

ما هو الـ

JSON

الكاتب : حسين قبلان

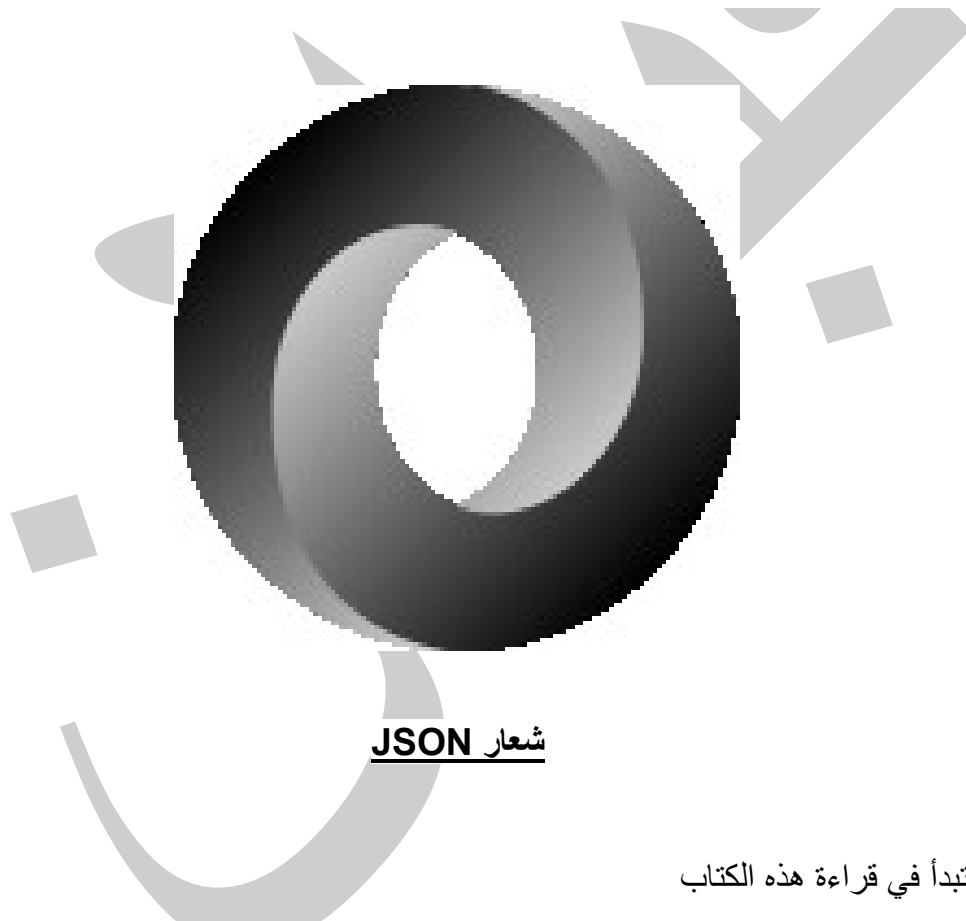
فلاح

ما هو JSON

JSON أو تدوين كائن جافا سكريبت هو معيار مفتوح خفيف الوزن يستند إلى نص مصمم لتبادل البيانات القابل للقراءة البشرية. تم تحديد تنسيق JSON في الأصل من قبل دوغلاس كروكفورد، ويرد وصفه في RFC 4627. إنترنت الرسمي ل JSON هو اسم التطبيق / json.ملحق اسم الملف JSON هو. json.

سوف نتعامل مع لغات مثل بي إتش بي، بيرل ، بايثون ، روبي ، جافا ، الخ.

وقد تم كتابة هذا الكتاب لمساعدة المبتدئين على فهم الوظائف الأساسية للتدوين كائن (JSON) JavaScript لتطوير تنسيق تبادل البيانات. بعد استكمال هذا الكتاب سيكون لديك فهم جيد من JSON وكيفية استخدامها مع جافا سكريبت، اجاكس، بيرل ، الخ.



