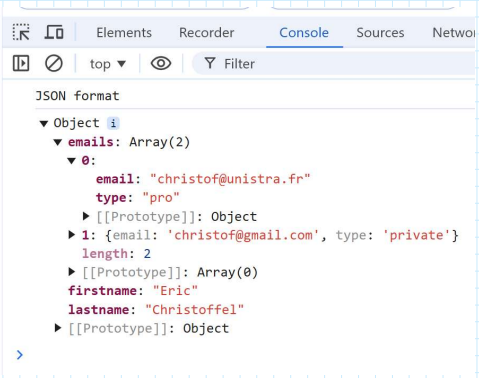
JSON Data Mining : properties and methods

1 JSON Structure

Example N1: exJSON_1.html

```
10  var contact={"lastname":"Christoffel","firstname":"Eric","emails":
    [{"email":"christof@unistra.fr","type":"pro"},{"email":"christof@gmail.
    com","type":"private"}]};
11
12  console.log("JSON format")
13  console.log(contact);
```

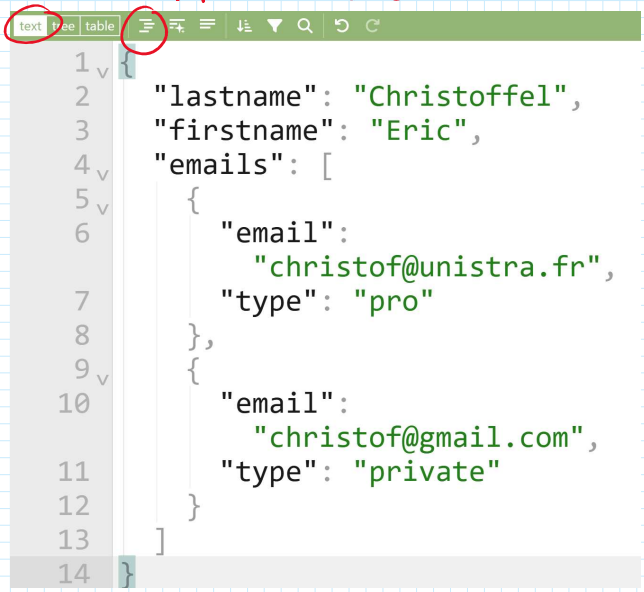
INSPECT Tool + console tab



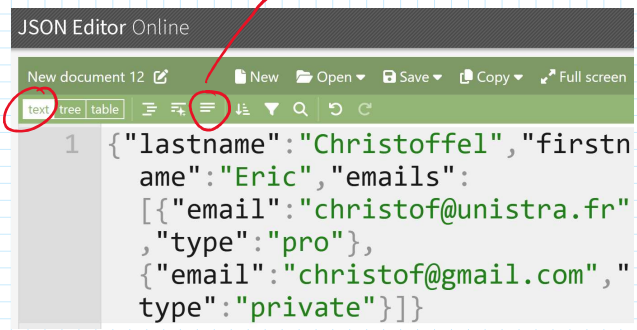
JSON Editor

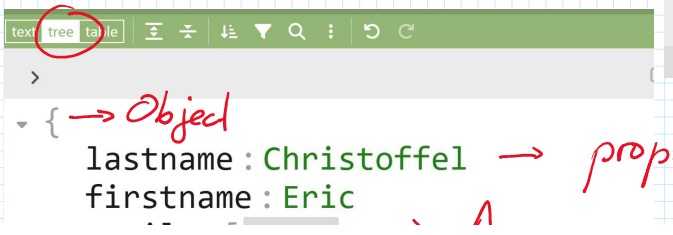
https://jsoneditoronline.org/

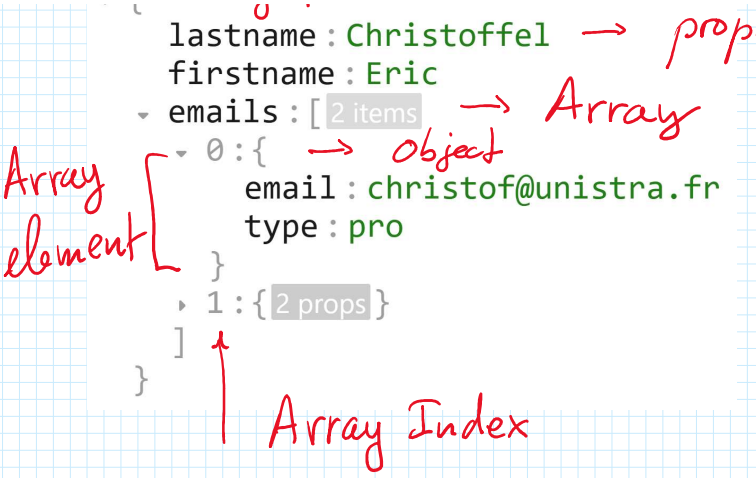
→ Format JSON + indentation



Compact JSON







② JSON parse(), stringify() methods

- Static methods parse() and stringify()
  ⇒ applied to the class, not to the object which is an instance of the class

