Semi-structured vs. Structured

Semi-structured data

- XML
  - Extensible Markup Language (XML)
  - Popularized by web services utilizing SOAP principles.
    - Simple Object Access Protocol
- JSON
  - JavaScript Object Notation
  - Popularized by web services utilizing REST principles.
    - Representational State Transfer

Extensible Markup Language (XML)

- Standard for data representation and exchange
- Document format similar to HTML
  - Tags describe content instead of formatting
- Also streaming format

```
<?xml version="1.0" ?>
<!-- Bookstore with no DTD  -->
- <Bookstore>
  - <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <Title>A First Course in Database Systems</Title>
    - <Authors>
      - <Author>
          <First_Name>Jeffrey</First_Name>
          <Last_Name>Ullman</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jennifer</First_Name>
          <Last_Name>Widom</Last_Name>
        </Author>
      </Authors>
    </Book>
  - <Book ISBN="ISBN-0-13-815504-6" Price="100">
      <Remark>Buy this book bundled with "A First Course" -- a great deal!</Remark>
      <Title>Database Systems: The Complete Book</Title>
    - <Authors>
```

```
<?xml version="1.0" ?>
<!-- Bookstore with no DTD -->
- <Bookstore>
  - <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <Title>A First Course in Database Systems</Title>
    - <Authors>
      - <Author>
          <First_Name>Jeffrey</First_Name>
          <Last_Name>Ullman</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jennifer</First_Name>
          <Last_Name>Widom</Last_Name>
        </Author>
      </Authors>
    </Book>
  - <Book ISBN="ISBN-0-13-815504-6" Price="100">
      <Remark>Buy this book bundled with "A First Course" -- a great deal!</Remark>
      <Title>Database Systems: The Complete Book</Title>
    - <Authors>
      - <Author>
          <First_Name>Hector</First_Name>
          <Last_Name>Garcia-Molina</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jeffrey</First_Name>
          <Last_Name>Ullman</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jennifer</First_Name>
```

Basic constructs

- Tagged elements (nested)
- Attributes
- Text

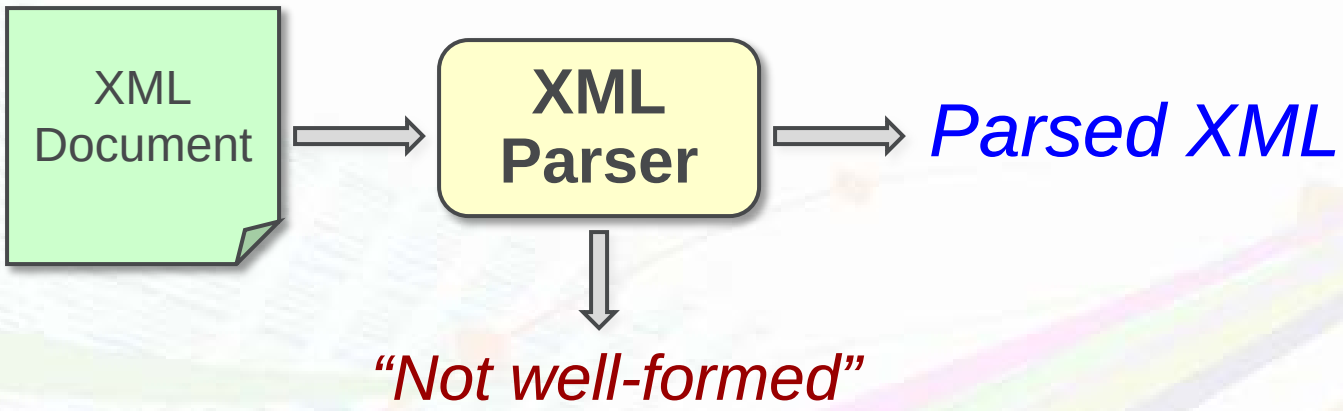
Relational Model versus XML

	Relational	XML
Structure	Tables	Hierarchy
Schema	Fixed	Flexible
Queries	Very Simple?	Simple?
Ordering	No.	Implied

