BERR2243

Database and Cloud System

Week 02:
Introduction to JS and JSON

Basic Javascript

- Variable Type:
  - `Number`
  - `Boolean`
  - `String`
  - `Array`
  - `Object`
  - `Function`
  - `Null`
  - `Undefined`

Basic Javascript

- Playground
  - Try to declare different types of variables

```
1   var num1 = 1;
2   var num2 = 3.142;
3
```

  - Try calling typeof to find out a variable's type
  - Try to compare the different between var and const declaration

```
3
4   const num3 = 5;
5
```

Array

```
53
54    var students = []; // empty array
55    var students = ["John", "Doe", "Alice", "Smith"];
56
57    |
```

- Playground:
  - Try print the array into the console
  - Try check the element one by one
  - Try change one of the element in the array
  - Try check the error message with index if beyond the array size

Array Methods

```
53
54    var students = []; // empty array
55    var students = ["John", "Doe", "Alice", "Smith"];
56
57    students.push("Brian");
58    students.unshift("Kelly");
59    students.pop();
60    students.shift();
61    students.sort();
62
```

- Playground
  - Try to console out the result for each method and understand their differences

Comments

```
// single-line comment

/* multi-line comment */
```

Null and Undefined

```
33
34    var ned = null;
35    var benson = 9;
36    // at this point in the code,
37    // ned is null
38    // benson's 9
39    // caroline is undefined
40
```

- `undefined` : has not been declared, does not exist
- `null` : exists, but was specifically assigned an empty or null value

Logical operators

- Greater Than: >
- Less Than: <
- Greater Than or Equal: >=
- Less Than or Equal: <=
- AND: &&
- OR: ||
- NOT: !
- Equal: ==
- Not Equal: !=
- Strict Not Equal: ===
- Strict No Equal: !==

Logical operators

- most logical operators automatically convert types:
  - 5 < "7" is true
  - 42 == 42.0 is true
  - "5.0" == 5 is true
- === and !== are strict equality tests; checks both type and value
  - "5.0" === 5 is false

Logical operators

```
1
2   var num1 = 1;
3   var num2 = 3.142;
4   var str1 = "Hello";
5   var str2 = "3.142";
6
7   console.log(num1 > 10)  // false
8   console.log(num2 < 5)   // true
9   console.log(str1 === "Hello")  // true
10  console.log(str2 !== "3.142")  // false
11
```

- Playground
  - Try yourself with other variables and logical operators

if/else statement

```
33
34  if (condition) {
35      statements;
36  } else if (condition) {
37      statements;
38  } else {
39      statements;
40  }
41
```

- identical structure to other programing language
- JavaScript allows almost anything as a condition

for loop

```
42
43  var sum = 0;
44  for (var i = 0; i < 100; i++) {
45      sum = sum + i;
46  }
47  console.log(sum);
48
```

```
39
40  var s1 = ["h", "e", "l", "l", "o"];
41  var s2 = "";
42  for (var i = 0; i < s1.length; i++) {
43      s2 += s1[i] + s1[i];
44  }
45  console.log(s2);
46
```

- Playground
  - Try to check the output from the console

forEach

```
47
48    var s1 = ["h", "e", "l", "l", "o"];
49    s1.forEach(element => {
50        console.log(element);
51    });
52
```

- The forEach method of Array instances executes a provided function once for each array element.

- Playground
  - Try to check the output from the console

while loop

```
57
58   while (condition) {
59       statements;
60   }
61
```

```
62
63   do {
64       statements;
65   } while (condition);
66
```

- JavaScript Object Notation (JSON)
  - light-weight
  - language independent
  - Easy to read and write
  - Text based, human readable data exchange format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28
}
```

JSON

- Standard JSON Format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28
}
```

JSON

- Standard JSON Format

```
{          key   :   value
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28
}
```

- Standard JSON Format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28
}
```

Data Separation

- Standard JSON Format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28,
   "phone" : {
        "mobile" : "0104001000",
        "office" : "6064442022"
    }
}
```

Sub-Document
(object list)

JSON

- Standard JSON Format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28,
   "phone" : [
      "0104001000",
      "6064442022"
   ]
}
```

Array Value
(array list)

- Standard JSON Format

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28,
   "phone" : [
      {"mobile" : "0104001000"},
      {"office" : "6064442022”}
    ]
}
```