شعار JSON

المتطلبات قبل ان تبدأ في قراءة هذه الكتاب

يجب أن يكون لديك فهم أساسي للويب تطبيق العمل عبر HTTP وأنا افترض أن لديك معرفة أساسية في جافا سكريبت.

(2).....	المواضيع
(3).....	نظرة عامة
(3).....	استخدامات ال JSON
(3).....	خصائص ال JSON
(5).....	الفرق بين JSON و XML
(7).....	بناء الجمل في JSON
(8).....	أنواع البيانات في JSON
(9).....	الارقام في JSON
(10).....	النصوص في JSON
(11).....	القيمة المنطقية في JSON
(11).....	المصفوفة في JSON
(12).....	التعليقات في JSON
(12).....	JSON مع Php
(15).....	JSON مع AJAX

نظرة عامة

JSON أو تدوين كائن جافا سكريبت هو معيار مفتوح خفيف الوزن يستند إلى نص مصمم ل تبادل البيانات القابل للقراءة البشرية. الاصطلاحات المستخدمة من قبل JSON معروفة للمبرمجين، والتي تشمل C ، ++C ، جافا ، بايثون ، بيرل ، الخ

- يرمز JSON الى JavaScript Object Notation .
- تم انشاءه بواسطة دوجلاس كروكفورد.
- تم تصميمها لتبادل البيانات التي يمكن للبشر قراءتها
- لقد تم انشاء جنباً مع الإصدار الثالث من جافا سكريبت عام 1999
- يكون امتداد ملف JSON هو {filename.json}
- الاسم الموحد والشائع هو {public.json}

استخدامات ال JSON

- يتم استخدامه أثناء كتابة التطبيقات المستندة إلى JavaScript والتي تتضمن ملحقات المتصفح والمواقع
- يتم استخدام تنسيق JSON للتسلسل ونقل البيانات المهيكلية عبر الشبكة الإتصال.
- يتم استخدامه بشكل أساسي لنقل البيانات بين الخادم وتطبيقات الويب.
- تستخدم خدمات الويب وواجهات برمجة التطبيقات تنسيق JSON لتوفير البيانات العامة.
- يمكن استخدامه مع لغات البرمجة الحديثة.

خصائص JSON

- JSON سهل القراءة والكتابة.
- له تنسيق تبادل خفيف يعتمد على النص.
- JSON لغة مستقلة

مثال بسيط في JSON

يوضح المثال التالي كيفية استخدام JSON لتخزين المعلومات المتعلقة بالافلام الأسماء ونوع الفلم وتقييم الفلم

```
[
  {
    "Title": "Born on the Fourth of July",
    "Major Genre": "Drama",
    "IMDB Rating": 7.2,
    "IMDB Votes": 32108
  },
  {
    "Title": "Basquiat",
    "Major Genre": "Drama",
    "IMDB Rating": 6.7,
    "IMDB Votes": 7935
  }
]
```

بعد فهم البرنامج أعلاه، سنجرب مثالاً آخر. دعونا ننشئ ملف `testjson.html` دعنا نسميه

```
<html dir="rtl">
<head>
  <title> مثال على JSON </title>
  <script language="javascript">
    var object1 = { "language": "Python", "author": "حسين قبلان" };
    document.write("<h1>JSON مع JAVASCRIPT</h1>");
    document.write("<br>");
    document.write("<h3>اللغة = " + object1.language + "</h3>");
    document.write("<h3>الكاتب = " + object1.author + "</h3>");
    var object2 = { "language": "JavaScript", "author": "سعدون الساعدي" };
    document.write("<br>");
    document.write("<h3>اللغة = " + object2.language + "</h3>");
    JSON
    7
    document.write("<h3>الكاتب = " + object2.author + "</h3>");
    document.write("<hr />");
    document.write(object2.language + " + "لغة البرمجة يمكن دراستها "
      + "من الكتاب بقلم " + object1.author);
    document.write("<hr />");
  </script>
</head>

<body>
</body>
</html>
```

والان نقوم بحفظ الملف تم نفتحه في متصفح كروم او اي متصفح اخر يدعم جافا سكريبت سيكون الناتج كالاتي :

مثال على JSON مع JAVASCRIPT

اللغة = Python

الكاتب = حسين قبالان

اللغة = JavaScript

الكاتب = سعدون الساعدي

JavaScript لغة البرمجة يمكن دراستها من الكتاب بقلم حسين قبالان

XML و JSON

فيما يلي قائمة بالاختلافات بين XML و JSON.

