

Ajax

Ajax: Asynchronous JS And XML.

لما أتكلم السير أنا بعت request له أسينكر وني

ال ② صبح
 XML HTTP request = XMLHttpRequest
 XML HTTP Request

كيف نكلم ال server محتاجين website !?

```
function x() {
```

```
  var xhttp = new XMLHttpRequest();
```

لما مشا راس علي
 لما نطلب أتكلم بـ السير
 لما نعمل ال function

```
  console.log(xhttp);
```

ما بع نظري في سي ليه ؟! لأنني طاعت function عار trigger

```
  → function x() {
```

```
    var xhttp = new XMLHttpRequest();
```

```
    console.log(xhttp);
```

مناداة ال function X() مثل ال function

- هلا بيظهر output : form كل محتويات
 ب (0) ويكبر ال form لـ request ياي رقم تكلم في
 ال Server

Error 404 Not Found → status

new XMLHttpRequest()

بدى اكلم السير تبي

- حالات الطلب:

إذا كانت Request للسير كيت بيانات معينة، مشو راجع بر داي؟
موشو المراحل يلي بيرعاير ال Request ؟

ready state

* ال ready state : المراحل التي يمر بمرار request وهي (5) مراحل:

1- الوضع (0):

ready state: 0 : request not initialized
يعني لم ياهو تلو قمية ابتدائية
ال request لا طاهدت شوبك تعد ولا شوبك تغتم
أو تطلب.

2 ready state: 1 : تهيئة الطلب

* يعني حدد ال request شوبك (مثلا: افتح موقع افازي)
جهزت الطلب ← حالة تهيئة الطلب 1

3 ready state: 2 وصول ال request

* ال request وصل للسير بس لا السير ما عمل فيه شي
لا السير ما عمل Check عل ال request

4 ready state: 3 Process النتيجة

* العملية process التي رطبة عل ال request
العملية يلي السير ميشوف فير هل المتع يلي مطلوب من المراقم
الموجودة عنا بالسير ولا مش موجود عندنا أصلا ولا موجود
على السير ف Check عل ال request
(ملاحظة)

4) ready state: 4 response ready

الرد response الرد

note: الحالة (1) هي العرض (0)
الحالة (2) هي العرض (1)
الحالة (3) هي العرض (2)
الحالة (4) هي العرض (3)
الحالة (5) هي العرض (4)

status

بعد فتح الصفحة حالات الطلب يدرس الحالة (5) هي العرض (4)
Check

Status: الموقع راجع عدد على لوحة (3)

- 1] 200 : OK يعني الحالة عام
- 2] 404 : Page not found يعني الموقع لا يوجد
- 3] 403 : Forbidden : مسموع

Note: أحياناً الموقع سيكون موجوداً على السيرفر مع هيك يظهر (404) لأنو
يكون الموقع (مقطع) هذا السيرفر يعني في مقطع كبير قتل الموقع
فالموقع مقطوع من السيرفر

4. راجع ذلك مثال نكلم فيه موقع معين ونجيب من بيانات :

(==) : مقارنة
(=) : مساواة

/ /

مثلاً: موقع json placeholder
دوماً الداتا باي مستخدم json. تتنزل [json] وبتنزل لغة البرمجة
Parse كد المعالجة السابقة الـ back فتعلم شي جديد

كيف ندمج كود الموقع وديت منو البيانات؟

1- function data()

2- Var xhttp = new XMLHttpRequest() اول خطوة: هي البرنر
له فليك تقابلها
يا هو وديت فيفل
هاد

منو دمي قلو؟ لما اكسب الـ XMLHttpRequest حالة الـ readyState تكون
(4) يعني الرد جاهز ويرد علي انه state = 200

3- xhttp.onreadystatechange = function()

الـ readyState مكتوبة بي الـ XMLHttpRequest : و آخذة بي هدا (4)
بالط (3) عي قلو لما هي تغير عن (4)
عانة على xhttp

4- if (this.readyState == 4 && this.status == 200)

هذا دمي قلو: اعطيني الـ text او النص باي جهاز الحقول.

