**JSON:** A JSON (JavaScript Object Notation, JavaScript objektumjelölés) emberek számára is olvasható–írható, programozottan pedig könnyen feldolgozható és előállítható adatcsere-formátum.

Az alábbi sor egy JavaScript objhektumot definiál.

var car = {type:"Fiat", model:500, color:"white"};

Alul egy egyszerű JSON dokumentum található:

{

"book": [

{

"id":"01",

"language": "Java",

"edition": "third",

"author": "Herbert Schildt"

},

{

"id":"07",

"language": "C++",

"edition": "second"

"author": "E.Balagurusamy"

} ]

}

A [ ] jelek tömböt reprezentálnak.

A JSON alap adattípusai:

* Szám

Lehet lebegőpontos és egész. A C és Java számaihoz hasonló. A különbség az, hogy oktális és hexadecimális formátum itt nem használható.

* String

A karakterlánc nulla vagy több, idézőjelek közé zárt Unicode karakter, szükség szerint visszaper-jellel kivédve. A karakter egy hosszúságú karakterláncnak felel meg. A karakterlánc nagyban hasonlít a C vagy Java karakterláncaihoz.

* Boolean

Értéke true (igaz) vagy false (hamis).

* Tömb

A tömb értékek rendezett halmaza. A tömb [ (nyitó szögletes zárójel)-lel kezdődik és ] (záró szögletes zárójel)-lel zárul. Az értékeket , (vessző)-vel választjuk el egymástól. Az értékeknek nem kell azonos típusúnak lenniük.

* Objektum

Az objektum név–érték párok rendezetlen halmaza. Egy objektum { (nyitó kapcsos zárójel)-lel kezdődik és } (záró kapcsos zárójel)-lel zárul. A ':' karakter választja el a kulcsot és az értéket. A név–érték párok , (vessző)-vel tagoltak. A neveknek sztringeknek kell lenniük és különbözniük egymástól. Érték lehet idézőjelek közé írt karakterlánc, szám, logikai igaz, logikai hamis, null, objektum vagy tömb. A struktúrák egymásba ágyazhatók.

* null Üres érték

Példa szám típusra: var obj = {marks: 97}

Tömb példa:

**{**

**"books": [**

**{ "language":"Java" , "edition":"second" },**

**{ "language":"C++" , "lastName":"fifth" },**

**{ "language":"C" , "lastName":"third" }**

**]**

**}**

Látható ,hogy [] között a tömb elemei vannak.Itt a tömb elemei objektumok.

Objektumok {} között vannak!!!!!

Objektum példa:

{

"id": "011A",

"language": "JAVA",

"price": 500,

}

JSON obketumokat létrehozhatunk JavaScriptben.

Példák:

-üres objektum: var JSONObj = {};-var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";

Konkrét példa:Látható,hogy létrehoztam egy Pascal tömböt és egy Scala tömböt.A Pascal tömbnek két eleme van,amik objektumok.Olyan objektumok,amiknek van name és price attribútumuk.alert() eredménye „Pascal Made Simple”

<script>

var books = { "Pascal" : [

{ "Name" : "Pascal Made Simple", "price" : 700 },

{ "Name" : "Guide to Pascal", "price" : 400 }

],

"Scala" : [

{ "Name" : "Scala for the Impatient", "price" : 1000 },

{ "Name" : "Scala in Depth", "price" : 1300 }

]

}

alert(books.Pascal[0].Name);

</script>

Egyszerű példa,hogy JQueryben :

A data.json tartalma:

{

"name": "Zara Ali",

"age" : "67",

"sex": "female"

}

A html ben lévő jquery:

$.getJSON("data.json", function(jd) {

alert(jd.age);

alert(jd.name);

});

Ez a jd egyetlen objektumot reprezentál jelenleg.

Kicsit komplikáltabb példa:

A data.json dokumentum:

{ "users":[

{

"firstName":"Ray",

"lastName":"Villalobos",

"joined": {

"month":"January",

"day":12,

"year":2012

}

},

{

"firstName":"John",

"lastName":"Jones",

"joined": {

"month":"April",

"day":28,

"year":2010

}

}

]

}

jquery:

$.getJSON("data.json", function (json) {

$.each(json,function(index,value){

alert(value[0].joined.month);

});

alert(json.users[0].firstName); //January lesz az eredmény

});

Ha a két firstname-t akarom megkapni akkor:

$.getJSON("data.json", function (json) {

$.each(json,function(index,value){

$.each(value,function(index,val){

alert(val.firstName);

});

});

});

A külső each()-vel a json ( a két külső {})összes objektumát járja be,ami csak a users.Ezáltal megkapjuk a userst.Majd a users objektumot átadjuk egy másik $.each()-nek bejárásra.

A json a users objetumot jelöli ki,ezért kell indítani még1 $.each()-et ami

A következő megoldás mind a 2 obktumnak kiírja a firstName tagját:

$.getJSON("data.json", function (json) {

$.each(json,function(index,value){

$.each(value,function(index,value){

alert(value.firstName);

});

});

});