ت	JSON	XML
1	يرمز JSON إلى JavaScript Object Notation.	XML تعني لغة التوصيف الموسعة.
2	JSON سهل القراءة والكتابة.	XML أقل بساطة من JSON.
3	JSON سهل التعلم.	XML أقل سهولة من JSON.

4	JSON موجه للبيانات.	XML هو مستند المنحى.
5	لا يوفر JSON إمكانيات العرض.	يوفر XML القدرة على عرض البيانات لأنها لغة ترميز.
6	يدعم JSON الصفيف.	XML لا يدعم المصفوفة.
7	JSON أقل أمانًا من XML.	XML أكثر أمانًا.
8	ملفات JSON أكثر قابلية للقراءة من ملفات XML.	ملفات XML أقل قابلية للقراءة البشرية.
9	يدعم JSON نوع بيانات النص والأرقام فقط.	يدعم XML العديد من أنواع البيانات مثل النص والأرقام والصور والمخططات والرسوم البيانية وما إلى ذلك. علاوة على ذلك ، يقدم XML خيارات لنقل تنسيق أو هيكل البيانات مع البيانات الفعلية.

مثال JSON

```
{
  "employees": [
    {
      "name": "Vimal",
      "email": "vjaiswal1987@gmail.com"
    },
    {
      "name": "Rahul",
      "email": "rahul12@gmail.com"
    },
    {
      "name": "Jai",
      "email": "jai87@gmail.com"
    }
  ]
}
```

مثال XML

```
<employees>
  <employee>
    <name>Vimal</name>
    <email>vjaiswal1987@gmail.com</email>
  </employee>
  <employee>
    <name>Rahul</name>
    <email>rahul12@gmail.com</email>
  </employee>
  <employee>
    <name>Jai</name>
    <email>jai87@gmail.com</email>
  </employee>
</employees>
```


أوجه التشابه بين XML و JSON

- كلاهما بسيط ومفتوح
- كلاهما يدعم يونيكود. لذا فإن التدويل مدعوم من قبل JSON و XML.
- كلاهما يمثل بيانات تصف الذات.
- كلاهما قابل للتشغيل المتبادل أو مستقل عن اللغة.

بناء الجملة في JSON

دعنا نلقي نظرة سريعة على البنية الأساسية لـ JSON. يعتبر بناء جملة JSON في الأساس ملف مجموعة فرعية من بناء جملة JavaScript ؛ يتضمن ما يلي:

- يتم تمثيل البيانات في أزواج الاسم / القيمة.
- الأقواس المتعرجة تحمل الكائنات ويتبع كل اسم بـ ":" (النقطتان) ، الاسم / القيمة الأزواج مفصولة بفاصلة.
- الأقواس المربعة تحمل المصفوفات والقيم مفصولة بعلامة (فاصلة).

مهلا لم تفهم اليس كذلك اذا دعني ابسطه لك عبر هذه المثال

```
[
  {
    "Title": "Born on the Fourth of July",
    "Major Genre": "Drama",
    "IMDB Rating": 7.2,
    "IMDB Votes": 32108
  },
  {
    "Title": "Basquiat",
    "Major Genre": "Drama",
    "IMDB Rating": 6.7,
    "IMDB Votes": 7935
  }
]
```

يدعم JSON بنيتي البيانات التاليتين:

- جمع أزواج الاسم / القيمة: هيكل البيانات هذا مدعوم بمختلف لغات البرمجة.
- قائمة القيم المرتبة: تتضمن مصفوفة أو قائمة أو متجه أو تسلسل إلخ.

