

# **Módulo 1b - Javascript**

T04 #TDW #MCTW

10/11/2021

Carlos Santos



# ***HTML CSS and JavaScript for Beginners - A Web Design Course***

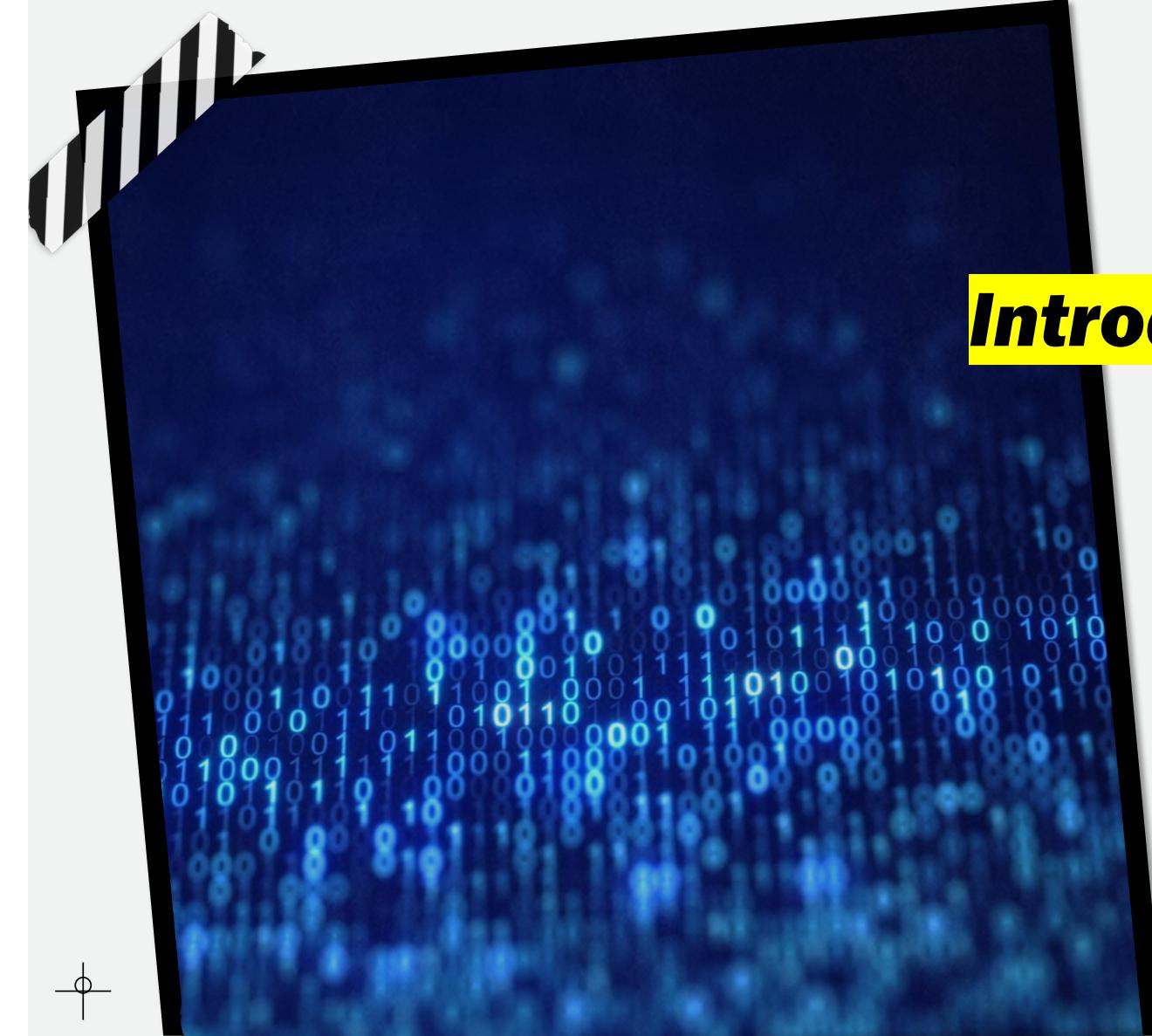
<https://learning.oreilly.com/videos/html-css-and/9781838551278/>



# ***Chapter 6: JavaScript***

## ***JSON and AJAX***





# ***Introduction to JSON***

- JavaScript Object Notation

# **What is JSON**

- Transferir informação na Web
- {...}
- Par -> Name: Value
- Separar pares com vírgula
- Tomou o lugar que antes era ocupado pelo XML



# **Basics of JSON**

key/name value pairs

```
{ "name" : "value" }
```

Objects are comma separated

```
{ "name1" : "value" , "name2" : "value" , "name3" : "value" }
```

Arrays have square brackets with values separated by comma

```
{ "name" : [ { "name" : "value" } , { "name" : "value" } ] }
```



