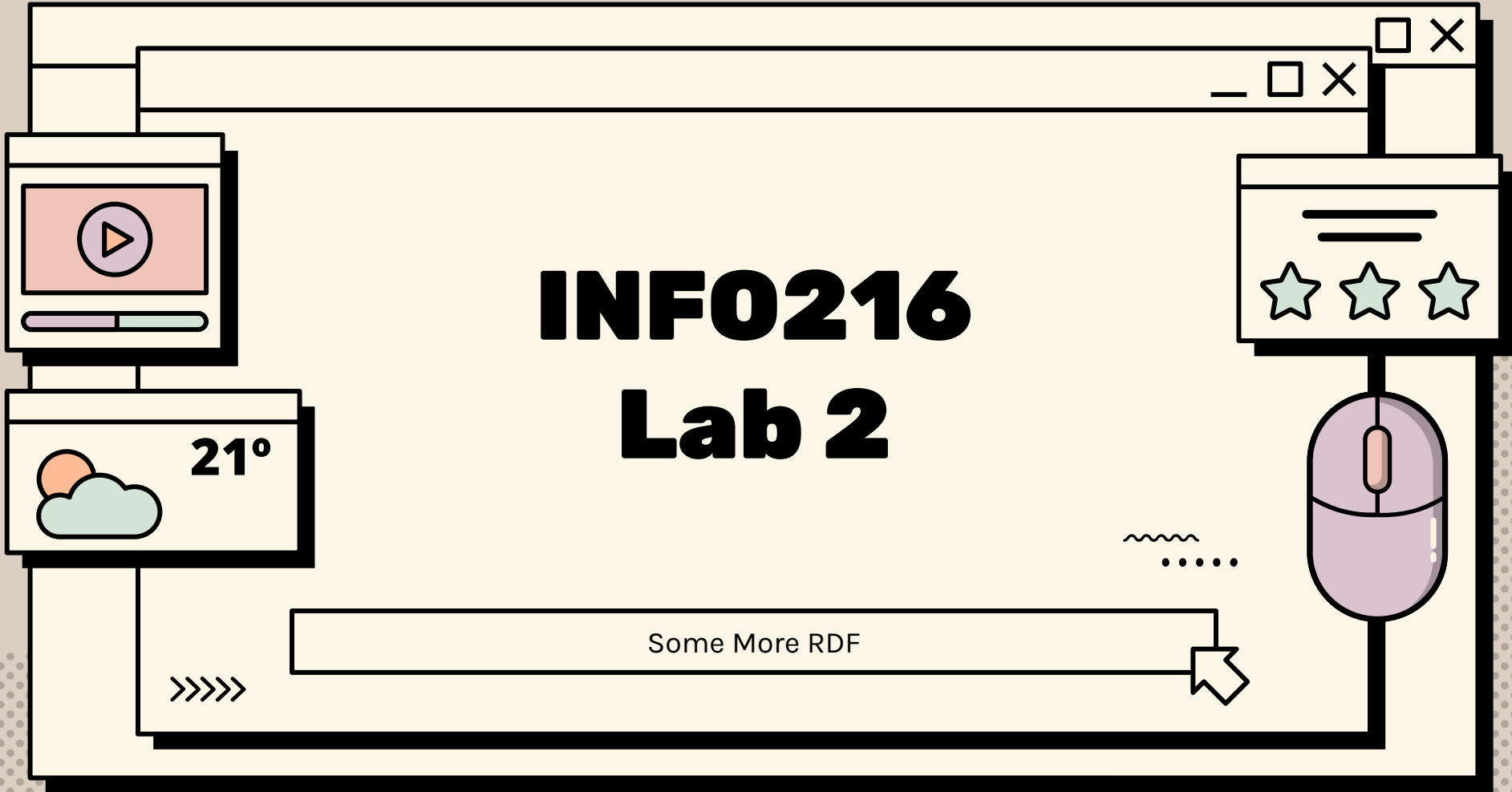


# INFO216

## Lab 2





# Useful Methods



## **g.remove()**

Used to remove triples from the graph. Same syntax as g.add



## **Collection()**

Used to add more members to your triple (often objects)



## **BNode()**

An anonymous node, different use cases (see lecture notes)



## **g.parse()**

Used to retrieve graphs from external sources (e.g. file)



## **RDF.type**

Predicate used to specify what a subject is (e.g. a Person)



## **g.bind()**

Used to add your prefix to the graph. Otherwise it's ns1, ns2 etc.



