Introduction to Web Technology

JSON

Joseph Tonien
School of Computing and Information Technology
University of Wollongong

JavaScript: Array

```
var arrayName = [item0, item1, ...];
var emptyArray = [];
```

```
var subjectList = ["ISIT206", "MATH121", "CSCI301"];
subjectList[1] = "LOGIC101";
subjectList[6] = "LAW201"; // this will create holes in array

// loop through an array

for(var i = 0; i < subjectList.length; i++){
  alert(subjectList[i]);
}
```

JavaScript: Object

Object is defined by a list of `property:value`

```
var objectName = {property1:value1, property2:value2, ...};
```

```
var emptyObject = {};
```

```
var info = {
  name: "John",
  dob: new Date("1996-01-20"),
  year: 2
};
```

Object values can be obtained by **two ways**:

```
obj.property
```

```
obj["property"]
```

```
// two ways:
info.year
info["year"]
```

JavaScript: Array vs Object

```
var arrayName = [item0, item1, ...];
```

```
var objectName = {property1:value1, property2:value2, ...};
```

Arrays use numbered index:

```
arrayName[0] = "LOGIC101";
arrayName[1] = "CSCI111";
```

Objects use named index:

```
objectName["firstName"] = "John";
objectName.lastName = "Lee";
```

JavaScript Object Notation (JSON)

- In most web applications, XML and JSON are used to store or transport data
- JSON is "self-describing" and easy to understand

This is an example of a JSON describing a student object:

```
{
  "fullname": "John Smith",
  "studentNumber": "U1234567",
  "age": 20,
  "csMajor": true
}
```

JSON

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects

```
{
  "fullname": "John Smith",
  "studentNumber": "U1234567",
  "age": 20,
  "csMajor": true
}
```

JSON

Square brackets hold arrays

```
[
  {
    "firstName":"John",
    "lastName":"Smith"
  },
  {
    "firstName":"Kate",
    "lastName":"Williams"
  }
]
```

JSON

- Curly braces hold objects
- Square brackets hold arrays

```
{
  "firstName":"John",
  "lastName":"Smith",
  "subjectList":[
    {
      "code":"MATH101",
      "title":"Algebra"
    },
    {
      "code":"CSIT122",
      "title":"C programming"
    }
  ]
}
```

JSON

Translate from Javascript object to JSON string

```
objJSON = JSON.stringify(obj);
```

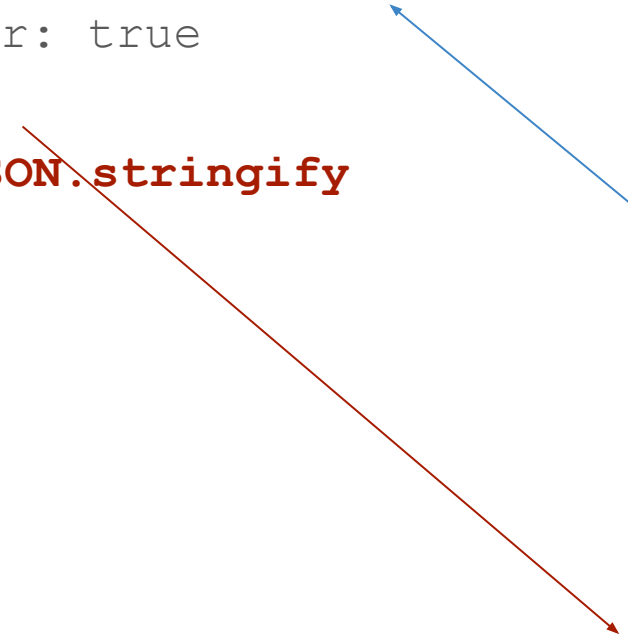
Translate from JSON string to javascript object

```
obj = JSON.parse(objJSON);
```

JSON

OBJECT

```
{
  fullname: "John Smith",
  studentNumber: "U1234567",
  age: 20,
  csMajor: true
}
```



JSON

```
{
  "fullname": "John Smith",
  "studentNumber": "U1234567",
  "age": 20,
  "csMajor": true
}
```

JSON.stringify

The **`JSON.stringify`** method converts a JavaScript value to a JSON string.

```
Syntax:  JSON.stringify(jsvalue, replacer, space)
```

- `jsvalue:` the javascript value to convert to a JSON string.
- `replacer` (Optional): selecting/filtering which properties of the object to be included in the JSON string. If the `replacer` is null or not provided, all properties of the object are included in the resulting JSON string.
- `space` (Optional): use for indentation, specifying white spaces in the output JSON string for readability purposes.

JSON.stringify function demo

Enter information to construct a student object:

Full name: John Smith

Student number: U1234567

Age: 20

CompSci major: ☐

Click View buttons to see JSON string of the student object.

