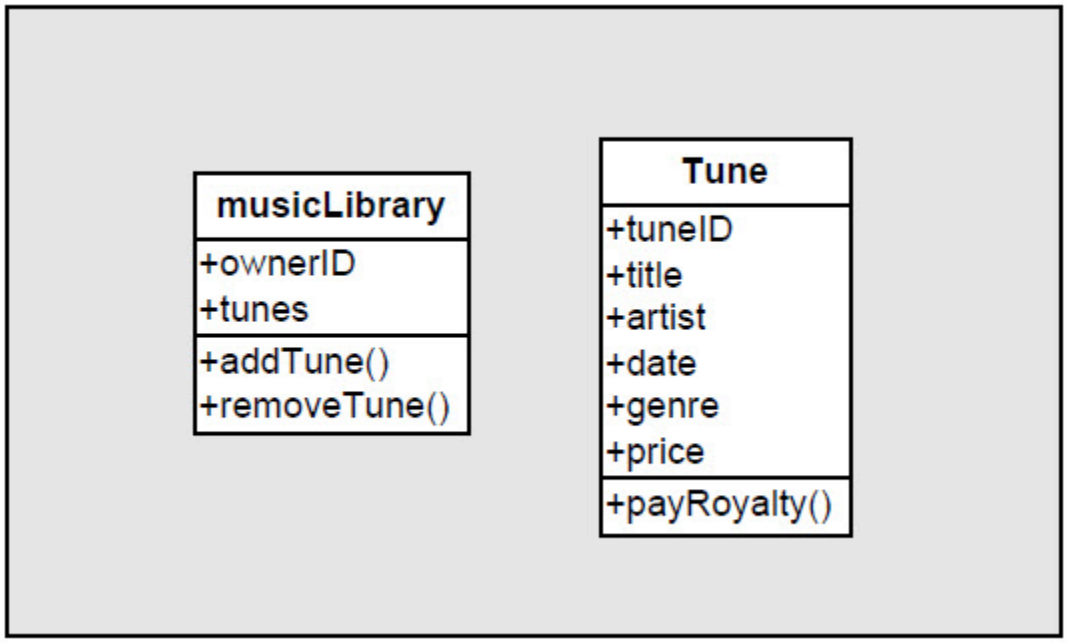
# DIAGRAM EXAMPLES

## THE OBJECT DIAGRAM

The object diagram is the main building block of JavaScript structural modeling.