**LET'S TAKE A LOOK AT BLANK NODES. WE WANT USE BLANK NODES WHENEVER:**

- **WE ARE NOT YET SURE WHAT THE URI (UNIFORM RESOURCE IDENTIFIER) WILL BE**
- **WE DO NOT WANT TO IDENTIFY THE URI**
- **WE WANT TO GROUP PROPERTIES THAT ARE RELATED**

**THE LATTER IS WHAT WE ARE GOING TO BE WORKING WITH TODAY. HERE IS A QUICK EXAMPLE OF HOW YOU COULD HAVE EXPRESSED SOMEONE'S ADDRESS WITH WHAT YOU LEARNED LAST LAB**

**@prefix ex: <<http://example.org/>>.**

```
ex:Emma ex:streetName ex:SomeStreetName;  
  ex:streetNumber "104"^^xsd:integer;  
  ex:postalCode "5555"^^xsd:integer;  
  ex:areaName ex:SomeAreaName;  
  ex:city ex:Bergen;  
  ex:county ex:Vestland;  
  ex:country ex:Norway.
```



**NOW USING BLANK NODES, WE CAN GROUP THE RELATED PROPERTIES FROM THE PREVIOUS EXAMPLE. WE CAN THEN CHANGE THE GRAPH TO THE FOLLOWING:**

**@prefix ex: <<http://example.org/>>.**

**ex:Emma **ex:address** [ ex:streetName ex:SomeStreetName;  
ex:streetNumber "104"^^xsd:integer;  
ex:postalCode "5555"^^xsd:integer;  
ex:areaName ex:SomeAreaName;  
ex:city ex:Bergen;  
ex:county ex:Vestland;  
ex:country ex:Norway ].**

**ESSENTIALLY WHAT IS SAID HERE IS THE FOLLOWING:**

**ex:Emma ex:address **\_:b0** (blank node)  
**\_:b0** ex:streetName ex:SomeStreetName (and then the rest of the triples)**



**IF WE WANT TO ADD THE PREVIOUS EXAMPLE TO OUR GRAPH WITH RDFLIB, WE CAN DO THE FOLLOWING:**

```
from rdflib import Graph, Namespace, BNode, Literal, XSD

g = Graph()

ex = Namespace('http://example.org/')

EmmaAddress = BNode()

g.add((EmmaAddress, ex.streetName, ex.SomeStreetName))
g.add((EmmaAddress, ex.streetNumber, Literal(104, datatype=XSD.integer)))
g.add((EmmaAddress, ex.postalCode, Literal(5555, datatype=XSD.integer)))
g.add((EmmaAddress, ex.areaName, ex.SomeAreaName))
g.add((EmmaAddress, ex.city, ex.Bergen))
g.add((EmmaAddress, ex.county, ex.Vestland))
g.add((EmmaAddress, ex.country, ex.Norway))
g.add((ex.Emma, ex.address, EmmaAddress))

print(g.serialize(format='ttl'))
```



**TO SOLVE LAST WEEK'S LAB YOU HAD TO WRITE SEVERAL TRIPLES TO SOLVE "IT INVOLVED PAUL MANAFORT, RICK GATES, GEORGE PAPADOPOULOS, MICHAEL FLYNN, MICHAEL COHEN, AND ROGER STONE." E.G. LIKE THIS:**

**@prefix ex: <<http://example.org/>>.**

**ex:MuellerInvestigation ex:involved ex:RogerStone;  
ex:involved ex:PaulManafort; (and so on)**

**THERE ARE MORE EFFECTIVE WAYS OF WRITING IT, USING COLLECTION. WE CAN USE THIS TO GROUP RESOURCES. SAY WE WANT TO SOLVE LAST WEEK'S LAB TASK:**

**@prefix ex: <<http://example.org/>>.**

**ex:MuellerInvestigation ex:involved (ex:RogerStone ex:PaulManafort  
ex:RickGates ex:GeorgePapaDopoulos ex:MichaelFlynn ex:MichaelCohen).**



**IF WE WANT TO ADD THE PREVIOUS EXAMPLE TO OUR GRAPH WITH RDFLIB, WE CAN DO THE FOLLOWING:**

```
from rdflib import Graph, Namespace, BNode  
from rdflib.collection import Collection #Note you need rdflib.collection!  
  
g = Graph()  
  
ex = Namespace('http://example.org/')  
  
MuellerInvestigationConviction = BNode()  
Collection(g, MuellerInvestigationConviction, [ex.RickGates, ex.RogerStone,  
                                                ex.PaulManafort, ex.GeorgePapaDopoulos,  
                                                ex.MichaelFlynn, ex.MichaelCohen])  
  
g.add((ex.MuellerInvestigation, ex.involved, MuellerInvestigationConviction))  
  
print(g.serialize(format='ttl'))
```