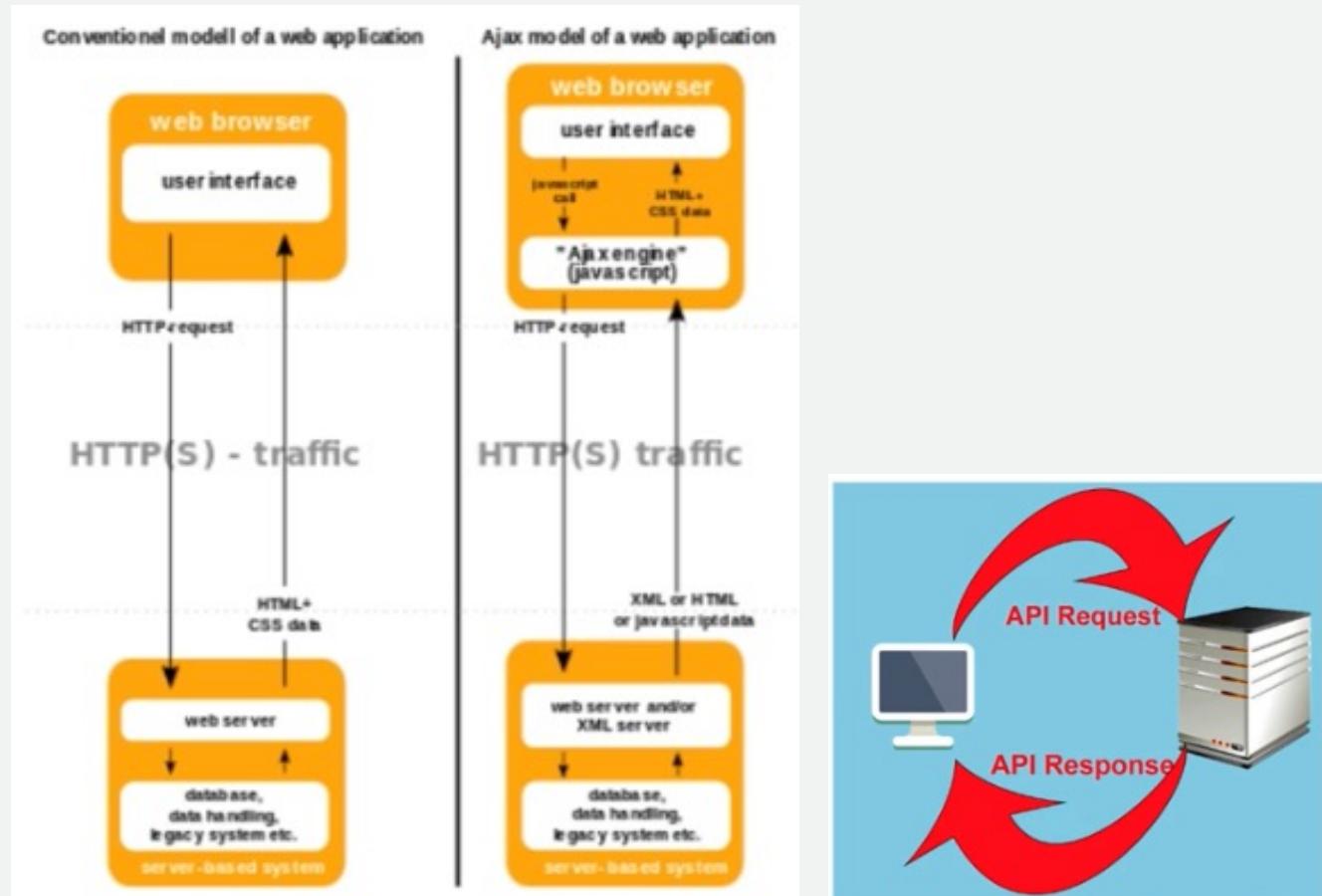
# ***Introduction to JSON***

- Para facilitar a leitura de JSON
- [jsonlint.com](http://jsonlint.com)
- Procurem uma extensão para o vosso browser
  - + Chrome: JSON Viewer
- Recursos
  - + <https://randomuser.me>
  - + <https://randomuser.me/api/?results=10>



# JSON as Data

- Informação estruturada
- APIs



# **JSON as Data**

- Data Types
  - + Number {"name": 10}
  - + String {"name": "Hello world"}
  - + Boolean {"name": true}
  - + Array {"name": [{"name1": 1}, "hello", "world"]}
  - + Object {"name": {"name1": 1, "name2": 2}}
  - + Null {"name": null}
- <https://jsonschema.net>



```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "age": 25,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021"  
  },  
  "phoneNumber": [  
    {  
      "type": "home",  
      "number": "212 555-1234"  
    },  
    {  
      "type": "fax",  
      "number": "646 555-4567"  
    }  
  ],  
  "gender": {  
    "type": "male",  
    "value": true  
  }  
}
```

```
<person>  
  <firstName>John</firstName>  
  <lastName>Smith</lastName>  
  <age>25</age>  
  <address>  
    <streetAddress>21 2nd Street</streetAddress>  
    <city>New York</city>  
    <state>NY</state>  
    <postalCode>10021</postalCode>  
  </address>  
  <phoneNumber>  
    <type>home</type>  
    <number>212 555-1234</number>  
  </phoneNumber>  
  <phoneNumber>  
    <type>fax</type>  
    <number>646 555-4567</number>  
  </phoneNumber>  
  <gender>  
    <type>male</type>  
    <value>true</value>  
  </gender>  
</person>
```

```
firstName: John  
lastName: Smith  
age: 25  
address:  
  streetAddress: 21 2nd Street  
  city: New York  
  state: NY  
  postalCode: '10021'  
phoneNumber:  
- type: home  
  number: 212 555-1234  
- type: fax  
  number: 646 555-4567  
gender:
```