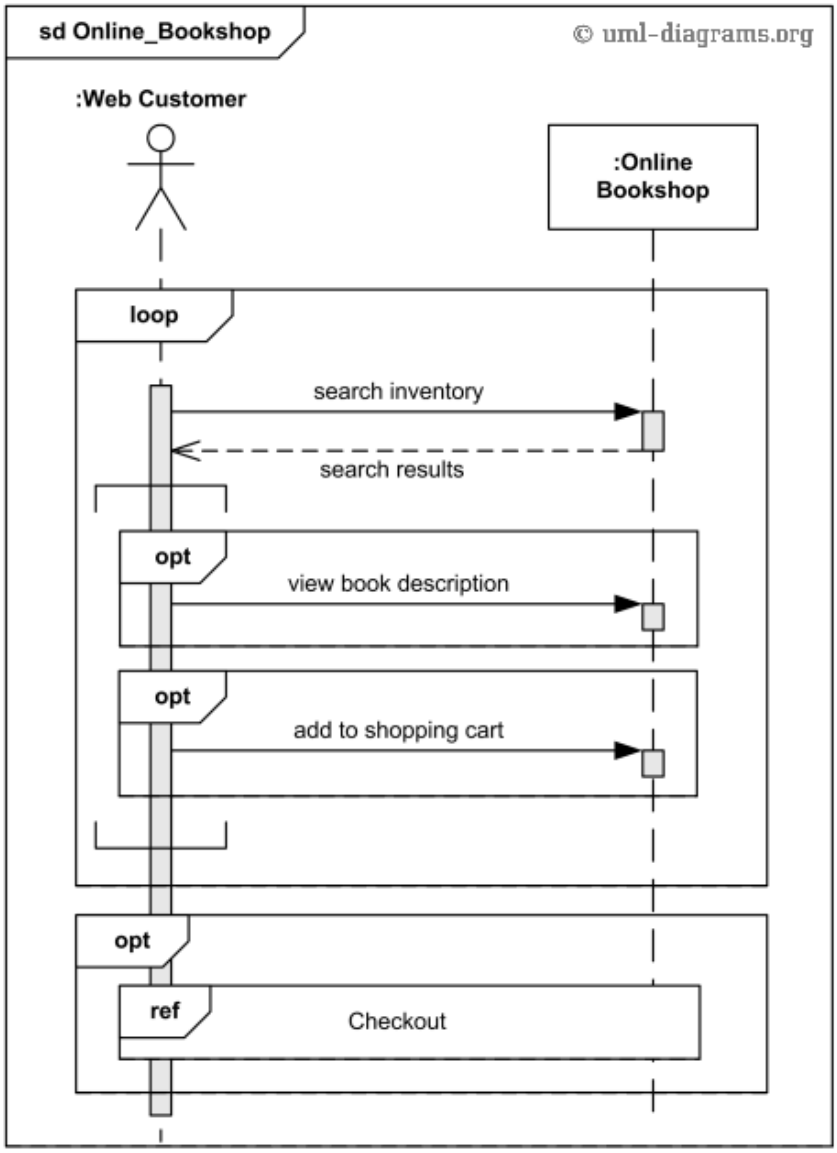

An Object Diagram has three sections, represented with boxes which contain three parts:

- The top part contains the name of the object. It is printed in bold and centered. It may be helpful to designate the type of object you're trying to represent: Literal or Constructor. There's no real convention for this, you can choose to make a note or merely use capitalization.
- The middle part contains the value properties that define the object.

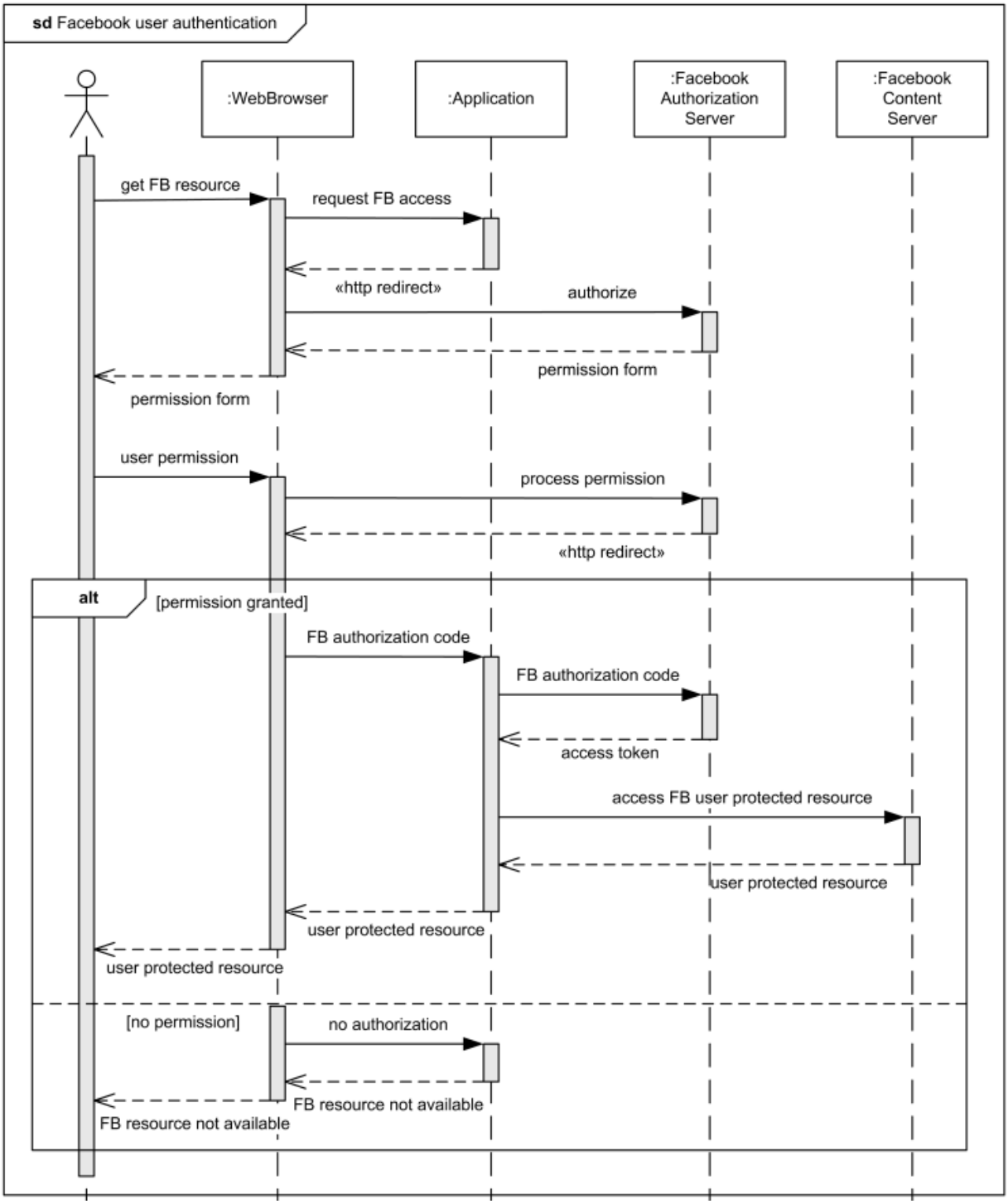
- The bottom part contains the method properties available in the object. (These are the actions that the object can perform.)

