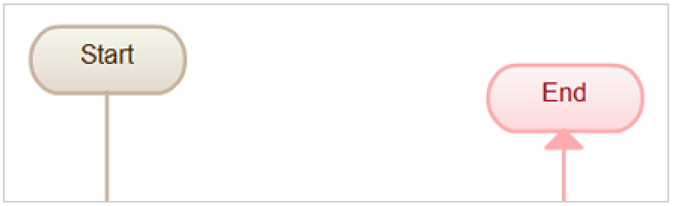
All Diagram Objects

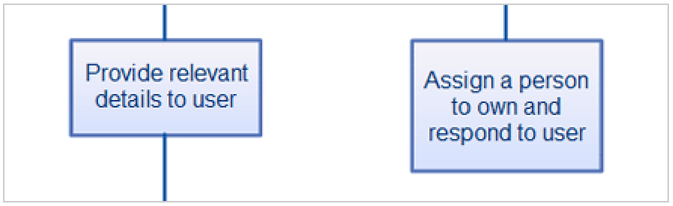
Terminal / Terminator

The terminator is used to show where your flow begins or ends. Ideally, you would use words like 'Start', 'Begin', 'End' inside the terminator object to make things more obvious.



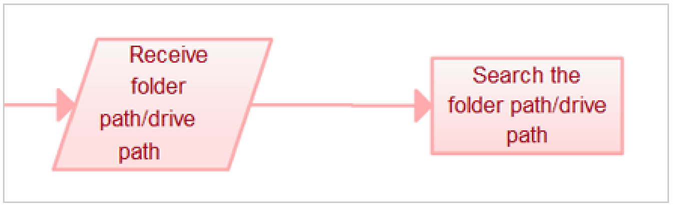
Process / Rectangle

Flowchart Process object is used to illustrate a process, action or an operation. These are represented by rectangles; and the text in the rectangle mostly includes a verb. Examples include 'Edit video', 'Try Again', 'Choose your Plan'.


Provide relevant details to user
Assign a person to own and respond to user

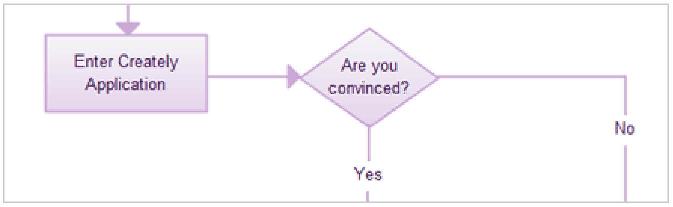
Data (I/O)

The Data object, often referred to as the I/O Shape shows the Inputs to and Outputs from a process. This takes the shape of a parallelogram.

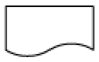

Receive folder path/drive path
Search the folder path/drive path

Decision / Conditional

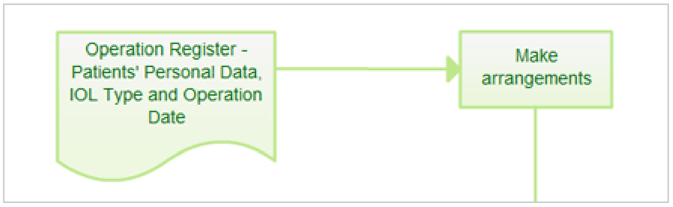
Decision object is represented as a Diamond. This object is always used in a process flow to as a question. And, the answer to the question determines the arrows coming out of the Diamond. This shape is quite unique with two arrows coming out of it. One from the bottom point corresponding to Yes or True and one from either the right/left point corresponding to No or False. The arrows should be always labelled to avoid confusion in the process flow.



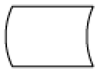
Document



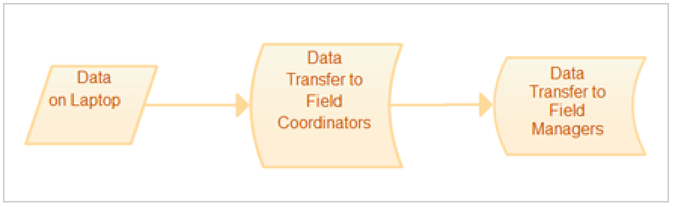
Document object is a rectangle with a wave-like base. This shape is used to represent a Document or Report in a process flow.