أنواع البيانات

يدعم تنسيق JSON أنواع البيانات التالية:

النوع	الوصف
الارقام	تكون القيمة عبارة عدد صحيحا أو حقيقيا أو فواصل عشرية.
النصوص	مجموعة من الأحرف النصية العادية تشكل عادة كلمة.
Boolean قيمة منطقية	ولها احتمالان True او False.
Array المصفوفة	مجموعة من القيم المترابطة على سبيل المثال: رواتب الموظفين ويتم تمثيلها كالتالي[2000, 5000, 6000]
Value القيمة	يمكن ان يكون نصوصا او ارقام او True , False او خالية والخ
Object الكائن	مجموعة مترابطة من أزواج من المفاتيح / القيم. Key / Value
Whitespace	يمكن استخدامها بين أي زوج من الرموز المميزة
null	فارغ

الأرقام

- إنه تنسيق مزدوج الدقة في JavaScript ويعتمد على التنفيذ
- لا يتم استخدام التنسيقات الثمانية والسادسية العشرية.
- لا يتم استخدام NaN أو Infinity في الأرقام

يوضح الجدول التالي أنواع الأرقام:

النوع	الوصف
عدد صحيح (integer)	الأرقام من 1 إلى 9 ، 0 وموجبة أو سالبة
جزء Fraction	كسور مثل 3. ، 9.
الأس Exponent	الأس مثل e ، e + ، e- ، E ، E +

بناء الجملة: Syntax

```
var json-object-name = {"string" : number_value, .....}
```

مثال

مثال يوضح نوع بيانات الرقم ، يجب عدم ذكر القيمة:

```
var obj = {"marks": 97}
```

النصوص

- وهو عبارة عن تسلسل من صفر أو أكثر من أحرف Unicode المزدوجة المقتبسة بشرطة مائلة للخلف
- الحرف هو سلسلة أحرف مفردة ، أي سلسلة بطول حرف 1.

يعرض الجدول العديد من الأحرف الخاصة التي يمكنك استخدامها في النصوص من ملفات JSON:

النوع	الوصف
"	"اقتباس مزدوج"
\	شرطة مائلة للخلف
/	مائل لأمام
b	مسافة للخلف
f	شكل النص
n	خط جديد
r	إرجاع
t	علامة التبويب الأفقية
u	أربعة أرقام سداسية عشرية

بناء الجملة: Syntax

```
v var json-object-name = { string : "string value", .....}
```

مثال

مثال يوضح نوع بيانات النصوص :

```
var obj = {"name": "حسين مربع"}
```

قيمة منطقية (Boolean)

يتضمن قيم صحيحة أو خاطئة.

Syntax: بناء الجملة:

```
var json-object-name = { string : true/false, .....}
```

مثال

```
var obj = {"name": "حسين مربع", "marks": 33, "distinction": false}
```

المصفوفة (Array)

- إنها مجموعة مرتبة من القيم.
- هذه محاطة بأقواس مربعة مما يعني أن المصفوفة تبدأ بـ [] و ينتهي مع [].
- القيم مفصولة بعلامة (فاصلة). ,,,,,,
- يمكن بدء فهرسة الصفيف عند 0 أو 1
- يجب استخدام المصفوفات عندما تكون أسماء المفاتيح أعدادًا صحيحة متسلسلة.

Syntax: بناء الجملة

```
[ value, .....]
```

مثال

مثال يوضح مصفوفة تحتوي على كائنات متعددة:

```
{
  "names": [
    { "language": "Java" , "edition": "second" },
    { "language": "C++" , "lastName": "fifth" },
    { "language": "C" , "lastName": "third" }
  ]
}
```

التعليقات في JSON

JSON لا يدعم التعليقات. إنه ليس معيار.

لكن يمكنك القيام ببعض الحيل مثل إضافة سمة إضافية للتعليق في كائن JSON ، على سبيل المثال:

```
{
  "employee": {
    "name": "حسين مربع",
    "salary": 56000,
    "comments": "اصلع ومربع"
  }
}
```

هنا ، يمكن التعامل مع سمة "التعليقات" على أنها تعليق.