## THE SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another, arranged in time sequence. They are sometimes called event diagrams or event scenarios.

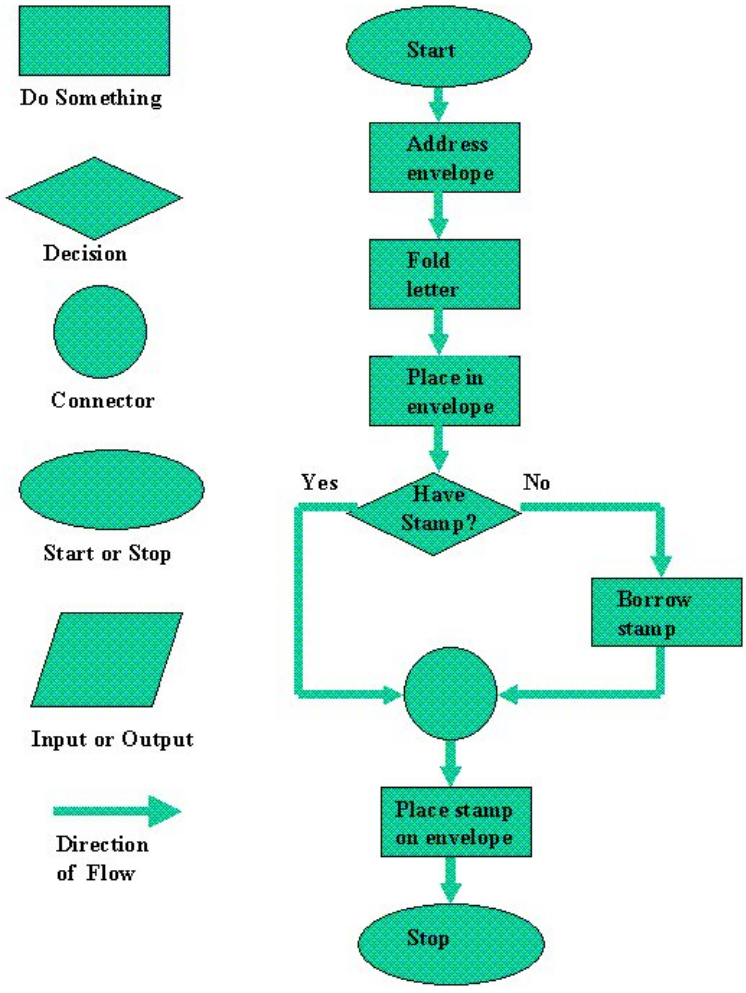



A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.


sd Facebook user authentication
:WebBrowser
:Application
:Facebook Authorization Server
:Facebook Content Server
get FB resource
request FB access
«http redirect»
authorize
permission form
permission form
user permission
process permission
«http redirect»
alt
[permission granted]
FB authorization code
FB authorization code
access token
access FB user protected resource
user protected resource
user protected resource
user protected resource
[no permission]
no authorization
FB resource not available
FB resource not available

## THE FLOWCHART

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

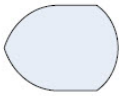


# COMMON FLOWCHART SYMBOLS

Sean Olson
Riverside JS Workshop
RiversideJS.net
3/13/2015

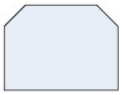
## Common Flowchart Symbols


### Start/End Symbol
The terminator symbol marks the starting or ending point of the system. It usually contains the word "Start" or "End."

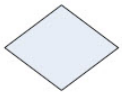


### Display Symbol
Indicates a step that displays information.

### Action or Process Symbol
A box can represent a single step ("add two cups of flour"), or and entire sub-process ("make bread") within a larger process.

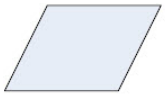


### Loop Limit Symbol
Indicates the point at which a loop should stop.



### Decision Symbol
A decision or branching point. Lines representing different decisions emerge from different points of the diamond.

### Data Storage or Stored Data Symbol
Indicates a step where data gets stored.

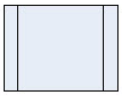


### Input/Output Symbol
Represents material or information entering or leaving the system, such as customer order (input) or a product (output).

### Internal Storage Symbol
Indicates that information was stored in memory during a program, used in software design flowcharts.

### Connector Symbol
Indicates that the flow continues where a matching symbol (containing the same letter) has been placed.

### Off Page
Indicates that the process continues off page.



### Subroutine Symbol
Indicates a sequence of actions that perform a specific task embedded within a larger process. This sequence of actions could be described in more detail on a separate flowchart.

### Delay Symbol
Indicates a delay in the process.

Sean Olson
Riverside JS Workshop
RiversideJS.net
4/5/2015