Operation Register - Patients' Personal Data, IOL Type and Operation Date
Make arrangements

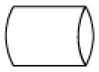
Stored Data



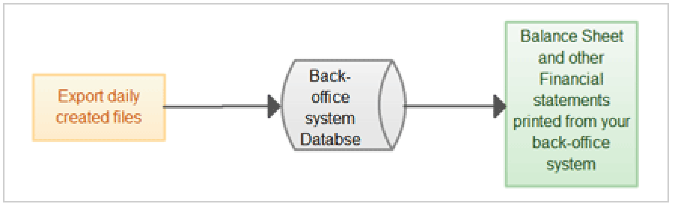
This is a general data storage object used in the process flow as opposed to data which could be also stored on a hard drive, magnetic tape, memory card, of any other storage device.


Data on Laptop
Data Transfer to Field Coordinators
Data Transfer to Field Managers

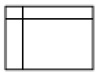
Direct Data



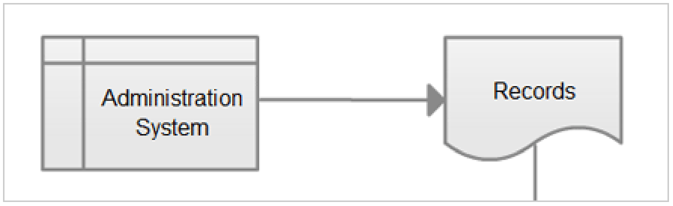
Direct Data object in a process flow represents information stored which can be accessed directly. This object represents a computer's hard drive.


Export daily created files
Back-office system Databse
Balance Sheet and other Financial statements printed from your back-office system

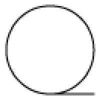
Internal Storage



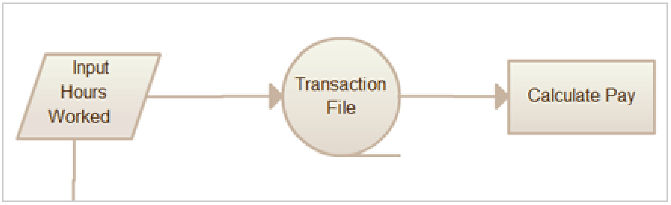
This is an object which is commonly found in programming flowcharts to illustrate the information stored in memory, as opposed to on a file. This shape is often referred to as the magnetic core memory of early computers; or the random access memory (RAM) as we call it today.



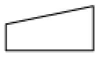
Sequential Access



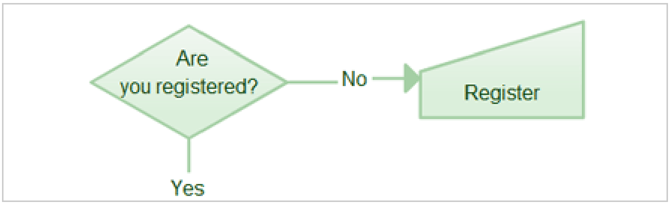
This object takes the shape of a reel of tape. It represents information stored in a sequence, such as data on a magnetic tape.



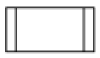
Manual Input



This object is represented by rectangle with the top sloping up from left to right. The Manual Input object signifies an action where the user is prompted for information that must be manually input into a system.

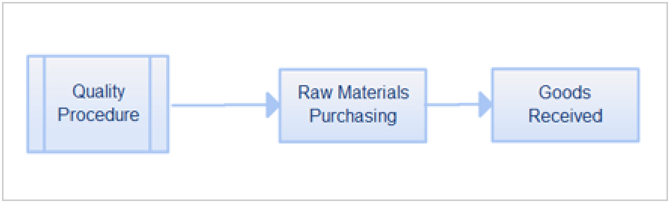


Subroutine / Predefined Process



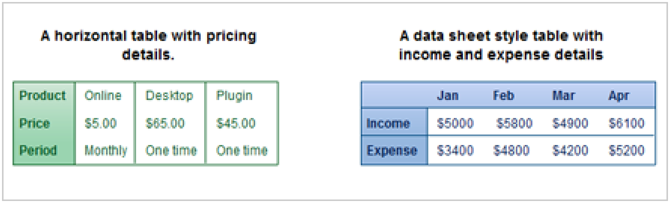
This shape takes two names - 'Subroutine' or 'Predefined Process'. Its called a subroutine if you use this object in flowcharting a software program. This allows you to write one subroutine and call it as often as you like from anywhere in the code.