(ex. today = new Date())

class → JSON . stringify( ) ← static method

exJSON_2.html

[JSON data as an argument

```
11  var contactString=JSON.stringify(contact);
12
13  console.log("JSON format stringified as a String :")
14  console.log(contactString);
15  console.log("is of type : "+typeof(contactString));
16  console.log("The previous JSON data  :");
17  console.log(contact);
```

type of operator

String

```
JSON format stringified as a String :                                    exJSON_2.html:13
{"lastname":"Christoffel","firstname":"Eric","emails":                   exJSON_2.html:14
[{"email":"christof@unistra.fr","type":"pro"},
{"email":"christof@gmail.com","type":"private"}]}
is of type : string                                                      exJSON_2.html:15
The previous JSON data  :                                                exJSON_2.html:16
▼ Object i                                                               exJSON_2.html:17
  ▶ emails: (2) [{…}, {…}]
    firstname: "Eric"
    lastname: "Christoffel"
  ▶ [[Prototype]]: Object
```

exJSON_3.html JSON.parse( ) String as an argument

String

```
9  //JS String delimited with sigle quotes '
10 var contactString='{"lastname":"Christoffel","firstname":"Eric",
   "emails":[{"email":"christof@unistra.fr","type":"pro"},
   {"email":"christof@gmail.com","type":"private"}]}';
11
12 var contact=JSON.parse(contactString);
13
14 console.log("Initial String :");
15 console.log(contactString);
16 console.log("is of type : "+typeof(contactString));
17
18 console.log("JSON format parsed from a String :")
19 console.log(contact);
20 console.log("is of type : "+typeof(contact));
```

Delimiters

String

```
Initial String :                                              exJSON_3.html:14
{"lastname":"Christoffel","firstname":"Eric","emails":        exJSON_3.html:15
[{"email":"christof@unistra.fr","type":"pro"},
{"email":"christof@gmail.com","type":"private"}]}
is of type : string                                           exJSON_3.html:16
JSON format parsed from a String :                            exJSON_3.html:18
▼ Object i                                                    exJSON_3.html:19
  ▶ emails: (2) [{…}, {…}]
    firstname: "Eric"
    lastname: "Christoffel"
  ▶ [[Prototype]]: Object
is of type : object                                           exJSON_3.html:20
```

---

3 Reading JSON properties

- How to read a JSON property ?

a) dot notation

exJSON_4.html • dot notation object . property

dot

```
15 console.log("property lecture with the . dot
   notation : contact.lastname");
16 console.log(contact.lastname);
```

```
property lecture with the . dot notation : contact.lastname        exJSON_4.html:15
Christoffel                                                        exJSON_4.html:16
```

```
18  //if the property is an array
19  console.log(contact.emails[0].email);
```

property of the array element which is an object

index of the array element

The value of the property is an array

```
christof@unistra.fr                                                exJSON_4.html:19
```

5) Array notation

exJSON_5.html

array notation or bracket notation

object["property"]

array index

String or a variable

```
15  console.log("property lecture with the array notation : contact
    ['lastname']\nthe property is the array argument, index");
16  console.log(contact['lastname']);
17  console.log('if a property is an Array : contact.emails[1]["type"]');
18  console.log(contact.emails[1]["type"]);
```

```
property lecture with the array notation : contact['lastname']      exJSON_5.html:15
the property is the array argument, index
Christoffel                                                        exJSON_5.html:16
if a property is an Array : contact.emails[1]["type"]              exJSON_5.html:17
private                                                            exJSON_5.html:18
```

```
20  //using a variable
21  let propriete='firstname';
22  console.log("property lecture with the Array notation : contact
    [variable]");
23  console.log(contact[propriete]);
24  console.log(contact["firstname"]);
25
26
27  console.log('if a property is an Array : contact.emails[1].type');
28  console.log(contact.emails[0]['type']);
```

a variable

Index of the array

the property name

property lecture with the Array notation : contact[variable]	exJSON_5.html:22
Eric	exJSON_5.html:23
Eric	exJSON_5.html:24
if a property is an Array : contact.emails[1].type	exJSON_5.html:27
pro	exJSON_5.html:28

(4) Checking a property

- How to check if a JSON property (key) exists ?

exJSON_6.html object.hasOwnProperty() true/false

```
15  console.log("checking if a property exists using hasOwnProperty
    method");
16  console.log(" lastname exists ? "+contact.hasOwnProperty('lastname'));
17  console.log(" surname exists ? "+contact.hasOwnProperty('surname'));
```

checking if a property exists using hasOwnProperty method	exJSON_6.html:15
lastname exists ? true	exJSON_6.html:16
surname exists ? false	exJSON_6.html:17

(5) Displaying a list of properties and values

a) using a control statement:
for ... in loop

- How to display a list of properties ?

exJSON_7.html for ... in loop
=> loop over enumerable properties

```
15  console.log("list of properties, for...in loop\narray notation with a
    constant as an argument, index");
16  for(const property in contact){
17    console.log('Property name : '+property+' , value : '+contact
      [property]);
18  }
```

using Array notation : contact[property]

↑ Constant as an index

```
list of properties, for...in loop                                    exJSON_7.html:15
array notation with a constant as an argument, index
Property name : lastname , value : Christoffel                       exJSON_7.html:17
Property name : firstname , value : Eric                             exJSON_7.html:17
Property name : emails , value : [object Object],[object Object]     exJSON_7.html:17
```

because emails property has an Array value !

