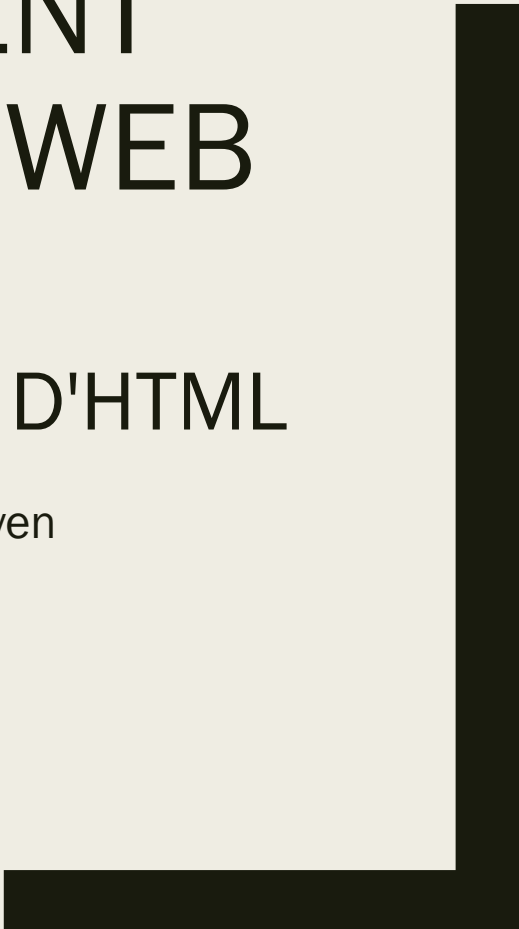




DÉVELOPPEMENT D'APPLICATIONS WEB

COURS 8: JSON

Enseignante: NGUYEN Thi Minh Tuyen



JSON

- JSON: JavaScript Object Notation.
- Une syntaxe pour stocker et échanger des données.
- Un texte écrit avec la notation objet JavaScript.
 - *Les données sont en paires nom / valeur*
 - *Les données sont séparées par des virgules*
 - *Les accolades tenir des objets*
 - *Les crochets détiennent des tableaux*

JSON données

- Écrit sous forme de paires nom/valeur.
- Exemple:

`"name" : "John"`

JSON - Évalue à des objets JavaScript

- Le format JSON est presque identique aux objets JavaScript.
- En JSON, les clés doivent être des chaînes, écrites avec des guillemets doubles. Exemple:

```
{ "name": "John" }
```

- En JavaScript, les touches peuvent être des chaînes, des nombres ou des noms d'identification. Exemple:

```
{ name: "John" }
```

Types de données valides [1]

En JSON, les valeurs doivent être l'un des types de données suivants:

- un string

```
{ "name": "John" }
```

- un numéro

```
{ "age": 30 }
```

- un objet (objet JSON)

```
{  
  "employee": { "name": "John",  
                "age": 30,  
                "city": "New York" }  
}
```

Types de données valides [2]

En JSON, les valeurs doivent être l'un des types de données suivants:

- un tableau

```
{  
    "employees": [ "John", "Anna", "Peter" ]  
}
```

- un booléen

```
{ "sale":true }
```

- nul

```
{ "middlename":null }
```

JSON.parse()

- Une utilisation courante de JSON est d'échanger des données vers / depuis un serveur web.
- Lors de la réception des données à partir d'un serveur Web, les données sont toujours une chaîne.
- Analyser les données avec `JSON.parse()`, et les données devient un objet JavaScript.
- Exemple:

```
JSON: '{ "name":"John", "age":30, "city":"New York"}'  
var obj = JSON.parse('{ "name":"John", "age":30,  
"city":"New York"}');
```

JSON.stringify() [1]

- Lors de l'envoi des données à un serveur Web, les données doit être une chaîne.
- Conversion d' un objet JavaScript dans une chaîne `JSON.stringify()`.
- Exemple:

```
var obj = { name: "John", age: 30, city: "New York" };  
var myJSON = JSON.stringify(obj);  
document.getElementById("demo").innerHTML = myJSON;
```


JSON.stringify() [2]

```
var arr = [ "John", "Peter", "Sally", "Jane" ];  
var myJSON = JSON.stringify(arr);  
document.getElementById("demo").innerHTML = myJSON;
```

```
var obj = { name: "John", today: new Date(), city  
: "New York" };  
var myJSON = JSON.stringify(obj);  
  
document.getElementById("demo").innerHTML = myJSON;
```

```
var obj = { name: "John", age: function ()  
{return 30;}, city: "New York"};  
var myJSON = JSON.stringify(obj);  
  
document.getElementById("demo").innerHTML = myJSON;
```

Exemple

```
var obj = { name: "John",  
            age: function () {return 30;},  
            city: "New York" };  
  
obj.age = obj.age.toString();  
  
var myJSON = JSON.stringify(obj);  
  
document.getElementById("demo").innerHTML = myJSON;
```

JSON objets

- Entourés par des accolades {}.
- Écrits par paires **clé/valeur**.
 - *Les clés doivent être des chaînes, et les valeurs doivent être un type de données JSON valide (chaîne, nombre, objet, tableau, booléen ou nulle).*
 - *Les clés et les valeurs sont séparées par deux points.*
 - *Chaque paire clé / valeur est séparée par une virgule.*
- Exemple:

```
{ "name":"John", "age":30, "car":null }
```

Accès aux valeurs d'objet

- Accéder aux valeurs d'objet en utilisant la notation de point (.).

- Exemple:

```
myObj = { "name":"John", "age":30, "car":null };  
name = myObj.name;  
age = myObj["age"];
```

Emboîter les objets JSON

- Les valeurs dans un objet JSON peut être un autre objet JSON.

```
myObj = {  
  "name": "John",  
  "age": 30,  
  "cars": {  
    "car1": "Ford",  
    "car2": "BMW",  
    "car3": "Fiat"  
  }  
}
```

Accéder à des objets JSON imbriqués

```
x = myObj.cars.car2;  
// ou :  
x = myObj.cars["car2"];
```

Modification des valeurs

```
myObj.cars.car2 = "Mercedes";
```

```
myObj.cars["car2"] = "Mercedes";
```

Tableaux en JSON

- Sont presque les mêmes que les tableaux en JavaScript.
- Les valeurs de tableau doit être de type chaîne, nombre, objet, tableau booléen ou nulle .
- Exemple: ["Ford", "BMW", "Fiat"]

{

 "name":"John",

 "age":30,

 "cars":["Ford", "BMW", "Fiat"]

}

Tableaux emboîtés dans des objets JSON

```
myObj = {  
  "name": "John",  
  "age": 30,  
  "cars": [  
    { "name": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },  
    { "name": "BMW", "models": [ "320", "X3", "X5" ] },  
    { "name": "Fiat", "models": [ "500", "Panda" ] }  
  ]  
}
```

Exemple

- Créez un page web qui permet de chercher la définition à partir d'un mot-clé entré par l'utilisateur.
- Consigne: Chargez fichier `dictionary.json` et chercher la définition d'un mot-clé.

Question ?