The same object is also called a Predefined Process. This means the flowchart for the predefined process has to be already drawn, and you should reference the flowchart for more information.



Table

Table object from Creately's UI Mockup set is a **real ease to use**! It can be used for anything as simple as a **data sheet**, to a complex **sudoku game** and even a **word puzzle**.



A horizontal table with pricing details.

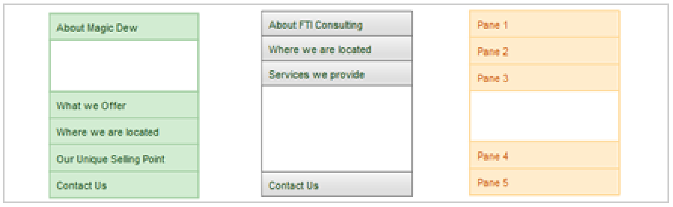
Product	Online	Desktop	Plugin
Price	$5.00	$65.00	$45.00
Period	Monthly	One time	One time

A data sheet style table with income and expense details

	Jan	Feb	Mar	Apr
Income	$5000	$5800	$4900	$6100
Expense	$3400	$4800	$4200	$5200

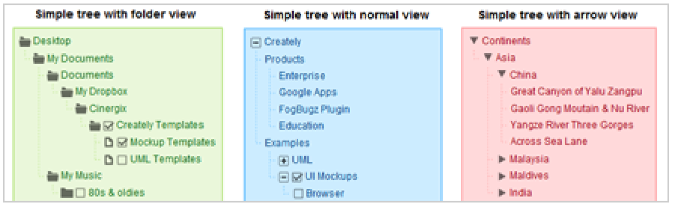
Accordion

Completely **customizable**, collapsible Accordion pane is just a few keystrokes away. Get **started** right away.



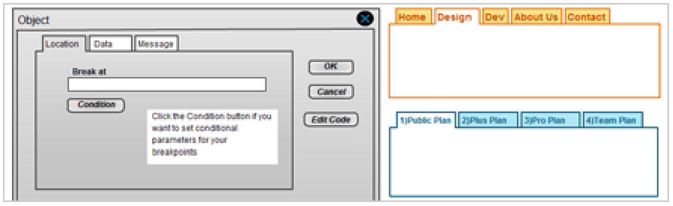
Tree View Control

Tree View control is a **real child's play** with Creately. Use it in your UI mock-ups to execute hierarchical data sets, folder views and other similar data structures. This is the most simple Text-driven Smart KObject from the list. **Go give it a spin**!



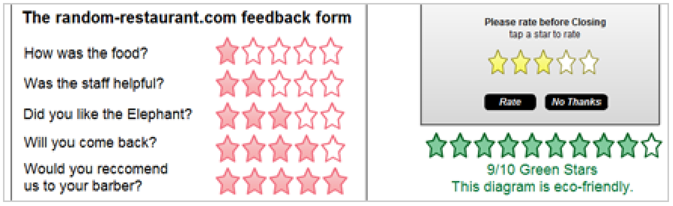
Tab Control

Are you looking for a solution to **create multiple tab pages** in seconds? With Creately's **smart KObject improvements** you can create as many Tabs as you might inside the Tab Control. Check this for some **usage examples of Tab Controls**.



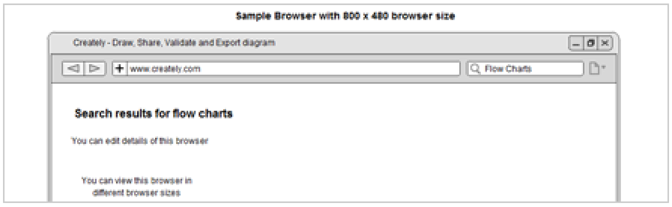
Star Rating

Creately's Star Rating object provides an **easy rating experience** that allows users to select the number of stars that represents their rating. Wanna see some examples? Click **here**!