“Well-Formed” XML

Adheres to basic structural requirements

- Single root element
- Matched tags, proper nesting
- Unique attributes within elements



Displaying XML

Use rule-based language to translate to HTML

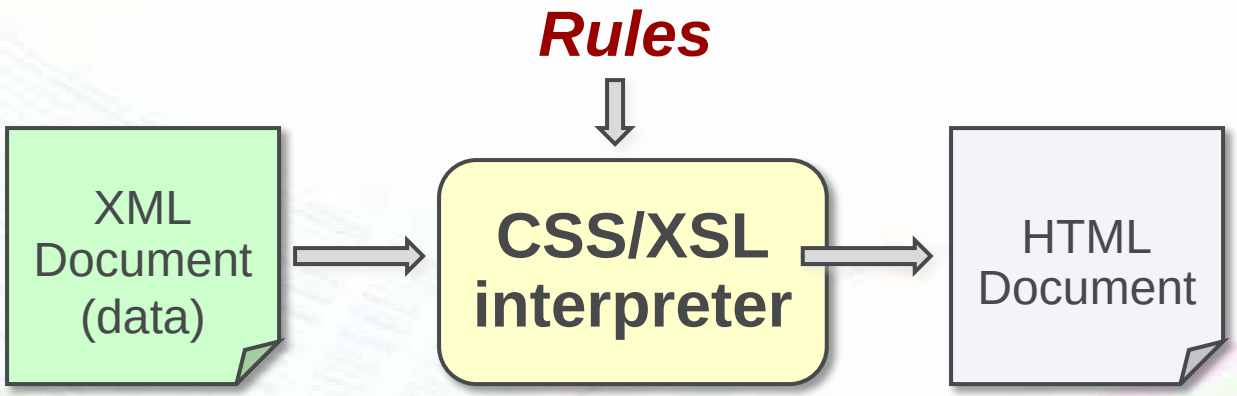
- *Cascading stylesheets* (CSS)
- *Extensible stylesheet language* (XSL)

```
<?xml version="1.0" ?>
<!-- Bookstore with no DTD  -->
- <Bookstore>
  - <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <Title>A First Course in Database Systems</Title>
    - <Authors>
      - <Author>
          <First_Name>Jeffrey</First_Name>
          <Last_Name>Ullman</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jennifer</First_Name>
          <Last_Name>Widom</Last_Name>
        </Author>
      </Authors>
    </Book>
  - <Book ISBN="ISBN-0-13-815504-6" Price="100">
      <Remark>Buy this book bundled with "A First Course" -- a great deal!</Remark>
      <Title>Database Systems: The Complete Book</Title>
    - <Authors>
```

Displaying XML

Use rule-based language to translate to HTML

- *Cascading stylesheets* (CSS)
- *Extensible stylesheet language* (XSL)



