RDF Schema provides a data-modeling vocabulary for RDF data. With RDFS we can specify

* That something is a **class**, *ex:Person rdf:type rdfs:Class.*
* That something is an **instance of a class**, *ex:bob rdf:type ex:Person.*
* That a **class is included in another class**, *ex:Student rdfs:subClassOf ex:Person.*
* The **domain and range of a property**, *ex:authorOf rdfs:domain ex:Person; rdfs:range ex:Book.*
* That a **property is included in another property**, *ex:parentOf rdfs:subPropertyOf ex:relativeOf*

OWL2: Semantic Web language for expressing ontologies

Ontology: descriptive statements about some part of the world, is made up of two parts

* **Terminology** (DL:TBox) which involves classes (DL:concepts), properties (DL:roles) and relationship between them (*Student are people* or *Happy parents are parent with at least one happy child*
* **Assertional knowledge** (DL:Abox) stating facts about entities of the domain of interest (*Bob is a student, Alice is an happy person, Bob knows Alice*)

Let R be a binary relation on a set X, R ⊆ X × X. Then R is

* **Reflexive**: if ∀x € X: (x,x) € R ⟹ expressed as *R rdf:type owl:ReflexiveProperty*
* **Irreflexive**: if ∀x € X: (x,x) ∉ R ⟹ expressed as *R rdf:type owl:IrreflexiveProperty*
* **Symmetric**: if ∀x,y € X: (x,y) € R implies (y,x) € R ⟹ expressed as *R rdf:type owl:SymmetricProperty*
* **Transitive**: if ∀x,y,z € X: (x,y) € R and (y,z) € R implies (x,z) € R ⟹ expressed as *R rdf:type owl:AsymmetricProperty*
* **Functional**: if ∀x,y,z € X: (x,y) € R and (x,z) € R implies y = z ⟹ expressed as *R rdf:type owl:TransitiveProperty*
* **Inverse functional**: if ∀x,y,z € X: (y,x) € R and (z,x) € R implies y = z ⟹ expressed as *R rdf:type owl:FunctionalProperty*