The random-restaurant.com feedback form
How was the food?
Was the staff helpful?
Did you like the Elephant?
Will you come back?
Would you reccomend us to your barber?
Please rate before Closing
tap a star to rate
Rate
No Thanks
9/10 Green Stars
This diagram is eco-friendly.

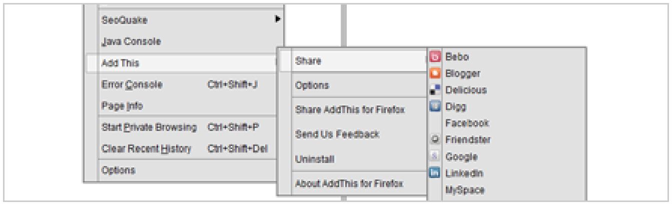
Browser

Have you designed a mockup, and want to see how it'll appear in a browser? This is simple with **Creately's Browser object**! Simply drag-n-drop a Browser object and **configure the properties** to create your browser.


Sample Browser with 800 x 480 browser size
Creately - Draw, Share, Validate and Export diagram
www.creately.com
Flow Charts
Search results for flow charts
You can edit details of this browser
You can view this browser in different browser sizes

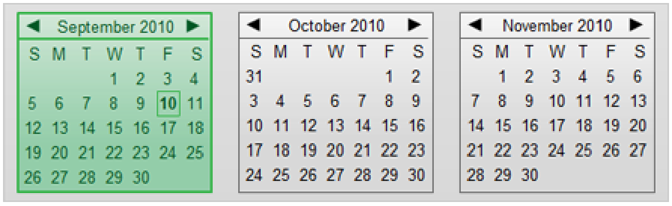
Dropdown Menu

Create **stylish Dropdown Menus** in just a few minutes! Make a few **text edits** on the Smart Dropdown Menu object to configure your Dropdown Menu. Then select a suitable Dropdown menu style and attach it to your Menu bar.


SeoQuake
Java Console
Add This
Error Console Ctrl+Shift+J
Page Info
Start Private Browsing Ctrl+Shift+P
Clear Recent History Ctrl+Shift+Del
Options
Share
Options
Share AddThis for Firefox
Send Us Feedback
Uninstall
About AddThis for Firefox
Bebo
Blogger
Delicious
Digg
Facebook
Friendster
Google
LinkedIn
MySpace

Calendar

Creately's Calendar object **re-sizes** pretty well to fit into mockup or wireframe. Add the Month, Year and Day to highlight the day. Check out here for some real **Calendar object examples**!



September 2010

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

October 2010

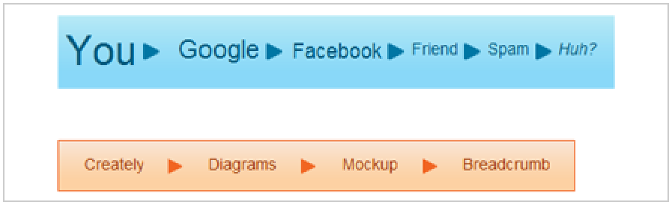
S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

November 2010

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

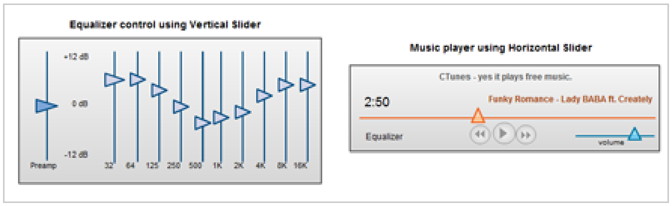
Breadcrumb Navigation

Designing a Breadcrumb Navigation in Creately is **easy and simple** as typing in some comma separated text. Then select the arrow styles to display the trail of links; and you're **good to go**!

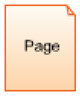


Slider

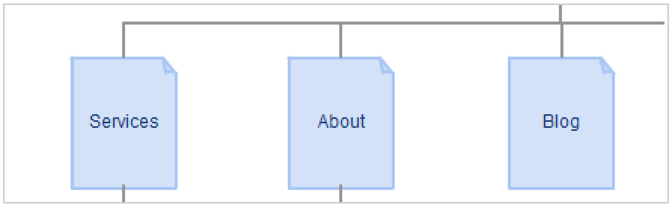
In Creately, the **Slider object's orientation** can be horizontal or vertical. You can use these slider objects to illustrate the equalizer controls, volume controls and more. Take a sneak peek at these **examples** and you'll know what I mean!