---

⑧ Returns a property list of an object as an array

exJSON_8.html

Object.keys()

Object (instance of a class) as an argument, or JSON object

class

static method

Returns an Array → Array.isArray()

↳ Returns true or false

```
15  console.log("keys() static method of the class Object, returns list of
    properties as an Array, confirmed with isArray() static method of Array
    class");
16  let propertyList=Object.keys(contact);
17  console.log("list of properties :");
18  console.log(propertyList);
19  console.table(propertyList);
```

Alternative to log() for Arrays

```
keys() static method of the class Object, returns list of properties   exJSON_8.html:15
as an Array, confirmed with isArray() static method of Array class
list of properties :                                                   exJSON_8.html:17
▼ (3) ['lastname', 'firstname', 'emails'] i                            exJSON_8.html:18
    0: "lastname"
    1: "firstname"
    2: "emails"
```

keys() returns a list of the object properties

```
0: "lastname"
1: "firstname"
2: "emails"
length: 3
```

keys() returns a list of the object properties

exJSON_8.html:19

(index)	Value
0	'lastname'
1	'firstname'
2	'emails'

Classe → méthode statique ← argument = objet

```
21  console.log("is an array ? "+Array.isArray(propertyList));
```

⇒ is propertylist an Array ?

```
is an array ? true                                    exJSON_8.html:21
```

---

(9) forEach() method on Array to iterate on each array element.

exJSON_9.html

object.forEach() method

[an iterable object :
- an array
- a node collection / html collection (see later)

```
12  console.log("JSON format")
13  console.log(contact.emails);
```

```
▼ Array(2) i
  ▶ 0: {email: 'christof@unistra.fr', type: 'pro'}
  ▶ 1: {email: 'christof@gmail.com', type: 'private'}
    length: 2
```

```
▼ 0:
    email: "christof@unistra.fr"
    type: "pro"
▼ 1:
    email: "christof@gmail.com"
    type: "private"
```

if → if is an Array

forEach() method

for ... in loop control Statement

```
15  console.log("forEach loop on an Array");
16
17  if(Array.isArray(contact.emails)){
18      contact.emails.forEach(function(itemEmail){
19          console.log("--------------------");
20          for(var property in itemEmail){
21              console.log("Property : "+property+", value : "+itemEmail
                [property]);
```

```
21      console.log("Property : "+property+", value : "+itemEmail
        [property]);
22    }
23  });
24  }
```

forEach method

Array Notation

```
forEach loop on an Array                                  exJSON_9.html:15
--------------------                                      exJSON_9.html:19
Property : email, value : christof@unistra.fr             exJSON_9.html:21
Property : type, value : pro                              exJSON_9.html:21
--------------------                                      exJSON_9.html:19
Property : email, value : christof@gmail.com              exJSON_9.html:21
Property : type, value : private                          exJSON_9.html:21
```

(10) Return an Array of 2 elements for each property : value pair

exJSON_10.html

Object.entries() returns a Map object, a list of key, value pairs

Object.entries()

Object (instance of a class) as an argument, or JSON object

class

Static method

for ... of loop

⇒ loop over iterable object: Array, Map

```
16  propertyValuePairs=Object.entries(contact);
17  console.log(propertyValuePairs);
```

2 array elements

```
▼ 0: Array(2)
    0: "lastname"
    1: "Christoffel"
    length: 2
```

```
Object.entries() returns a Map object                    exJSON_10.html:14
▼ (3) [Array(2), Array(2), Array(2)]                     exJSON_10.html:17
  ▶ 0: (2) ['lastname', 'Christoffel']
  ▶ 1: (2) ['firstname', 'Eric']
  ▶ 2: (2) ['emails', Array(2)]
    length: 3
```

Array

property : value pair

```
{
  lastname : Christoffel
  firstname : Eric
  emails : [2 items]
```

JSON Contact

```
firstname : Eric
emails : [2 items]
  0 : {
      email : christof@unistra.fr
      type : pro
    }
  1 : {
      email : christof@gmail.com
      type : private
    }
]
}
```

```
▼ 1: Array(2)
    0: "firstname"
    1: "Eric"
    length: 2
```

```
    }
  ]
}
```

```
▼ 1: Array(2)
  ▼ 0:
      email: "christof@unistra.fr"
      type: "pro"
    ▶ [[Prototype]]: Object
  ▶ 1: {email: 'christof@gmail.com', type: 'private'}
    length: 2
  ▶ [[Prototype]]: Array(0)
  length: 2
```

```
20  for([key,value] of propertyValuePairs){
21    console.log(`${key} : ${value}`);
22  }
```

```
lastname : Christoffel                                  exJSON_10.html:21
firstname : Eric                                        exJSON_10.html:21
emails : [object Object],[object Object]                exJSON_10.html:21
```