JSON, AJAX

ITP 301
Spring 2021

JavaScript Object Notation

Java**S**cript **O**bject **N**otation (JSON):

- Lightweight data-exchange format,
- Human-readable,
- Derived from JavaScript,
- Language-independent.

Common Uses:

- Front-end ↔ back-end communication,
- Web server ↔ web application exchange.

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}
```

JSON vs. JS Objects

Key differences:

JSON	JS Objects
Language-independent	JavaScript only

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}
```

```
var person = {
  firstName: 'Tommy',
  lastName: 'Trojan',
  email : 'ttrojan@usc.edu',
  phone: {
    cell: '123-456-7890',
    home: '321-654-0987',
  },
  hobbies: ['Web Dev', 'Bruin Hunting'],
  intro: function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
  },
}
```

JSON vs. JS Objects

Key differences:

JSON	JS Objects
Language-independent	JavaScript only
Double quotes for keys and values (except numeric)	Any quotes only for string values

JSON vs. JS Objects

Key differences:

JSON	JS Objects
Language-independent	JavaScript only
Double quotes for keys and values (except numeric)	Any quotes only for string values
Cannot contain methods	Can contain JS methods

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}
```

```
var person = {
  firstName: 'Tommy',
  lastName: 'Trojan',
  email : 'ttrojan@usc.edu',
  phone: {
    cell: '123-456-7890',
    home: '321-654-0987',
  },
  hobbies: ['Web Dev', 'Bruin Hunting'],
  intro: function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
  },
}
```

JSON vs. JS Objects

Key differences:

JSON	JS Objects
Language-independent	JavaScript only
Double quotes for keys and values (except numeric)	Any quotes only for string values
Cannot contain methods	Can contain JS methods
No comma after last key:value pair	Comma after last key:value pair is okay

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}
```

```
var person = {
  firstName: 'Tommy',
  lastName: 'Trojan',
  email : 'ttrojan@usc.edu',
  phone: {
    cell: '123-456-7890',
    home: '321-654-0987',
  },
  hobbies: ['Web Dev', 'Bruin Hunting'],
  intro: function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
  },
}
```

Accessing JSON Objects

Same syntax as with JS Objects.

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}

 person.firstName
```

Accessing JSON Objects

Same syntax as with JS Objects.

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	Key: cell → 123-456-7890; Key: home → 321-654-0987
hobbies	Web Dev, Bruin Hunting

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}

person.firstName

person.phone
```

Accessing JSON Objects

Same syntax as with JS Objects.

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	Key: cell → 123-456-7890; Key: home → 321-654-0987
hobbies	Web Dev, Bruin Hunting

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}

person.firstName

person.phone

person.phone.cell
```

Accessing JSON Objects

Same syntax as with JS Objects.

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	Key: cell → 123-456-7890; Key: home → 321-654-0987
hobbies	Web Dev, Bruin Hunting

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}

person.firstName

person.phone

person.phone.cell

person.hobbies
```

Accessing JSON Objects

Same syntax as with JS Objects.

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>

```
var person = {
  "firstName": "Tommy",
  "lastName": "Trojan",
  "email ": "ttrojan@usc.edu",
  "phone": {
    "cell": "123-456-7890",
    "home": "321-654-0987"
  },
  "hobbies": ["Web Dev", "Bruin Hunting"]
}
```

```
person.firstName
```

```
person.phone
```

```
person.phone.cell
```

```
person.hobbies
```

```
person.hobbies[0]
```

Callback Function

Functions that are executed at a later time.

Typically passed to other functions or objects as arguments.

Frequently used with asynchronous JS.

Callback Example 1

1. `userGreeting` receives two parameters:
    a. variable `name`,
    b. callback function `callbackFunction`.
2. Save user input in variable `username`.
3. Invoke userGreeting function.
    a. Pass `username` variable,
    b. Pass anonymous callback function.
4. `userGreeting` invokes anonymous callback function when ready.

```
function userGreeting(name, callbackFunction){
  // Do asynchronous stuff.
  callbackFunction(name);
}


var username = prompt('Enter your name: ');

userGreeting(username, function(result){
  console.log(result);
});
```

Callback Example 2

1. `userGreeting` receives two parameters:
    a. variable `name`,
    b. callback function `callbackFunction`.
2. Save user input in variable `username`.
3. Invoke userGreeting function.
    a. Pass `username` variable,
    b. Pass `myCallback` callback function.
4. `userGreeting` invokes `myCallback` callback function when ready.

```
function userGreeting(name, callbackFunction){
  // Do asynchronous stuff.
  callbackFunction(name);
}


var username = prompt('Enter your name: ');


userGreeting(username, myCallback);


function myCallback(results){
  console.log( results );
}
```

AJAX

Asynchronous **J**avaScript **A**nd **X**ML.

Technique used to make asynchronous requests to the server.

jQuery's AJAX is one of the most popular implementations.

```
<script src="http://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

  $.ajax({
    url: API_ENDPOINT,
    dataType: "json",
    success: function(data){
      doSomething(data);
    },
    error: function(e){
      alert("AJAX Error");
      console.log(e);
    },
  });

</script>
```