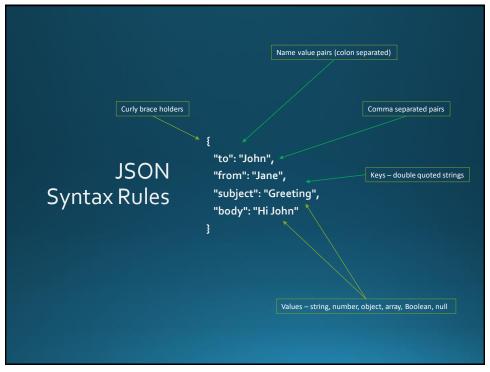


```
JSON

Example

| "to": "John",
    "from": "Jane",
    "subject": "Greeting",
    "body": "Hi John"
    }
```



```
JSON Files / Structure

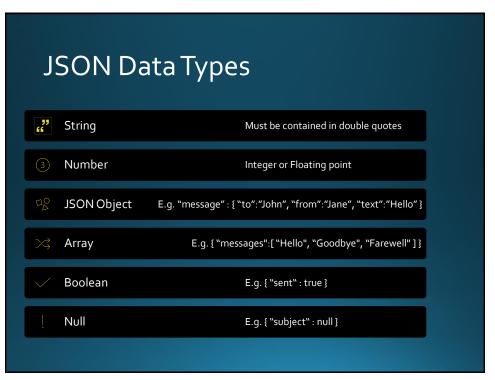
The file type for JSON files is

".json"

The Internet media type is

"application/json"

{
    "messages":[
    {
        "to": "John",
        "subject": "Greeting",
        "body": "Hi John"
    },
    {
        "to": "Mick",
        "from": "Joe",
        "subject": "Goodbye",
        "body": "See ya Mick"
    }
}
```



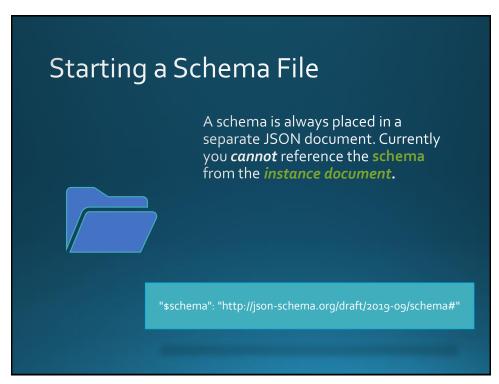
# JSON Schema

/

# JSON Schema

- JSON Schema is a vocabulary that allows you to annotate and validate JSON documents.
- > A JSON Schema is itself JSON.
- The JSON document being validated or described is called the instance, and the document containing the description is called the schema.
- Latest JSON Schema Specification draft update was December 2020





```
Length
                                             "type": "string",
                                               "minLength": 2,
JSON
                                               "maxLength": 3
Schema
                                 Regular
Expression
Types
String
                                              "type": "string",
                                               "pattern": "^(\\([o-9]{3}\\))?[o-9]{3}-[o-9]{4}$"
                                 Built-in
                                             Dates and Times
                                 Formats
                                             Email addresses
                                             IP addresses
                                             And others...
```

```
Integer

"type": "integer"

"type": "integer"

"type": "number"

"type": "number"

"type": "number",

"multipleOf":10

}

Range

{
"type": "number",

"multipleOf":10

}

Range

{
"type": "number",

"minimum":0,

"exclusiveMaximum":100

}
```

# JSON Schema Types Object

# **Properties**

key-value pairs

# **Required Properties**

Use of "required" keyword

### **Property names**

The names of properties can be validated against a schema irrespective of their values

# Size

Use of "minProperties" and "maxProperties"

# **Dependencies**

# Property dependencies

declare certain other properties must be present if a given property is present

### Schema dependencies

declare that the schema changes when a given property is present

13

# JSON Schema Types Array

### Items

Each element in an array may be of a different type

# List validation

a sequence of arbitrary length where each item matches the same schema

# Tuple validation

a sequence of fixed length where each item may have a different schema

### Length

Use of "minItems" and "maxItems" keywords

# Uniqueness

Use of the "uniqueltems" keyword e.g. array of numbers so that the same number is not repeated.

# JSON Schema Types Boolean & Null

{ "type": "boolean" }

Value of true or false (not "true" or "false")

{ "type": "null" }

Value of null (not "null" in double qoutes)
generally used to represent a missing value
– value must be null.