5- console.log(this.responseText)

This: مقصود بي xhttp

عم قلو هادي منو المحتوى النصي تبع الـ response المرفوع بالـ xhttp

يعني هادي النص المرفوع بالرد تبع السيرفر

① readyState تغير / بقي Function

② واختبرك إذا ال readyState = 4 (و) ال status = 200 (= الموقع موجود)

③ هات المحتوى النصي والعرضي بان بار Conside

④ ← قاي: المحتوى النصي يتابع مين؟ يتابع أي موقع

هو ال response عبارة عن الرد يعني بيكون جايي الموقع

بناطابو ويكن انما هاد داتا انو موقع لبي

← ال (XMLHTTP) هي في ادوات method + open

open: هي عبارة عن method موجودة بقوا ال form

بي بار XMLHTTP Request ليك حاضرك بار XMLHttpRequest

هذه ال open method: يكتب جواها الموقع يات بدك تحكيه

منه: method ونكتب طريقة ان يكتب بياكل (3) قيم

أول: (بدايت بيايات ولا نقيف بيايات ولا نختب بيايات)

get ← Post ← DELETE ←

↓
اع ناهضت لقدام

ثاني: URL (الموقع يات بدك بيايات)

ثالث: asynchronous / false

لا يراني الزمن / غير تزامني: Asynchronous: true

يراني الزمن في ترتيب / تزامني: Synchronous: false

الأحداث ← يعني: لو اشأحد انوار logic له سينكر ونس

ادال fun هادي منهية خان logic ثاني ← بييناها وما بينقده فوراً

مع يراني الزمن ← (لا تشغل فوراً)

true: بيغدها فوراً

هيدول متفرع انوبس "[]" لازم فترهم هيدول

6- `Xhttp.open(GET, https://jsonplaceholder, false)`

Method: `GET` (أصل البيانات)
 URL: `https://jsonplaceholder`
 Synchronous: `false` (رأعي الزمان)

صفحة ال Form
 بكتب في الموقع بال
 بي اتصال معو

سك قلونا : البعة هي البيانات في نفذ ال (6)
 Function: `open` (بالزبط)
 بتفرع في مصنع لعلو في البعة هي البيانات
 ال Send ← method أو fun ← (2) بال
 XMLHttpRequest

7. `Xhttp.send()` → Function ال open بي

الخطوة الأخيرة ← لنا تشتغل لازم اعل trigger أو
 call ال Function :

8- `data()` →

هناك مع تشتغل تمام ورم بيبي `jsonData` لخواص هيك فتر
 الموقع ← وهاد ال بيبي `jsonData` بيبي فتر عندي بال `Conside`
 هي عبارة عن `(responseText)`
 ← اذا بيبي حمل هاد `jsonData` لاستعمل بترم `jsonData.parse`

لماذا اصبحت عابطة [5] مضطربة الى ديان رح عبيد - JSON.parse

5. Console.log(JSON.parse(this.responseText))

لما نقول ال JSON 'object'
لما نقدر نحل على
وبعير ال key كمانتويين "

الموقع سام الداتا المعهودة فيه JSON وناحولها ل object

(طبيب) اذا انا لى access او title فقه من هاد object
ياي عندي كيه؟! منفرد بعدين

المثال كامل :

```
function data(){  
  var xhttp = new XMLHttpRequest()  
  xhttp.onreadystatechange = function(){  
    if (this.readyState == 4 && this.status == 200){  
      Console.log(JSON.parse(this.responseText))  
    }  
  }  
  xhttp.open("GET", "http://json.coma", false)  
  xhttp.send()  
  data()  
}
```

object

ال object له (3) أنواع :

1) object literals : الطريقة التقليدية

```
Var mobile1 = {  
  name : "iphone 12",  
  year : "2020" }
```

طرق access Value من object

① Console.log(mobile1.name)

object name له Key

② Console.log(mobile1["name"])

نقطة أقواس (array) ونقطة فراغ
ال Key بيت (" ")

2) object Instant: 5000 طريقة

نقطة object

يعني انشائي object

1// Var mobile1 = new Object()

Capital

بيانات data
نقطة object فاض

2// mobile1.name = "iphone 12" -- ذلك نقط فاض

3// mobile1.year = "2020"

4// Console.log(mobile1)

Console.log(mobile1.name)

الطريقة (1) انشائي object فاض

لذلك لا يبرر في key و value بعد استدعاء للزر

+ (.) + اسم ال key كما في الطريقة (2)

3] object.create (,) {هذه الطريقة قليلة الاستخدام}

Var mobile1 = {}

Var mobile2 = object.create(^{Value1}mobile2, {^{Value2}name: {key: value: ""}})

لماذا نحتاج هذا؟ (مقل العرافة)

ار Value في طريقة: object.create يكتب بطريقة
{value: ""}

Var mobile2: Object.create(^{Coma}mobile2, {^{Coma}name: {value: "iphoner2"},
Year: {value: "2020"}})

mobile2 inherit من object ثاني باي شيء mobile1

DOM

DOM: Document Object Model

OOR: Object Oriented Representation

ال DOM: هي نظام فنية مصطلح OOR وال (2) يعني
صفحة ال html ← index.html التي تتعامل مع ال
tags وال elements الموجودة في html.