“Valid” XML

Adheres to basic structural requirements

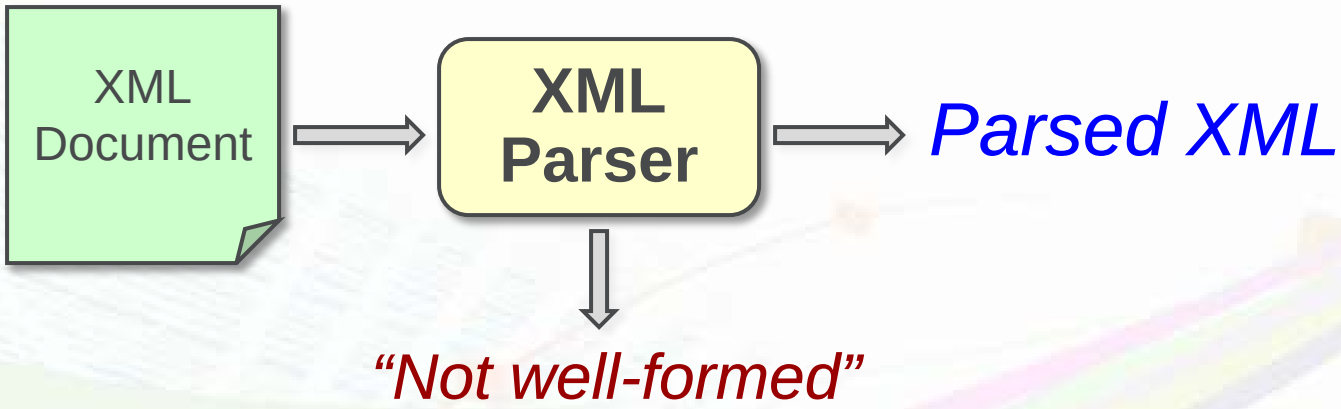
- Also adheres to content-specific specification
  - *Document Type Descriptor* (DTD)
  - *XML Schema* (XSD)

```
<?xml version="1.0" ?>
<!-- Bookstore with no DTD -->
- <Bookstore>
  - <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <Title>A First Course in Database Systems</Title>
    - <Authors>
      - <Author>
          <First_Name>Jeffrey</First_Name>
          <Last_Name>Ullman</Last_Name>
        </Author>
      - <Author>
          <First_Name>Jennifer</First_Name>
          <Last_Name>Widom</Last_Name>
        </Author>
      </Authors>
    </Book>
  - <Book ISBN="ISBN-0-13-815504-6" Price="100">
      <Remark>Buy this book bundled with "A First Course" -- a great deal!</Remark>
      <Title>Database Systems: The Complete Book</Title>
    - <Authors>
```

“Valid” XML

Adheres to basic structural requirements

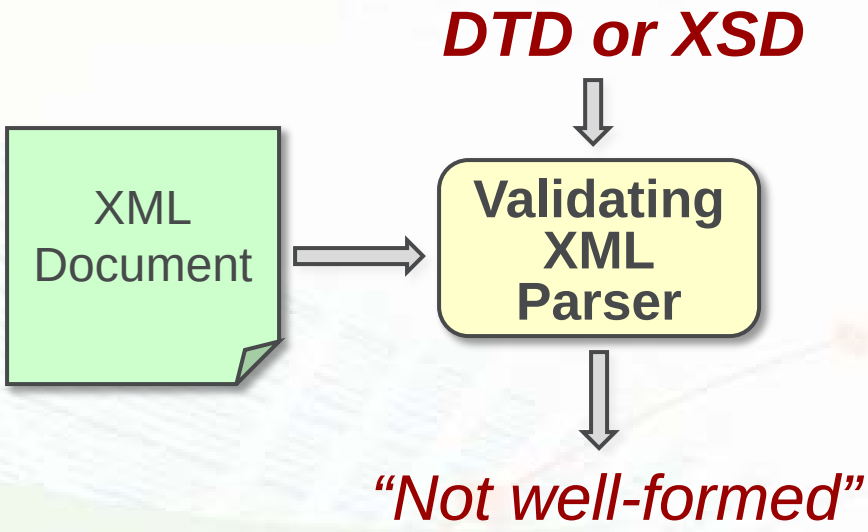
- Also adheres to content-specific specification



