The Carpentries Incubator



Metadata Schemas

Fundamentals of Scientific Metadata: Why Context Matters

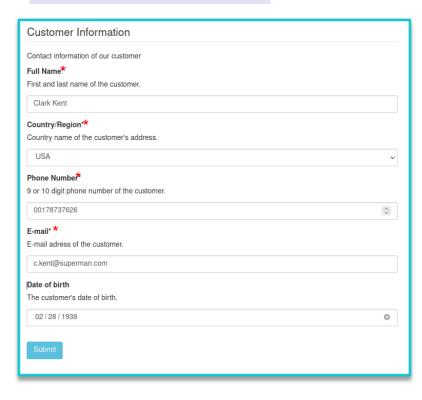
Metadata schemas



Metadata schemas express expectations in the structure of metadata records.

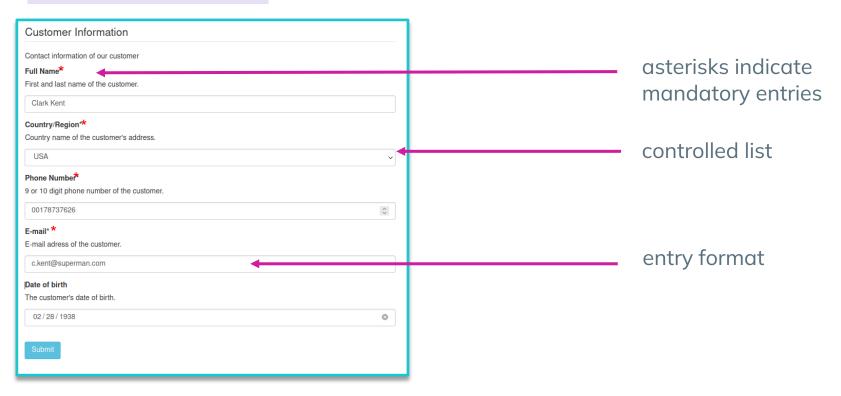


Customer Information



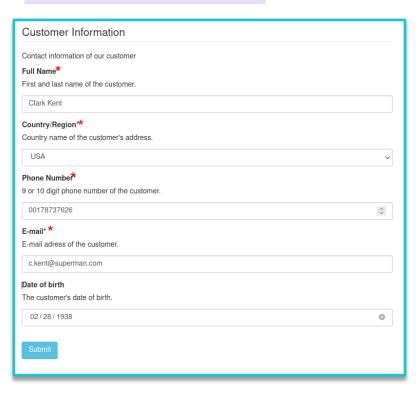


Customer Information





Customer Information

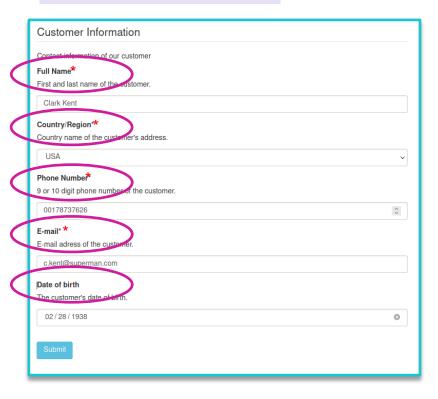


CustomerInformationKentC.json

```
"Full Name": "Clark Kent",
"Country/Region": "USA",
"Phone number": 00178737626,
"E-mail": "c.kent@superman.com",
"Date of birth": "1938-02-28"
}
```



Customer Information

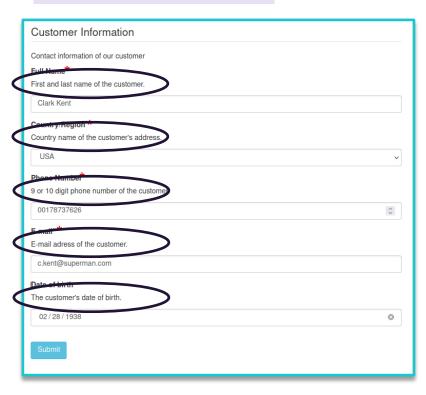


CustomerInformationKentC.json

```
"Full Name": "Clark Kent",
"Country/Region": "USA",
"Phone number": 00178737626,
"E-mail": "c.kent@superman.com",
"Date of birth": "1938-02-28"
}
```



Customer Information



CustomerInformationKentC.json

```
"Full Name": "Clark Kent",
"Country/Region": "USA",
"Phone number": 00178737626,
"E-mail": "c.kent@superman.com",
"Date of birth": "1938-02-28"
}
```

Benefits of schemas

A metadata schema is – basically – a set of **conventions or contraints**. [1]