* استخدامات ال Dom:

- ① تمكين التفاعل على عنصر داخل html.
 - ② تمكين التفاعل على text موجود في عنصر html.
 - ③ تعديل على Style أي عنصر في html، بحيث يتغير style بال JS.
 - ④ إذا كنت تريد حذف عنصر من html.
 - ⑤ تريد ان تصنيف عنصر في مكان معين من html.
- هام: أي حدث تريد تطبيقه على أي عنصر ←
تطبيقه باستخدام ال Dom ← هو يتعامل مع جميع التفاعلات
الموجودة في html.

ال Document شايعة كاشي بصفحة ال html.

* Console.log(document)

بنيت كاشي: URL شيء، الذاكرة شيء وإذا استندت ولا لا
قل (onclick).

* أي عنصر موجود في html وبه تسمية ← هي طريقة document

#HTML:

<!DOCTYPE html>

<html lang="en">

<head>

2/ <meta charset="UTF-8">

3/ <meta http-equiv=" "

4/ <meta name="viewport" "

<title> Task </title>

<link

<link

<link

</head>

<body>

<div class="container">

<h2 class="heading" id="title1">

< class="decst" > this is p </p> ^{Data1 } ^{Islam </h2>}

 This is span

<p class=" " > This is p </p>

 This is span

<p class=" " > " </p>

 "

<p class=" " > " </p>

 "

</div>

</body>

```
<form>
  <input type="text">
  <input type="submit">
</form>
```

JS :

اعرض العنصر الرابع بالترتيب ^{عند} ~~document[4]~~
 ← ~~undefined~~ بعض الشيء
 ⇒ document.all[4]

<meta name="viewport ... > ← output :

بإضافة العنصر الرابع بالترتيب
 Console.log(document.all[5]) → <title>Task </title>

Note : Document يسمى العناصر elements يسمى بالترتيب

❶ سؤال هام جداً جداً : إذا سألنا h2 استيعاب

```
Var x = document.getElementById("title1")
```

Console.log(x.innerText) → Data1 Islam
 h2 له هو id span له هو class

Console.log(x.textContent) → Data1 Islam

لكن الفرق بين innerText وبين textContent

يعطي الـ span ما بين الـ `<h2>` ← `display: none`

لا غناء المنصر

```
<h2 id="title" > <span class="sp1" style="display: 'none';>
Data1 </span> Islam </h2>
```

`Console.log(x.innerText)` → Islam

`Console.log(x.textContent)` → Data1 Islam

الـ `textContent` : يعيد الـ `access` لجميع العناصر الحقيقية

والظاهر

الـ `innerText` : لا يعيد ان يرى الـ `display: block`

لعرض المحتوى الظاهر فقط

و لكن اذا سويت عدل في الـ (2) :

`x.innerText = "Ahmad"`

مباشرة لا اعد refresh مع يعيد Ahmad

و لكن علق لا اعد

`Console.log(x.innerText)` → Ahmad

`Console.log(x.textContent)` → Ahmad

بين الـ `textContent` ما عرض الـ `span` مكانه

الجواب : لا قلنا `x.innerText: 'Ahmad'` هدي عدل كلني بين

تاع البداية والاية يعني : كلني بالزحود :

```
<h2 id="title" > <span class="sp1" style="display: none;">Data1 </span> Islam </h2>
```

↓
تسجل كل هديل بين الجواخ الـ (h2)

Ahmad

يغير محتوى كل سنو بين تاخ البداية وتاخ الزاية

X.textContent = "osama"

بعد تغيير النص في الـ innertext = بعد كل شيء وداخليه `ch1` `</h1>`

Console.log(x.innertext) → # Osama

Console.log(x.textContent) → # Osama

CSS

كيف نعدل style على عنصر

* إضافة CSS في الـ `head` : نكتب ثم العنصر + `(.style)`
 `(.property="value")`
 أريد تغيير لون الـ `h2` :

X.style.color = "red"

X.style.backgroundColor = "yellow"

Capital

X.style.fontSize = "70px"

* دراسة إضافة العناصر بالـ `addEventListener` :

Var y = document.getElementsByTagName("div")

