

JSON and YAML

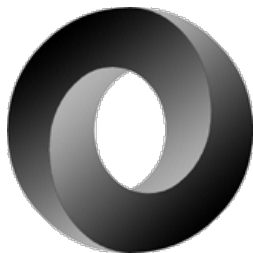
WASA: Web and Software Architecture

Prof. Emanuele Panizzi

JSON - JavaScript Object Notation

- Lightweight format to exchange data (e.g. data sent with RESTful requests and responses)
- Used also to store data (.json extension)
- Text format (based on Javascript subset)

```
{  
  "user": {  
    "firstName": "John",  
    "lastName": "Smith",  
    "age": 27  
  }  
}
```



- Json uses only two familiar concepts:
 - object
 - array
- easy for humans to read and write
- easy for machines to parse and generate

Object

Unordered collection of **name: value** pairs, enclosed in braces {}

- name:
 - string between ""
 - unique within the object
 - a.k.a. “key” or “field”
- value:
 - number, string, boolean, *null*, array, object

```
{  
  "WASA": {  
    "name": "Web and Software Architecture",  
    "semester": 1  
  }  
}
```

Ordered list of values, comma-separated, enclosed in brackets []

```
{  
  "wasaWeekdays": ["tuesday", "thursday"]  
}
```

JSON example

```
{  
  "anObject": {  
    "aNumber": 42,  
    "aString": "This is a string",  
    "aBoolean": true,  
    "nothing": null,  
    "anArray": [  
      1,  
      {"name": "value", "anotherName": 12},  
      "something else"  
    ]  
  }  
}
```

YAML - YAML Ain't Markup Language

- Human-friendly data serialization language
- for configuration files, or to store data (.yaml), or to exchange data
- Text format (based on Python indentation)

```
---
```

```
user:  
  firstName: John  
  lastName: Smith  
  age: 27
```



```
anObject:  
  aNumber: 42  
  aString: This is a string  
  aBoolean: true  
  nothing: null  
  anArray:  
    - 1  
    - anotherObject:  
      someName: some value  
      someOtherName: 1234  
    - something else
```


YAML superset of JSON

JSON files are valid YAML

- `{}` accepted for maps (objects)
- `[]` accepted for lists (arrays)
- `# Anything after a hash sign is a comment`
- documents can optionally begin with `---` and end with `...`
- a file may contain more documents

```
anotherArray: [1,2,3]
anotherObject: {"city": "Rome", "country": "Italy"}
aStringWithColon: "COVID-19: procedure di accesso"
aNumericString: "0649911"
aLongString: |
  this string spans
  more lines
```

- <https://www.json.org/json-en.html>
- <https://en.wikipedia.org/wiki/JSON>
- <https://jsonlint.com>
- <https://en.wikipedia.org/wiki/YAML>
- <https://yaml.org>
- <https://onlineyamltools.com/validate-yaml>
- https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html
- https://en.m.wikipedia.org/wiki/Recursive_acronym