Array of Key : Value
(array of object-list)

JSON

```
{
    "firstName" : "Chaitanya",
    "lastName" : "Singh",
    "age" :  28,
    "married" :  false,
    "phone" :  null
}
```

string

JSON Variable Types

```
{
    "firstName" : "Chaitanya",
    "lastName" : "Singh",
    "age" :  28,
    "married" :  false,
    "phone" :  null
}
```

number

JSON Variable Types

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28,
   "married" :  false,
   "phone" :  null
}
```

boolean

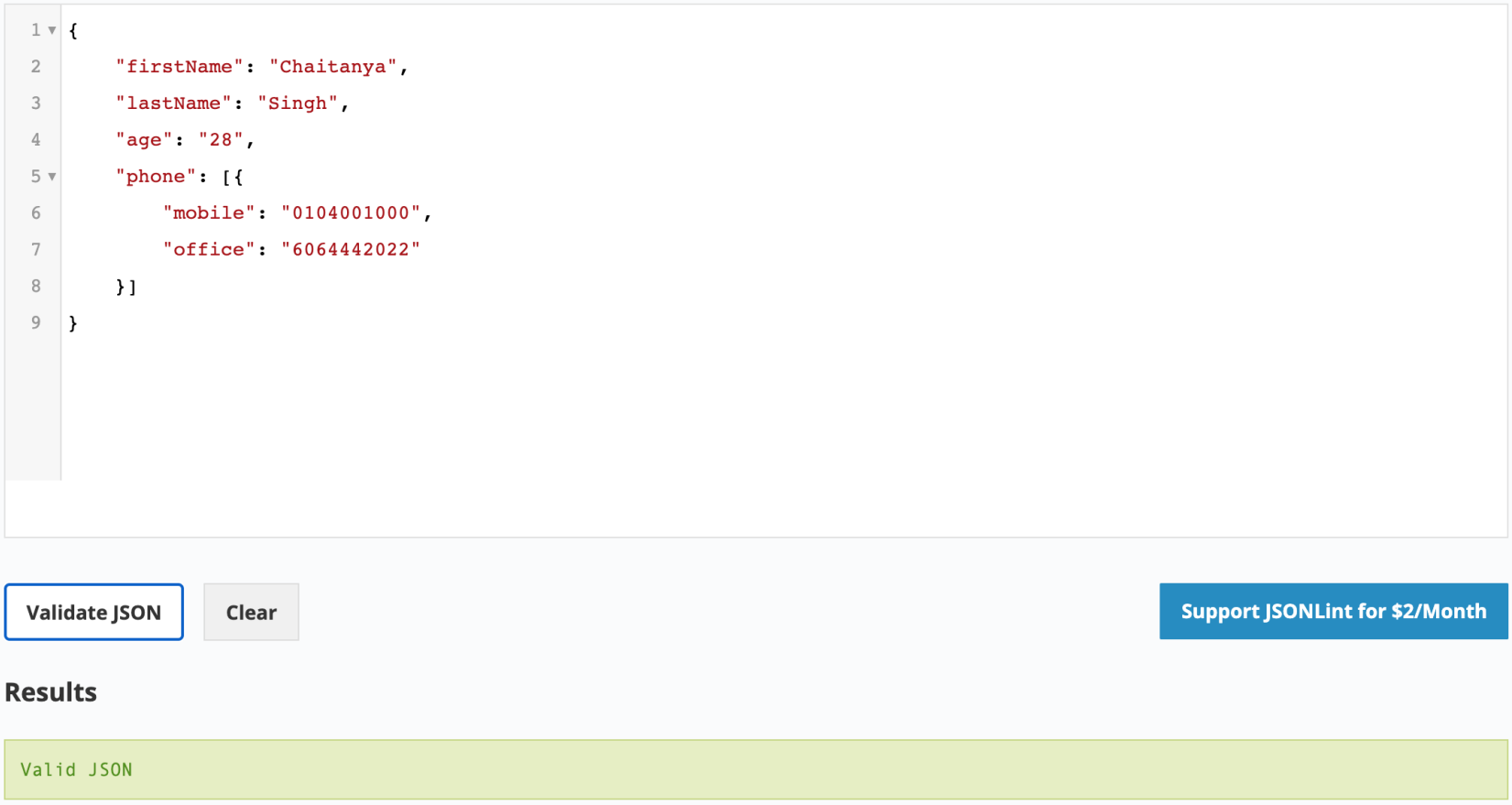
JSON Variable Types