“Valid” XML

Adheres to basic structural requirements

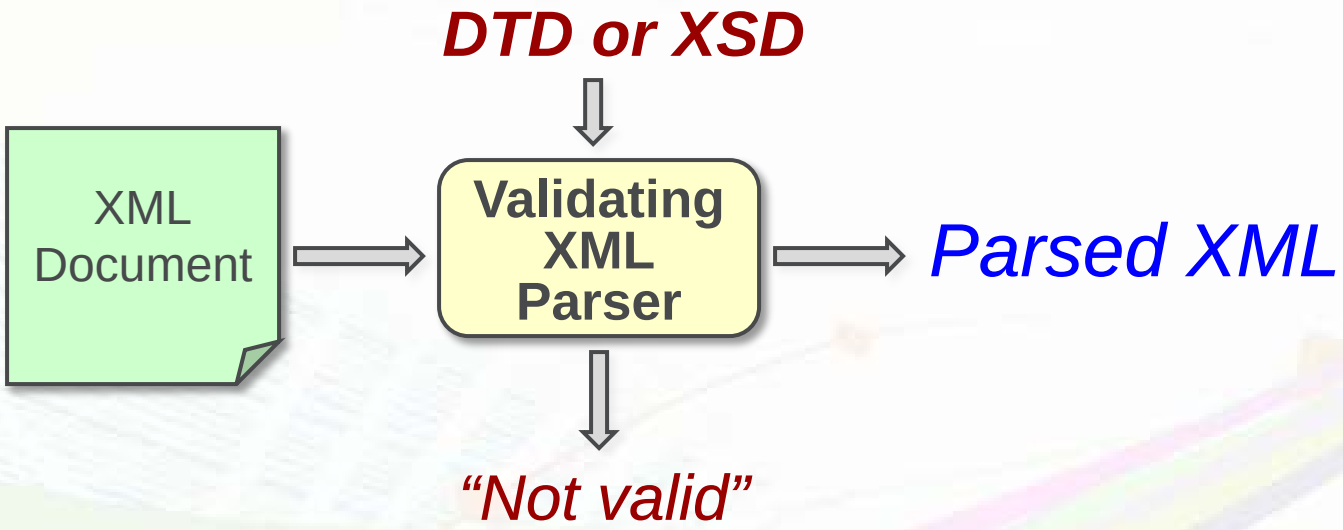
- Also adheres to content-specific specification



“Valid” XML

Adheres to basic structural requirements

- Also adheres to content-specific specification



Document Type Descriptor (DTD)

- Grammar-like language for specifying elements, attributes, nesting, ordering, #occurrences
- Also special attribute types ID and IDREF(S)

```
<!DOCTYPE Bookstore [
   <!ELEMENT Bookstore (Book*, Author*)>
   <!ELEMENT Book (Title, Remark?)>
   <!ATTLIST Book ISBN ID #REQUIRED
              Price CDATA #REQUIRED
              Authors IDREFS #REQUIRED>
   <!ELEMENT Title (#PCDATA)>
   <!ELEMENT Remark (#PCDATA | BookRef)*>
   <!ELEMENT BookRef EMPTY>
   <!ATTLIST BookRef book IDREF #REQUIRED>
   <!ELEMENT Author (First_Name, Last_Name)>
   <!ATTLIST Author Ident ID #REQUIRED>
   <!ELEMENT First_Name (#PCDATA)>
   <!ELEMENT Last_Name (#PCDATA)>
]>
```

DTD/XSD versus none (well-formed)

+ DTD/XSD
Advantages of typing
- DTD/XSD
Advantages of no typing

XML Schema (XSD)

- Extensive language

nts, attributes,
es

```
<?xml version="1.0" ?>
<!-- XSD for Bookstore-XSD.xml -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="Bookstore">
      <xsd:complexType>
         <xsd:sequence>
      </xsd:complexType>
      <xsd:key name="BookKey">
         <xsd:selector xpath="Book" />
         <xsd:field xpath="@ISBN" />
      </xsd:key>
      <xsd:key name="AuthorKey">
         <xsd:selector xpath="Author" />
         <xsd:field xpath="@Ident" />
      </xsd:key>
      <xsd:keyref name="AuthorKeyRef" refer="AuthorKey">
         <xsd:selector xpath="Book/Authors/Auth" />
         <xsd:field xpath="@authIdent" />
      </xsd:keyref>
      <xsd:keyref name="BookKeyRef" refer="BookKey">
   </xsd:element>
   <xsd:complexType name="BookType">
      <xsd:sequence>
         <xsd:element name="Title" type="xsd:string" />
         <xsd:element name="Authors">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Auth" maxOccurs="unbounded">
```

```
<xsd:complexType name="AuthorType">
   <xsd:sequence>
      <xsd:element name="First_Name" type="xsd:string" />
      <xsd:element name="Last_Name" type="xsd:string" />
   </xsd:sequence>
   <xsd:attribute name="Ident" type="xsd:string" use="required" />
</xsd:complexType>
```