Schemas are expressed in formal languages like XML, JSON or else so that (meta)data can be parsed and validated automatically according to the schema. [2]

^[1] https://www.merriam-webster.com/dictionary/schema, https://www.merriam-webster.com/dictionary/protocol

^[2] https://gitlab.hzdr.de/hmc/hmc/cct-7-semantics/hmc-glossary-initialization/-/blob/master/terms/schema.yaml (HMC CCT7, not yet ratified Image: Child plays with wooden shape sorter toy, https://unsplash.com/photos/ehaO7XywMGM

Writing schemas

XML Schemas (.xsd) are written in XML and used to describe & syntactically validate the structure of XML documents or (meta)data records. [1]

The JSON Schema vocabulary is used to describe & syntactically validate the structure of JSON (meta)data records. [2]

^[1] https://www.merriam-webster.com/dictionary/protocol

^[2] https://gitlab.hzdr.de/hmc/hmc/cct-7-semantics/hmc-glossary-initialization/-/blob/master/terms/schema.yaml (HMC CCT7, not yet ratified) Image: Child plays with wooden shape sorter toy, https://unsplash.com/photos/ehaO7XywMGM

Metadata schemas



We will **focus on writing JSON Schema** in our next hands-on task.



- JSON Schema version in \$schema
- list of required properties
- one required property
- one optional property
- data type constraints
- descriptions for the human reader

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"description": "In real life you would add a meaningful
             description here.",
"type": "object",
"required": [
  "superhero"
"properties": {
  "superhero": {
    "description": "A mandatory string property.",
    "type": "string"
  "power": {
    "description": "An optional numeric property.",
    "type": "integer"
```



- list of required properties
- one required propaerty
- one optional property
- data type constraints
- descriptions for the human reader

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"description": "In real life you would add a meaningful
             description here.",
"type": "object",
"required": [
  "superhero"
"properties": {
  "superhero": {
    "description": "A mandatory string property.",
    "type": "string"
  "power": {
    "description": "An optional numeric property.",
    "type": "integer"
```



- list of required properties
- one required propaerty
- one optional property
- data type constraints
- descriptions for the human reader

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"description": "In real life you would add a meaningful
             description here.",
"type": "object",
"required": [
  "superhero"
"properties": {
  "superhero": {
    "description": "A mandatory string property.",
    "type": "string"
  "power": {
    "description": "An optional numeric property.",
    "type": "integer"
```



- list of required properties
- one required propaerty
- one optional property
- data type constraints
- descriptions for the human reader

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"description": "In real life you would add a meaningful
             description here.",
"type": "object",
"required": [
  "superhero"
"properties": {
  "superhero": {
    "description": "A mandatory string property.",
    "type": "string"
  "power": {
    "description": "An optional numeric property.",
    "type": "integer"
```



- list of required properties
- one required propaerty
- one optional property
- data type constraints
- descriptions for the human reader

A JSON instance is syntactically valid, if it conforms to the definition described by the JSON schema.

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"description": "In real life you would add a meaningful
             description here.",
"type": "object",
"required": [
  "superhero"
"properties": {
  "superhero": {
    "description": "A mandatory string property.",
    "type": "string"
  "power": {
    "description": "An optional numeric property.",
    "type": "integer"
```

Validate JSON record



A JSON instance is syntactically valid, if it conforms to the definition described by the JSON schema.

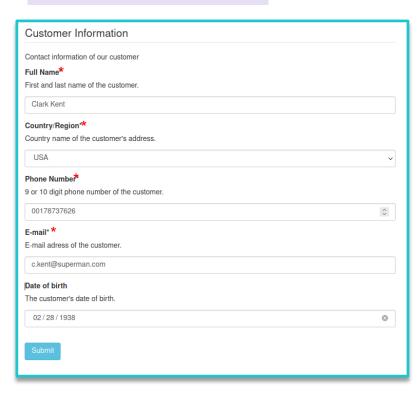
```
"superhero": "String Hero"

"superhero": 5
```





Customer Information



DISCLAIMER

This slide deck is part of the Lesson

<u>Fundamentals of Scientific Metadata:</u> <u>Why Context Matters</u>

published on **The Carpentries Incubator**.

Please cite this presentation as:

Gerlich, S., Strupp, A., Hofmann, V., Sandfeld, S. (2023). Fundamentals of Scientific Metadata: Why Context Matters. The Carpentries Incubator. DOI: <u>10.5281/zenodo.10091708</u>

You can find more information about this course on **Github**.



image:

https://c.pxhere.com/photos/35/f5/coffee_notebook_wooden_background_orange_work_table_office-1222115.jpg!d