Console.log(y)
 لكل العناصر التي أمده `class=dec1` (P) عنابر

ال (y) عبارة عن `array`

يعرّفنا في هذا الـ `array` كل العناصر التي `class="dec1"`

إذا أردنا عمل الـ (P) الثاني واستغل عليه كيف ؟

→ إذا (y) `P` نرسل `array` = `y` بعد أن نمرر البيانات التي فيه

أو نعدل في سبي مصفوفة الـ `array`

جاء انوال (y) بترجم \Rightarrow array

index 0 \rightarrow p
index 1 \rightarrow span
index 2 \rightarrow p
}

index 7

انترفيو - علاقة العنصر الأخير في array بـ length

\rightarrow length - 1 = index العنصر الأخير

ال (p) الثاني هو العنصر الثالث في ال array \Rightarrow index=2

y[2].style.backgroundColor = "red"

index=2

ولكن علة لا طبقت لماذا؟!

لأننا لازم نتأكد أنه اسم ال class قبل الاسم بأي حاصيتوب

getElementByClassName("")

* هذا أنا حاصيتين مع به الميزة
كان كاتب غير اسم (ن) (ف)

y[2].style.color = "white"

* SetAttribute(" ", " ") إضافة خاصية

y[2].setAttribute("price", "5000\$")

الخاصية اسم ال خاصية

* getAttribute : إذا اردت صيغ ال attribute قبل العنصر اقرأها

أقرأ ال Value
بال attribute

Console.log (X[2])

بسيطين كلشي آخر ال (P) الثاني .

← ليس عظمة ال (8) عناصر نفس ال class يدك طاعطرين id ؟
لأن ال id ما بيغير عناصر نفس ال id .
unic ← غير العنصر

- سيعطيني لون خلفية لذلك (P) X[0].style.backgroundColor="red"

لو سيعطيني (100) عنصر مثلاً أي ال (8) عناصر نفس لون الخلفية كيف ؟ for

Var y = document.getElementsByClassName("decst")

Console.log(y)

طول المصفوفة
(الجواب) for (var i = 0; i <= y.length; i++) {

y[i].style.backgroundColor = "red"

* إذا بقيت عندك في محتوى ال (P) الثاني : الصفحة : </p>

X[2].textContent = "Ahmad Gomaa"

من الصفحة
innerText

* كيف استدعي كل ال (P) الموجودة في الصفحة :

* tag Name:

Var x = document.getElementsByTagName("p")

Console.log(x) → array

X[4].style.Color = "red"

ال (P) الخامس:

العنصر
الخامس

ملاحظة: الأفضل في الاستخدام `querySelector` & `querySelectorAll` لأنه استبعاد أكثر من عنصر

traversing

<form> <input type="text"> <input type="submit"> </form>

Var inp1 = document.querySelector("input [type = submit].")

استبعاد عن طريق inp1

* **Parent Element:** العنصر الأب

Console.log (inp1.parentElement) → # form [inp1]

* **Parent Node:** العنصر الأب

Console.log (inp1.parentNode) → # form [inp1]

* إذا لم يعد ال [submit] الكلمة إلى طريقة inp1

يبدأ هنا بين تاج الباقة بالزينة → inp1.textContent = "send"
ولكن إذا لم يعد ال value وليس النص بين تاج الباقة بالزينة

inp1.value = "send" → [send] ✓

* إذا لم يعد ال عا لناب أحوار form

inp1.parentElement.style.backgroundColor = "red"

.style.width = "200px"

.height = "50px"

Var f1 = document.querySelector("form")

* Child Nodes : تقرأ العنصرات كأشياء

Console.log(f1.childNodes) → [text, input, text, input, text]

(5) أولاد. ← تسمى (كل) الأبناء حتى العنصر
يعتبرها (ابن)

عنصر <form> input
text <input type="text">
text <input type="submit">
text </form>

إذا العنصر العنصرات يعرفون عنصر (input, input)

إذا علت عنصر عنصر (tab) بين input و input
(input, text, input)

* Children الأبناء الحقيقية (العناصر) الأبناء بدون العنصر

Console.log(f1.Children) → [input, input]

* FirstChild : الابن الأول تقرأ العنصرات

Console.log(f1.firstChild) → text

* FirstElementChild : الابن العنصر الأول

Console.log(f1.firstElementChild) → input

<form> ---
</form>
text <div>

</div>

* nextSibling : الأخ الأبنة مباشرة (بقراءة العنصرات)

Console.log(f1.nextSibling) → text

* nextElementSibling : الأخ العنصر المباشر

Console.log(f1.nextElementSibling) → div