Equalizer control using Vertical Slider
+12 dB
0 dB
-12 dB
Preamp
32
64
125
250
500
1K
2K
4K
8K
16K
Music player using Horizontal Slider
CTunes - yes it plays free music.
2:50
Funky Romance - Lady BABA ft. Creately
Equalizer
volume

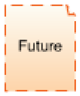
Page



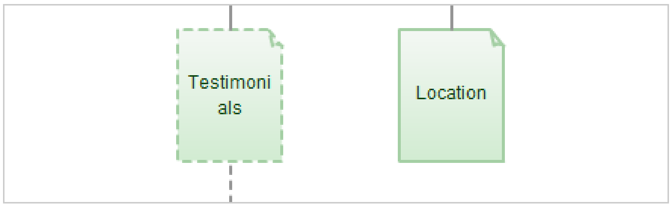
The Page shape is used to represent general page content. This can be ideally used in sitemaps and website structures to indicate any type of page (Homepage, About Us, Contact, Products, etc).



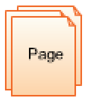
Future Page



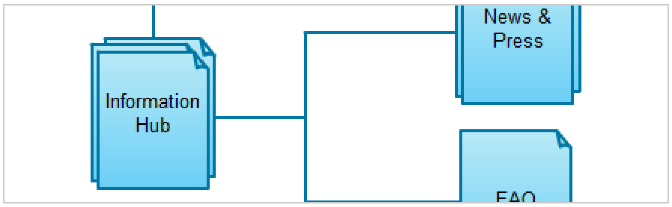
The Future page is used to show any page which is expected to be designed in the future. These pages are represented by dotted lines.



Page Cluster

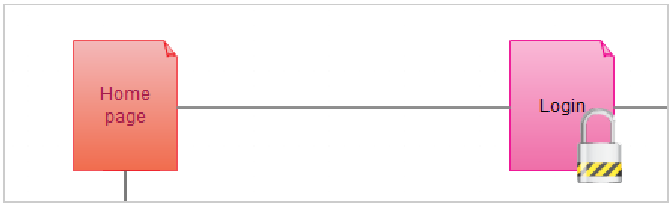


Page Cluster shape is used to represent a number of pages of the same kind. This shape can be used to illustrate content pages or article pages of the same kind.



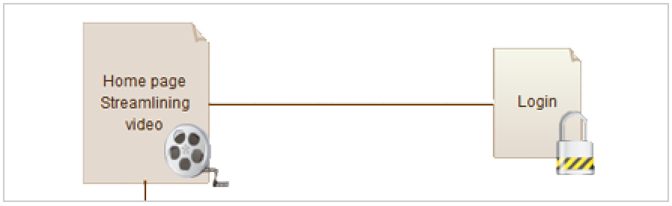
Security Indicator

Security Indicators can be used on pages to indicate secure login. When people try to visit any page on your site, there might be pages that might require login if they are not already logged in. Thus, this Security Indicator icon can be used on pages that require login.



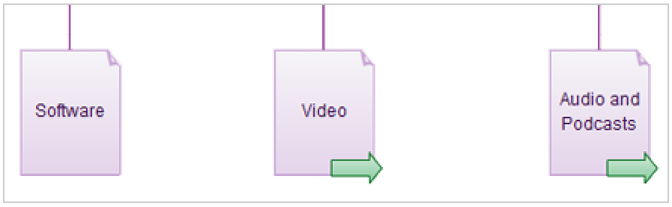
Rich Media

Rich media shape/icon can be used on any Web page that has elements which use advanced technology such as streaming video, downloaded programs that interact instantly with the user, and dynamic ads that change when the user mouse hover.



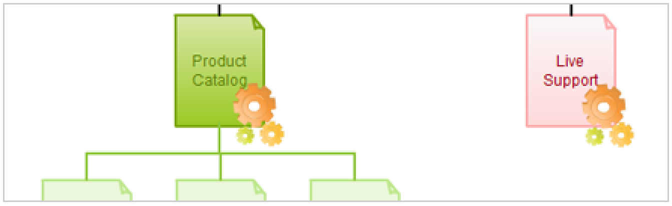
External Link

External Link Icon can be used on any page in the sitemap structure to indicate links to external web sites. For example - you can add the external link icon on your Home page, so in the sitemap structure you will know at once that the homepage contains a link to an external web site.