View `JSON.stringify(studentObj)`

```
{"fullname":"John Smith","studentNumber":"U1234567","age":20,"csMajor":false}
```

View `JSON.stringify(studentObj, null, 2)`

```
{
  "fullname": "John Smith",
  "studentNumber": "U1234567",
  "age": 20,
  "csMajor": false
}
```

View `JSON.stringify(studentObj, ["studentNumber", "csMajor"]);`

```
{"studentNumber":"U1234567","csMajor":false}
```

View `JSON.stringify(studentObj, ["studentNumber", "csMajor"], 2)`

```
{
    "studentNumber": "U1234567",
    "csMajor": false
}
```

JSON.stringify

```
var studentObj = {
  fullname: "John Smith",
  studentNumber: "U1234567",
  age: 20,
  csMajor: false
};
```


JSON.stringify(studentObj)

```
{"fullname":"John Smith","studentNumber":"U1234567","age":20,
"csMajor":false}
```

output JSON sticks together
make it hard to read

JSON.stringify

```
var studentObj = {
  fullname: "John Smith",
  studentNumber: "U1234567",
  age: 20,
  csMajor: false
};
```



```
JSON.stringify(studentObj, null, 2)
```

using 2 spaces indentation

```
{
  "fullname": "John Smith",
  "studentNumber": "U1234567",
  "age": 20,
  "csMajor": false
}
```

JSON.stringify

```
var studentObj = {
  fullname: "John Smith",
  studentNumber: "U1234567",
  age: 20,
  csMajor: false
};
```



```
JSON.stringify(studentObj, ["studentNumber", "csMajor"])
```

only output the student number and compsci major

```
{"studentNumber":"U1234567","csMajor":false}
```

JSON.stringify

```
var studentObj = {
  fullname: "John Smith",
  studentNumber: "U1234567",
  age: 20,
  csMajor: false
};
```



```
JSON.stringify(studentObj, ["studentNumber", "csMajor"], 2)
```

only output the student number and compsci major, using 2 spaces indentation

```
{
  "studentNumber": "U1234567",
  "csMajor": false
}
```

Example 1: JSON.stringify

```
function showObjectJSON(){
  //create a student object
  var studentObj = {};
  studentObj.fullname = "John Smith";
  studentObj.studentNumber = "U1234567";
  studentObj.age = 20;
  studentObj.csMajor = true;

  //get JSON string from the javascript object
  var studentJSON = JSON.stringify(studentObj);

  //print the JSON string to the console
  console.log(studentJSON);
}

<button onClick="showObjectJSON()">
Click here to see JSON string
</button>
```

Example 2: JSON.parse

```
function showObject(){
 //JSON string
 var studentJSON = '{"fullname":"John Smith","studentNumber":
"U1234567","age":20,"csMajor":true}';

 //get javascript object from JSON string
 var studentObj = JSON.parse(studentJSON);

 //print the object to the console
 console.log(studentObj);
 console.log("Full name is " + studentObj.fullname);
}


<button onClick="showObject()">
Click here to see object from JSON
</button>
```

Example 3: JSON.stringify

```
function showArrayJSON(){
  var user1 = {};
  user1.firstName = "John";
  user1.lastName = "Smith";

  var user2 = {};
  user2.firstName = "Kate";
  user2.lastName = "Williams";

  //create an array of user objects
  var userList = [user1, user2];

  //get JSON string from the javascript array
  var userListJSON = JSON.stringify(userList);

  //print the JSON string to the console
  console.log(userListJSON);
}
```

```
<button onClick="showArrayJSON()">
Click here to see JSON string
</button>
```

Example 4: JSON.parse

```
function showArray(){
 //JSON string
 var userListJSON = '[{"firstName":"John","lastName":"Smith"},
                     {"firstName":"Kate","lastName":"Williams"}]';

 //get javascript array from JSON string
 var userList = JSON.parse(userListJSON);

 //print the object to the console
 console.log(userList);
 console.log("There are " + userList.length + " users");
}
```

```
<button onClick="showArray()">
Click here to see array from JSON
</button>
```

Example 5: JSON.stringify

```
function showObjectJSON(){
  var studentObj = {}; //create a student object
  studentObj.firstName = "John";
  studentObj.lastName = "Smith";
  studentObj.subjectList = []; //empty array to hold subjects

  var subjectObj1 = {};
  subjectObj1.code = "MATH101";
  subjectObj1.title = "Algebra";
  //add subject into array
  studentObj.subjectList.push(subjectObj1);

  var subjectObj2 = {};
  subjectObj2.code = "CSIT122";
  subjectObj2.title = "C programming";
  //add subject into array
  studentObj.subjectList.push(subjectObj2);

  //get JSON string from obj and print it on console
  var studentJSON = JSON.stringify(studentObj, null, 2);
  console.log(studentJSON);
}
```

Example 5: JSON.stringify

```
{
  "firstName":"John",
  "lastName":"Smith",
  "subjectList":[
    {
      "code":"MATH101",
      "title":"Algebra"
    },
    {
      "code":"CSIT122",
      "title":"C programming"
    }
  ]
}
```

References

- http://www.w3schools.com/json

- https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON