

JSON

Author

What is JSON?

JavaScript Object Notation

Serialized JavaScript Objects

Stricter rules

No functions

<http://json.org/>

Why use JSON?

JSON is native to JavaScript

Supported in most languages

AJAX requests & responses

Storing data

value in localStorage key/value pair

in backend database

AJAX?

Originally

Asynchronous JavaScript and XML

But now (mostly)

Asynchronous JavaScript and JSON

Because

JSON is easier to parse than XML

XML Attributes

```
<?xml version="1.0" encoding="utf-8">  
<my-object  
  xmlns="urn:example:my-object"  
  name="my object"  
  is_mine="true"  
  current_age="1" />
```

XML Elements

```
<?xml version="1.0" encoding="utf-8">  
<my-object xmlns="urn:example:my-object">  
  <name>my object</name>  
  <is_mine>true</is_mine>  
  <current_age>1</current_age>  
</my-object>
```

XML Problems

Can use without schema...

But hard to understand structure:

Are multiple **name** elements allowed?

What is the type of **current_age**?

Namespaces

JS Object Literal

```
var myObject = {  
  name: "my object",  
  is_mine: true,  
  current_age: 1,  
  age: function () {  
    this.current_age++;  
  }  
};
```


JSON

```
> JSON.stringify(myObject);
```

```
'{"name": "my object",  
  "is_mine": true, "current_age": 1}'
```

```
> JSON.stringify(myObject, null, 2);
```

```
'{\n  "name": "my object",\n  "is_mine": true,\n  "current_age":  
1\n}'
```

Strings

```
> var foo = JSON.parse(' "Hi!" ')
```

```
undefined
```

```
> foo
```

```
'Hi!'
```

```
> typeof foo
```

```
'string'
```

Numbers

```
> var foo = JSON.parse('3.14')
```

```
undefined
```

```
> foo
```

```
3.14
```

```
> typeof foo
```

```
'number'
```

Booleans

```
> var foo = JSON.parse('true')
```

```
undefined
```

```
> foo
```

```
true
```

```
> typeof foo
```

```
'boolean'
```

```
> typeof JSON.parse('"true"')
```

```
'string'
```

Arrays

```
> var foo =  
JSON.parse('["statler","waldorf"]');  
undefined  
> foo  
[ 'statler', 'waldorf' ]  
> typeof foo  
'object'  
> typeof foo[0]  
'string'
```

Objects

```
> var foo = JSON.parse( '{"pi":  
3.14}' )
```

```
undefined
```

```
> foo
```

```
{ pi: 3.14 }
```

```
> typeof foo
```

```
'object'
```

```
> typeof foo.pi
```

```
'number'
```

Functions

```
> var foo = JSON.stringify(  
  {"1":function(){return 1;}})
```

```
undefined
```

```
> foo
```

```
' {} '
```

RegExps

```
> var foo = JSON.stringify(  
  {re:/abc/})
```

```
undefined
```

```
> foo
```

```
'{"re":{}}'
```


Modifying Stringify

```
> var re = { re: /abc/,  
... toJSON: function () {  
... return { re: this.re.source } } },  
... foo = JSON.stringify(re);
```

undefined

```
> foo
```

```
'{"re": "abc"}'
```

```
> typeof JSON.parse(foo).re
```

```
'string'
```

Pretty Stringify

```
> var pi = { pi: 3.14, kansas: 4.0 },  
... json = JSON.stringify(pi),  
... pretty = JSON.stringify(pi, null, 4)  
> console.log(json)  
{"pi":3.14,"kansas":4}  
> console.log(pretty)  
{  
    "pi": 3.14,  
    "kansas": 4  
}
```

Partial Stringify

```
> var o = { a: 1, b: 2, c: 3 },  
... gt1 = function (k, v) {  
...     return (  
...         typeof v === 'number'  
...         && v < 2  
...     ) ? undefined : v; }  
> JSON.stringify(o, gt1)  
'{"b":2,"c":3}'
```

JSON Object Keys

Must be enclosed in double quotes

Can contain (almost) any characters

Sometimes not easily supported in
other languages

Yeah, Java, I'm looking at you

Dynamic Languages

JavaScript

`JSON.stringify, JSON.parse`

PHP

`json_encode, json_decode`

Ruby

`require 'json'`

`json.generate, json.parse`

PHP

```
$myobject = array(  
    "name"           => "my object",  
    "current_age"    => 1,  
    "is_mine"        => true  
);  
echo json_encode($myobject);  
  
{"name": "my object", "current_age":  
1, "is_mine": true}
```

Ruby

```
require 'json'
print JSON.generate({
  "name"           => "my_object",
  "current_age"    => 1,
  "is_mine"        => true
})
```

```
{"name": "my_object", "current_age":
1, "is_mine": true}
```

Python

```
import json
print json.dumps({
    'name':          'my_object',
    'current_age': 2,
    'is_mine':       True
})
```

```
{"current_age": 2, "name":
"my_object", "is_mine": true}
```


JSON & Java

Extra Libraries

e.g. Jackson JSON Library

Can parse into Objects or Maps

Maps can be safer

arbitrary object keys (“foo!bar-baz”)

“less native”

<http://github.com/scottylogan/jackson-json-odd-properties>