# JSON vs XML vs YAML

## **JSON vs JavaScript Object**

- JSON obriga a chaves terem aspas
- {"foo": "bar"}
- var o = {foo: "bar"};
- Exercício: <https://jsoneditoronline.org/>



## **JSON Details**

- ```
var myJSON1 = {};
myJSON1.car1 = "black";
myJSON1.car2 = "blue";
```
- ```
var myJSON2 = {};
myJSON2["car1"] = "black";
myJSON2["car2"] = "blue";
```
- ```
var myJSON3 = {"car1": "black", "car2": "blue"};
```



## **JSON Details**

- var myObj = {"firstName": "Mike", "lastName": "Smith"};  
var name = "Name";  
  
// Output JSON to HTML  
output1.innerHTML = myObj.firstName;  
output2.innerHTML = myObj['last' + name];



## **Arrays of Objects**

```
var obj1 = {  
    "car":["blue","black"]  
};
```

obj1.car[0]

```
var obj2 = {  
    "car1":{ "color":"blue" },  
    "car2":{ "color":"black" }  
};
```

obj2.car1.color

```
var obj3 = {  
    "car1":{  
        "color":"blue",  
        "model":"Mustang"  
    },  
    "car2":{  
        "color":"black",  
        "model":"Honda"  
    }  
};
```

obj3.car1.color



```
var obj = {  
  "people": [  
    {  
      "firstName": "Laurence",  
      "last": "Svekis",  
      "age": 30  
    },  
    {  
      "firstName": "Mike",  
      "last": "Smith",  
      "age": 50  
    }  
  ]  
}
```

## **Array of objects**

- `obj.people[0].firstName`



```
var obj = {
  "people": [
    {
      "firstName" : "Laurence",
      "last" : "Svekis",
      "age" : 30
    },
    {
      "firstName" : "Mike",
      "last" : "Smith",
      "age" : 50
    }
  ]
}
```

```
for(var x=0;x<obj.people.length;x++){
  var person = obj.people[x];
  console.log(person);
  output1.innerHTML += person.firstName + " "
                      + person.last + "<br>";
}
```

## ***Loop Object Data***

,,,





```
var temp = {
    "firstName": "Alex"
    , "last": "Micheals"
    , "age": 22
}
obj.people.push(temp);
```

**Add to Object**



## **JSON Methods**

- `JSON.stringify(...);`
- `JSON.parse(...);`



## **Local Storage Values**

```
var temp = {  
    "firstName": "Alex"  
    , "last": "Micheals"  
    , "age": 22  
}  
var tempString = JSON.stringify(temp);  
//localStorage.setItem('test',tempString);  
var tempObj = localStorage.getItem('test');  
console.log(tempObj);  
var obj2 = JSON.parse(tempObj);
```



# **JavaScript**

## **Fetch**

- AJAX – XMLHttpRequest
- Fetch API

```
const url =  
  "https://api.myjson.com/bins/hqys2";  
fetch(url).then(function(response){  
  return response.json()  
}).then(function (data){  
  console.log(data);  
})
```



# **JavaScript**

## **Fetch**

```
const output =  
    document.getElementById("output");  
const url = "https://randomuser.me/api/";  
fetch(url).then(function(rep){  
    return rep.json()  
}).then(function(data){  
    console.log(data.results[0]);  
    let person = data.results[0].name;  
    output.innerHTML = person.first + " " +  
        person.last;  
})
```



## Long Function Syntax

```
var url = "https://api.myjson.com/
bins/hqys2";

fetch(url).then(function (res) {
    return res.json();
}).then(function (data) {
    console.log(data); }).catch
(function (error) {
    console.log(error);
});
```

```
const url =
    "https://api.myjson.com/bins/hqys2";
fetch(url).then(res => {
    return res.json();
}).then( data => {
    console.log(data);
}).catch(error => {
    console.log(error);
})
```

# **JavaScript Fetch Errors**





```
const url = "https://randomuser.me/api/?  
    results=5";  
fetch(url).then(function(res){  
    return res.json()  
}).then(function(data){  
    data.results.forEach(function(person){  
        console.log(person);  
    })  
    console.log(data);  
})
```

## ***JSON with Multiple Items***



# **JSON with**

## **Multiple**

### **Items**

```
const output =  
    document.getElementById("output");  
const url = "https://randomuser.me/api/?  
    results=5";  
fetch(url).then(function(res){  
    return res.json()  
}).then(function(data){  
    data.results.forEach(function(person){  
        console.log(person.name.first);  
        console.log(person.picture.thumbnail);  
        output.innerHTML += person.name.first + "  
        ";  
        output.innerHTML += "<img  
            src='"+person.picture.thumbnail+">  
            <br>";  
    })
```





**The end**