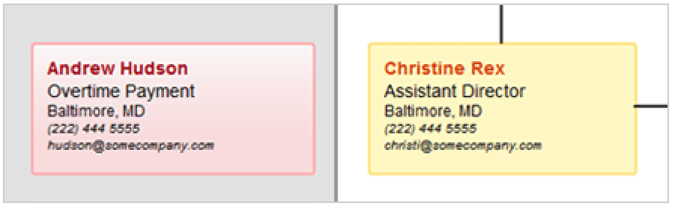
Dynamic Page

A dynamic page icon can be used on any page that has to be prepared with new content or layout for each individual viewing. These pages change dynamically with the time (news content), the user (preferences in a login session), the user location (preferences in language).



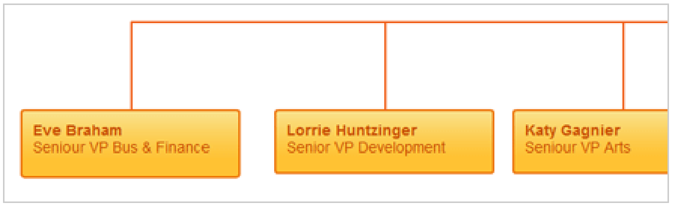
Full Detail

Full Detail shape from Creately's Organization Chart library set is a detailed rectangle built to contain all the information - the Full Name of the Employee, Job Title, Location, Phone Number and Email Address.



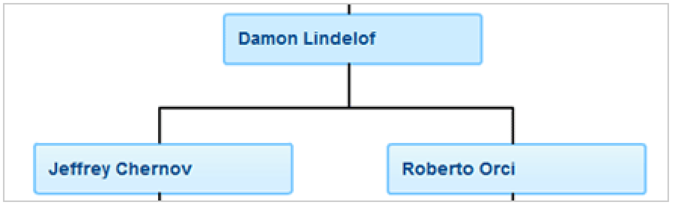
Name and Position

Name and Position shape contains only the basic information - Name and Job Title of the employee.



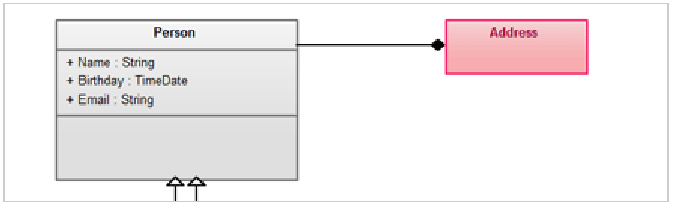
Name Only

Name Only shape contains only the Name of the employee. This is the most simple rectangle in the Organization Chart library set.



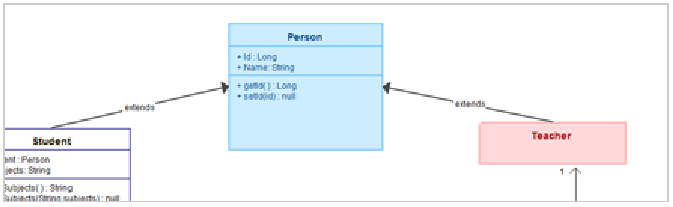
Simple Class

The core element of a **UML Class Diagram** is the class. This is a solid rectangle contains the class name.



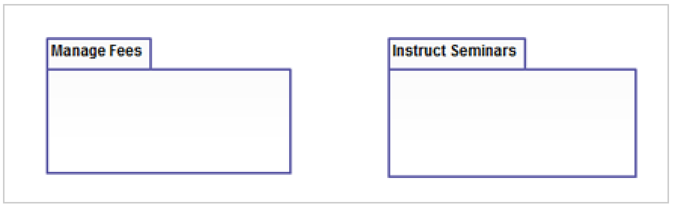
Class Object

A class object in a **UML Class Diagram** represents an entity of a given system that provides an encapsulated implementation of certain functionality of a given entity. This is a rectangle divided into three compartments. The topmost compartment contains the class name. the middle compartment contains the attributes while the lowest compartment contains the list of operations.