JavaScript Object Notation (JSON)

- Standard for “serializing” data objects, usually in files
- Human-readable, useful for data interchange
- Also useful for representing & storing semistructured data

```
{ "Books":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Price":85,
      "Edition":3,
      "Title":"A First Course in Database Systems",
      "Authors":[ {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Price":100,
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",
      "Title":"Database Systems:The Complete Book",
      "Authors":[ {"First_Name":"Hector", "Last_Name":"Garcia-Molina"},
                  {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
  ]
```

JavaScript Object Notation (JSON)

- No longer tied to JavaScript
- Parsers for many languages

```
{ "Books":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Price":85,
      "Edition":3,
      "Title":"A First Course in Database Systems",
      "Authors":[ {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Price":100,
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",
      "Title":"Database Systems:The Complete Book",
      "Authors":[ {"First_Name":"Hector", "Last_Name":"Garcia-Molina"},
                  {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
  ]
```

```
{ "Books":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Price":85,
      "Edition":3,
      "Title":"A First Course in Database Systems",
      "Authors":[ {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Price":100,
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",
      "Title":"Database Systems:The Complete Book",
      "Authors":[ {"First_Name":"Hector", "Last_Name":"Garcia-Molina"},
                  {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
  ],
  "Magazines":
  [
    { "Title":"National Geographic",
      "Month":"January",
      "Year":2009 }
    ,
    { "Title":"Newsweek",
      "Month":"February",
      "Year":2009 }
  ]
}
```

Basic constructs (recursive)

- Base values
  number, string, boolean, …
- Objects { }
  sets of label-value pairs
- Arrays [ ]
  lists of values

Syntactically valid JSON

Adheres to basic structural requirements

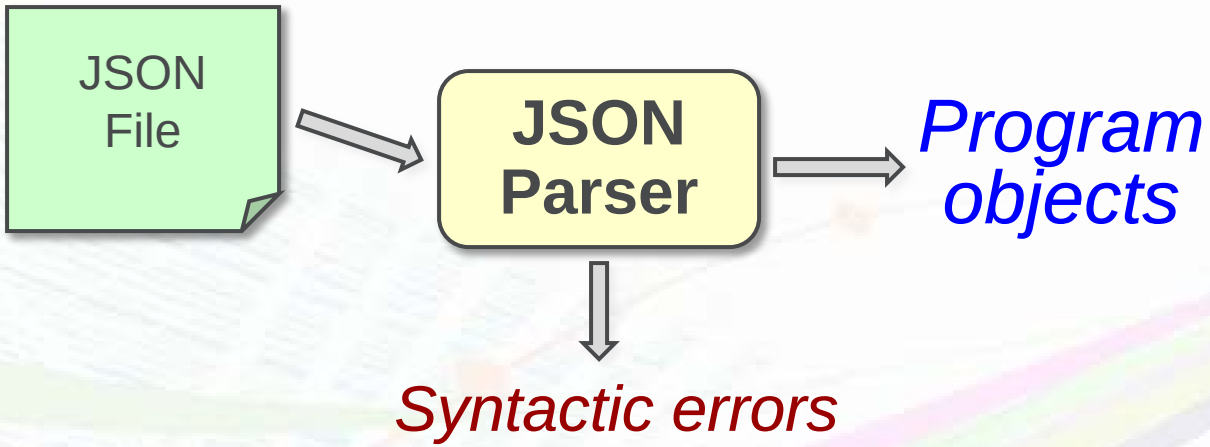
- Sets of label-value pairs
- Arrays of values
- Base values from predefined types

```
{ "Books":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Price":85,
      "Edition":3,
      "Title":"A First Course in Database Systems",
      "Authors":[ {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Price":100,
      "Remark":"Buy this book bundled with 'A First Course' - a great deal!",
      "Title":"Database Systems:The Complete Book",
      "Authors":[ {"First_Name":"Hector", "Last_Name":"Garcia-Molina"},
                  {"First_Name":"Jeffrey", "Last_Name":"Ullman"},
                  {"First_Name":"Jennifer", "Last_Name":"Widom"} ] }
  ]
```

