# JSON-LD anonymous named graphs

- Notation3 Formulae are effectively anonymous named graphs
  - JSON-LD has good support for anonymous named graphs:

```
{
  "@context": {
    "@base": “http://bigdata.com/",
    "foaf": "http://xmlns.com/foaf/0.1/",
    "dc": "http://purl.org/dc/elements/1.1/",
    "age": “dc:age",
    "creator": {"@id": "dc:creator", "@type": "@id"},
    "source": {"@id": "dc:source", "@type": "@id"}
  },
  "@graph": [
    {"@id": "bob", "foaf:name": "Bob"},
    {
      "@graph": {"@id": "bob", "age": 23},
      "creator": "http://example.com/crawlers#c1",
      "source": "http://example.net/homepage-listing.html"
    }
  ]
}
```

# Reasoning in JSON-LD?

- Leverage use of anonymously named graphs to do implication (**log:implies**).
  - Requires a way to identify universal quantifiers (existential quantifiers simply blank nodes)

## Notation3

```
@forAll :x, :y.
:Julie :parent :Suzie .
{ :x :parent :y } =>  { :y :child :x }.
```

```
:Julie :parent :Suzie .
{ ?x parent ?y } =>  { ?y :child ?x }.
```

## JSON-LD

```
{
  "@context": {
    "@base": "http://example.com/",
    "@vocab": "http://example.com/",
    "=>": {"@id": "http://www.w3.org/2000/10/swap/log#implies", "@container": "@graph"},
    "?x": {"@univar": true},
    "?y": {"@univar": true}
  },
  "@graph": [
    {"@id": "Julie", "parent": {"@id": "Suzie"}},
    {
      "@graph": {"@id": "?x", "parent": "?y"},
      "=>": {"@id": "?y", "child": "?x"}
    }
  ]
}
```

Note: @univar is totally hypothetical

Note: hand waiving on how “?x” is expanded