»

How do you set an execution policy to unrestricted in VS Code?

Procedure

1. Select Start > All Programs > Windows PowerShell version > Windows PowerShell.
2. Type Set-ExecutionPolicy RemoteSigned to set the policy to RemoteSigned.
3. Type Set-ExecutionPolicy Unrestricted to set the policy to Unrestricted.
4. Type Get-ExecutionPolicy to verify the current settings for the execution policy.

requirements.txt instalatzerakoan, bertsio batzuekin arazoak edukitzea, adibidez, **pycryptodomex==3.9.9** eta kasu honetan, egin behar den gauza bakarra bertsioa aldatzea da: **pycryptodomex==3.15.0** eta berriro saiatzea.

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

proyectos/base.html

```
{% block page_content %}
{% endblock %}
{% block javascript %}
{% endblock javascript %}
```

proyectos/proyecto_index.html (block page_content)

```
{% extends "base.html" %}
{% load static %}
{% block page_content %}
<h1>Proyectos</h1>
<div class="row">
  {% for proyecto in proyectos %}
    <div class="col-md-4">
      <div class="card mb-2">
        
        <div class="card-body">
          <h5 class="card-title">{{ proyecto.titulo }}</h5>
          <p class="card-text">{{ proyecto.descripcion }}</p>
          <p class="card-text">Tecnologia: <span id="{{ proyecto.id }}">{{ proyecto.tecnologia }}</span></p>
          <p class="card-text">Nahi duzu teknologia aldatu?</p>
          <form>
            {% csrf_token %}
            <input type="hidden" name="id" value="{{ proyecto.id }}" />
            <input type="text" name="tecnologia" />
            <input type="submit" value="Aldatu" />
          </form>
          <br>
          <a href="{% url 'proyecto_detail' proyecto.pk %}" class="btn btn-primary">
            Read More
          </a>
        </div>
      </div>
    </div>
  {% endfor %}
</div>
{% endblock page_content %}
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

proyectos/proyecto_index.html (block javascript)

```
{% block javascript %}
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $(document).on('submit', 'form', function (e) {
            e.preventDefault();
            datuak = $(this).serialize();
            $.ajax({
                type: 'POST',
                url: "{% url 'post_tecnologia' %}",
                data: datuak,
                success: function (response) {
                    nodoa = "#" + response.id;
                    $(nodoa).html(response.tecnologia);
                },
                error: function (response) {
                    alert("Ezin da aldatu. Saiatu beranduago" + response);
                }
            })
        })
    })
</script>
{% endblock javascript %}
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

proyectos/urls.py

```
urlpatterns = [  
    path("", views.proyecto_index, name="proyecto_index"),  
    path("<int:pk>/", views.proyecto_detail, name="proyecto_detail"),  
    path("preguntas/<int:pk>/", views.pregunta_guardar, name='pregunta_guardar'),  
    path("post_tecnologia/", views.post_tecnologia, name='post_tecnologia'),  
]
```

proyectos/views.py

```
from django.http import JsonResponse
```

```
.....
```

```
def post_tecnologia(request):  
    post_id = request.POST.get('id')  
    post_tekno = request.POST.get('tecnologia')  
    print(post_id) #id-a konprobatzeko  
    print(post_tekno) #tecnologia konprobatzeko  
    with connection.cursor() as cursor:  
        sql = "UPDATE proyectos_proyecto set tecnologia = %s where id = %s"  
        cursor.execute(sql, [post_tekno, post_id])  
    return JsonResponse({"id": post_id, "tecnologia": post_tekno}, status=200)
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Ajax-Django ariketa

Ajax erabiliz, proiektu baten galderak bistaratuko ditugu:

Proyectos



Mi primer proyecto

Proyecto de desarrollo web.

Tecnologia: Django

Nahi duzu teknologia aldatu?

Nahi dituzu galderak ikusi? Egin klik [hemen](#)

Nahi dituzu galderak ikusi? Egin klik [hemen](#)

- opopoppp
- eee',1);Delete from proyectos_pregunta;--
- ddd
- prueba1
- prueba 2
- fffff

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Ajax-Django ariketa

Pistak

proyecto_index.html

.....

```
<a id="{{ proyecto.pk }}" href="" class="btn btn-primary">
  hemen
</a>
```

.....

```
$("#.....").click(function (e) {
  e.preventDefault();
  let csrftoken = '{{ csrf_token }}'
  datuak = {id: $(this).prop("id")};
  $.ajax({
    type: 'POST',
    headers: {'X-CSRFToken': csrftoken},
    url: .....
    data: datuak,
    dataType: 'json',
    success: .....
```

5. Ajax- adibidea json-a irakurtzen eta bidaltzen Django erabiliz

Ajax-Django ariketa

Pistak

proyectos/views.py

.....

```
preguntas = list(Pregunta.objects.filter(proyecto__id=xxxxx).values())
```

```
return JsonResponse(preguntas, safe=False)
```

xxxxx -> proiektuaren id-a

Estekak:

JQUERY:

<http://api.jquery.com/>

AJAX - Django:

<https://www.pluralsight.com/guides/work-with-ajax-django>