Syntactically valid JSON

Adheres to basic structural requirements

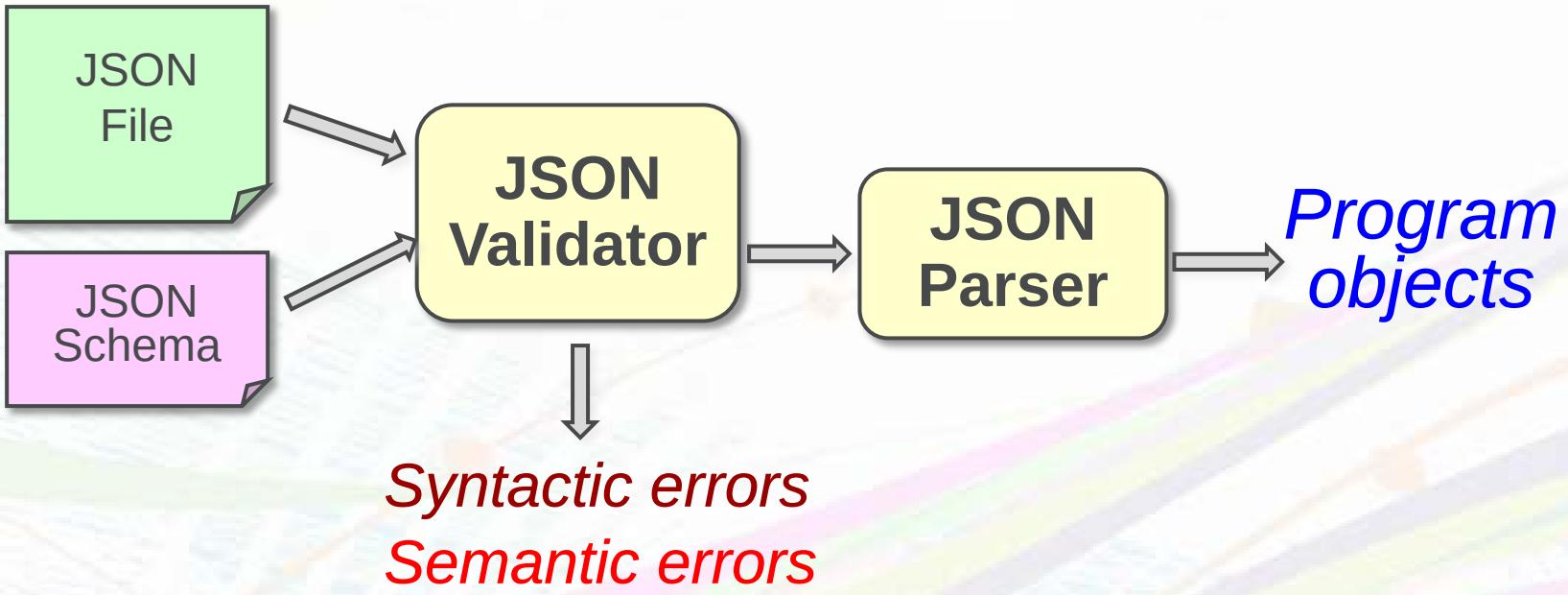
- Sets of label-value pairs
- Arrays of values
- Base values from predefined types



Semantically valid JSON

Adheres to basic structural requirements

+ conforms to specified schema



Semantically valid JSON

Adheres to basic structural requirements

+ conforms to specified schema

```
{ "type":"object",
  "properties": {
     "Magazines": {
        "type":"array",
        "items": {
           "type":"object",
           "properties": {
              "Title": { "type":"string" },
              "Month": { "type":"string",
                         "enum":["January","February"] },
              "Year": { "type":"integer" }}}}
}}
```

XML versus JSON

	XML	JSON
Performance		
Querying	XPath XQuery XLST	JSON Path JSON Query JAQL