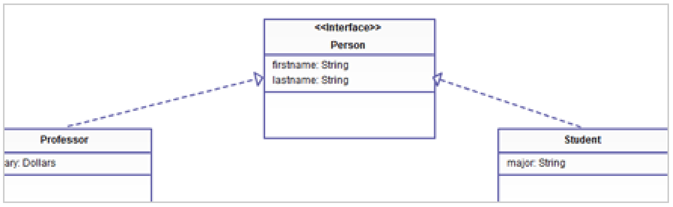
Package

A package object in a **UML Class** and **Use Case Diagram** provides the ability to group together classes and/or interfaces that are either similar in nature or related. Grouping these design elements in a package element provides for better readability of UML diagrams, especially complex diagrams.

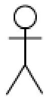


Interface

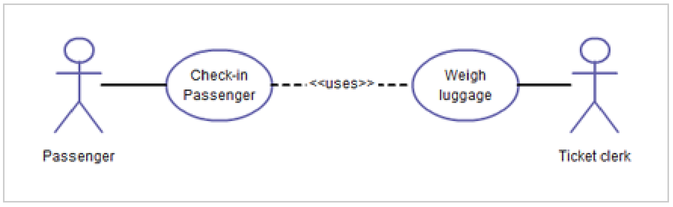
The Interface object found in a **UML Class Diagram** indicates a set of operations that would detail the responsibility of a class.



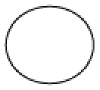
Actor



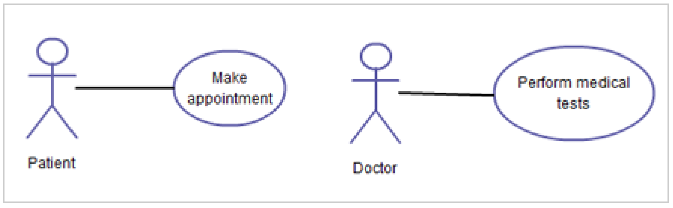
Actor in a **UML Use Case Diagram** is any entity (person, organization or external system) that performs a role in one given system. In a use case diagram, an actor interacts with a use case. For example, for modeling a reservation system, a passenger entity represents an actor in the application. Similarly, the ticket clerk who provides the service at the counter is also an actor.



Usecase

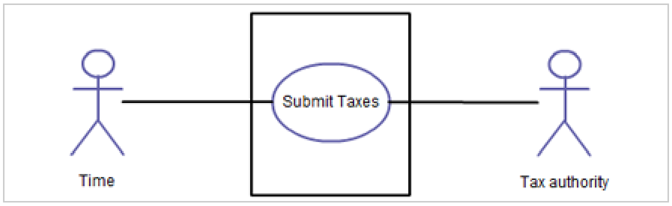


A use case in a **UML Use Case Diagram** gives a visual representation of a distinct business functionalities in a system. For example, for modeling a clinic system, the use cases will be "Make appointment" and "Perform medical tests".



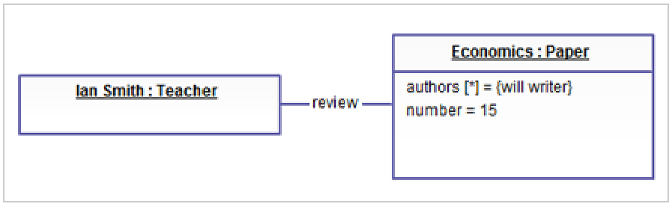
System

A system in a **UML Use Case Diagram** is a rectangle spanning all the use cases in the system that defines the scope of your system. Anything within the box represents functionality that is in scope and anything outside is not. Note that the actors in the system are outside the system.



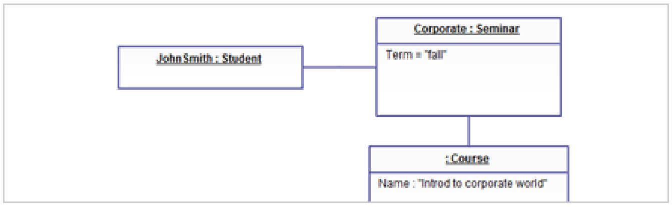
Simple Object

The simple object from the **UML Object Diagram** is a rectangle which displays the object name. This object name is usually underlined.