JSON مع PHP

تسمح لنا PHP بتشفير وفك تشفير JSON بمساعدة وظائف `json_encode()` و `json_decode()`.

PHP JSON_encode -1

تعرض الدالة `json_encode()` تمثيل JSON لقيمة. بمعنى آخر ، يحول متغير PHP يحتوي على مصفوفة إلى JSON.

بناء الجملة: Syntax

```
string json_encode ( mixed $value [, int $options = 0 [, int $depth = 512 ]] )
```

PHP json_encode مثال 1

```
<?php
$arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
echo json_encode($arr);
?>
```

سيكون الناتج كالاتي:

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```

مثال اخر :

```
<?php
$arr2 = array('firstName' => 'حسين', 'lastName' => 'قيلان', 'email' => 'husseinalikaplan72@gmail.com');
echo json_encode($arr2);
?>
```

سيكون الناتج كالاتي:

```
{"firstName":"حسين","lastName":"قبلان","email":"husseinalikaplan72@gmail.com"}
```

Php JSON_decode -2

تقوم الدالة (`json_decode()`) بتشفير سلسلة JSON. بمعنى آخر ، فإنه يحول سلسلة JSON إلى متغير php

بناء الجملة: Syntax

```
mixed json_decode ( string $json [, bool $assoc = false [, int $depth = 512 [, int $options = 0 ]]] )
```

مثال 1 :

```
<?php
$json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';
var_dump(json_decode($json, true)); // true تعني أنه سيتم تحويل الكائن المرتجع إلى مصفوفة ترابطية
?>
```

سيكون الناتج :

```
array(5) {
  ["a"] => int(1)
  ["b"] => int(2)
  ["c"] => int(3)
  ["d"] => int(4)
  ["e"] => int(5)
}
```


مثال 2 :

```
<?php
$json2 = '{"firstName" : "حسين", "lastName" : "قَبْلان", "email" : "husseinalikaplan72@gmail.com"}';
var_dump(json_decode($json2, true));
?>
```

سيكون الناتج كالاتي :

```
array(3) {
  ["firstName"]=> string(5) "حسين"
  ["lastName"]=> string(5) "قَبْلان"
  ["email"]=> string(15) "husseinalikaplan72@gmail.com"
}
```

مثال ل JSON مع AJAX

يمكننا الحصول على بيانات JSON بواسطة كود AJAX. توفر AJAX إمكانية الحصول على استجابة غير متزامنة. لا يعيد تحميل الصفحة ويحفظ النطاق الترددي.

مثال :

دعنا نرى مثلاً بسيطاً للحصول على بيانات JSON باستخدام كود AJAX.

```
<html>
<head>
<meta content="text/html; charset=utf-8">
<title>اجاكس مع جيسون بواسطة حسين قبيلان</title>
<script type="application/javascript">
function load()
{
    var url = "http://date.jsontest.com/"; // استخدام اي رابط للحصول على بيانات جيسون
    var request;

    if(window.XMLHttpRequest){
        request=new XMLHttpRequest(); // من اجل متصفح كروم او موزيلا
    }
    else if(window.ActiveXObject){
        request=new ActiveXObject("Microsoft.XMLHTTP"); // من اجل متصفح ايدج
    }
    request.onreadystatechange = function(){
        if (request.readyState == 4 )
        {
            var jsonObj = JSON.parse(request.responseText); // ارجاع كائنات جيسون
            document.getElementById("date").innerHTML = jsonObj.date;
            document.getElementById("time").innerHTML = jsonObj.time;
        }
    }
    request.open("GET", url, true);
    request.send();
}
</script>
</head>
<body>

التاريخ: <span id="date"></span><br/>
الوقت: <span id="time"></span><br/>

<button type="button" onclick="load()">تحميل المعلومات</button>
</body>
</html>
```

سيكون الناتج كالآتي :

:التاريخ

:الوقت

تحميل المعلومات

جميع الحقوق محفوظة للكاتب : حسين قبلان

تابع صفحة الكاتب على الفيسبوك للمزيد: <https://www.facebook.com/Husseinkaplan21>