```
{
   "firstName" : "Chaitanya",
   "lastName" : "Singh",
   "age" :  28,
   "married" :  false,
   "phone" :  null
}
```

NULL

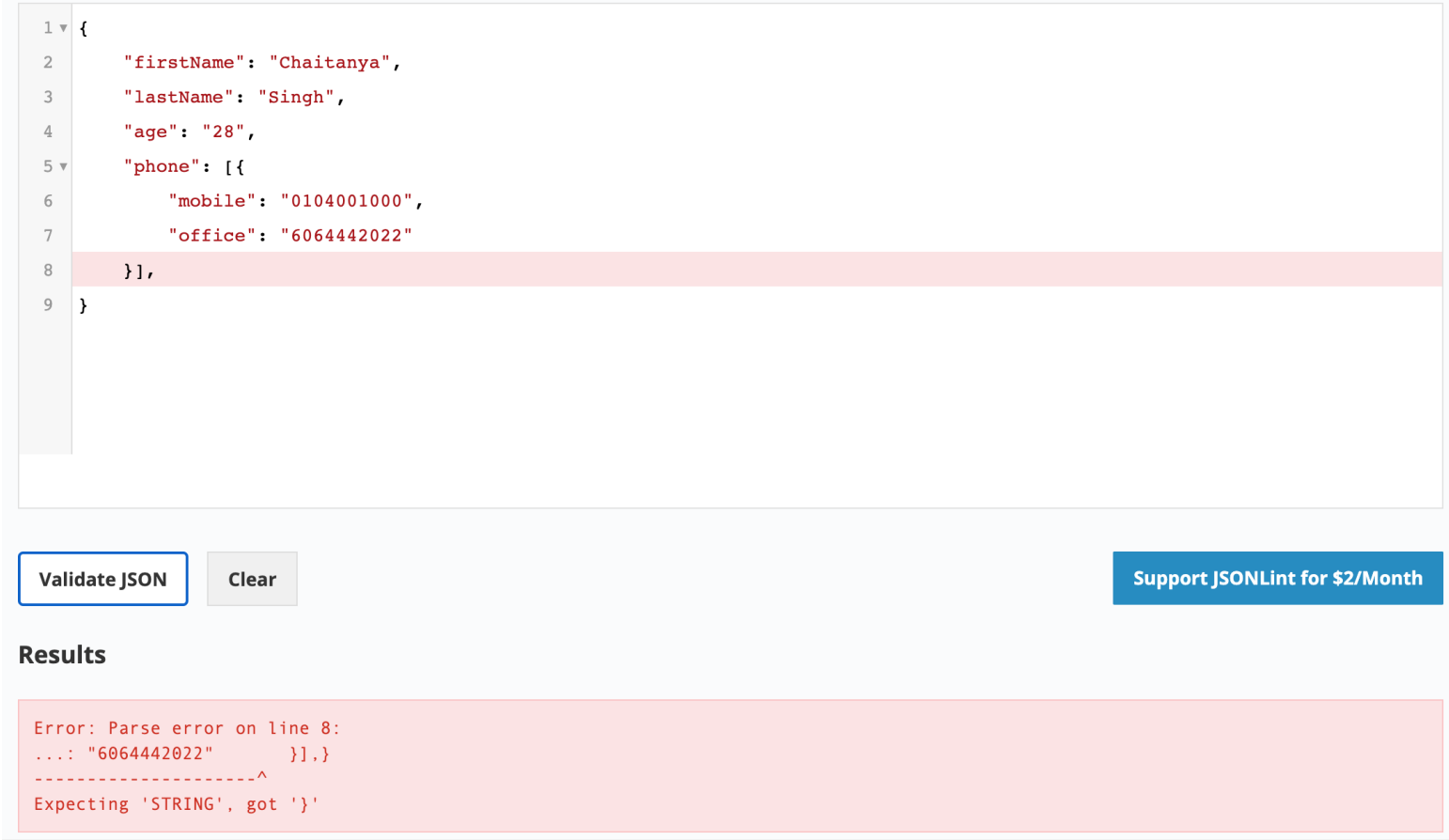
JSON Variable Types

- https://jsonlint.com/



JSON Validator

- https://jsonlint.com/



JSON Validator