JSON-LD*

- Extend JSON-LD syntax to allow a node object to be the value of @id.
    - Restricts the shape of that node object to describe a single triple.

```
{
  "@context": {
    "@base": "http://example.org/",
    "ex:" "http://example.org/".
    "foaf": "http://xmlns.com/foaf/0.1/"
  }.
  "@id": {
    "@id": "bob",
    "foaf:age": 23
  },
  "ex:certainty": 0.8
}
```

```
@base <http://example.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<<<bob> foaf:age 23>> <certainty> 8.0e-1 .
```

JSON-LD Logic

- Using Notation-3 data model and builtins, JSON-LD may become an alternative syntax for logic programming.
    - Requires additional resource types, blank node scoping semantics, and treating named graphs as quoted graphs/formulae.

```
:Joe :parent :Alan.
:Alan :sister :Susie.

{?x :parent ?y . ?y :sister ?z} => {?x :aunt ?z} .

# :Joe :aunt :Suzie .
```

```
{
  "@context": {
    "@base": "http://example.org/",
    "@vocab": "#",
    "log": "http://www.w3.org/2000/10/swap/log#",
    "=>": {"@id": "log:implies", "@container": "@graph"}
  },
  "@id": "Joe",
  "parent": {"@id": "Alan", "sister": {"@id": "Suzie"}},
  "@included": {
    "@graph": {
      "@id": "?x",
      "parent": {"@id": "?y", "sister": {"@id": "?z"}}
    },
    "=>": {
      "@id": "?x",
      "aunt": {"@id": "?x"}
    }
  }
}
```