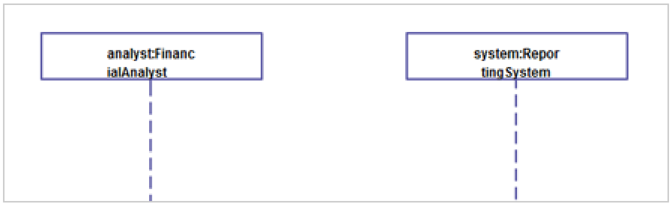
Object

The object element from the **UML Object Diagram** is a rectangle divided into two parts. The top part contains the name of the object, while the second part contains the attributes of the object. Note : This element should not be mistaken with the Class element which is divided into three parts.



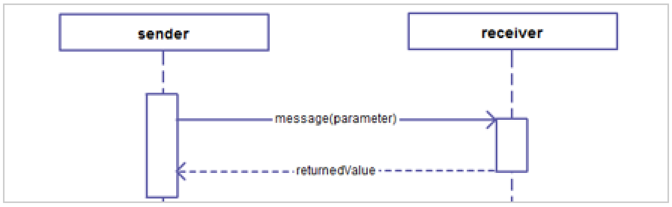
Lifeline Notation

The object notation of a **UML Sequence Diagram** is a rectangle with it's lifeline (a dashed line) descending from the center of its bottom edge. This element represents the life span of the object during the scenario being modeled.



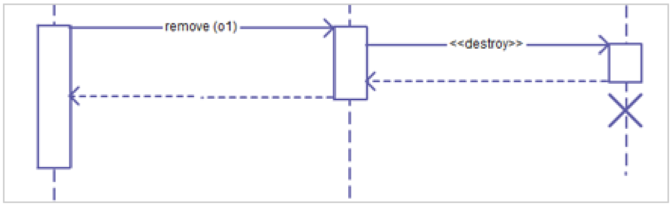
Activation

Activation elements in the **UML Sequence Diagram** are boxes on the lifelines. These are also called the method-invocation boxes, and indicate that an object is responding to a message. It starts when the message is received and ends when the object is done handling the message.



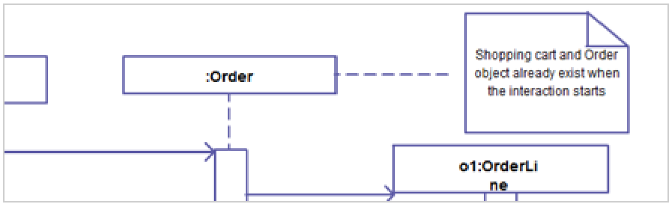
Destroy Object

Destroy object in a **UML Sequence Diagram** is a X at the bottom of an Activation box. This is a UML convention to indicate an object has been removed from memory.



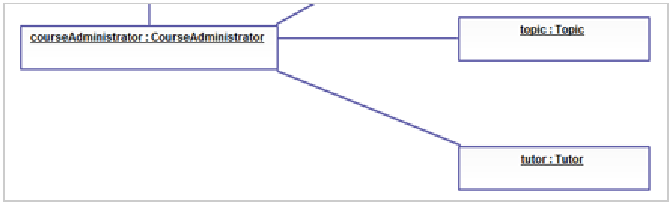
Comment

Comment object in a **UML Sequence Diagram** and **UML Activity Diagram** is shown in a rectangle with a folded-over corner. To relate the comment to any object on the diagram, the comment has to be connected to the object with dashed lines.


:Order
Shopping cart and Order object already exist when the interaction starts
o1:OrderLi ne

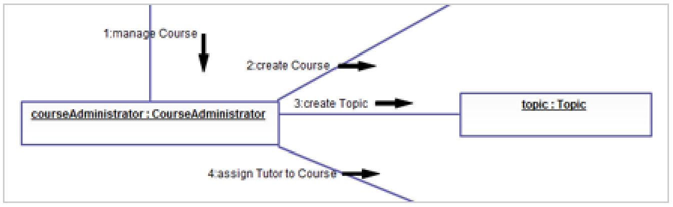
Object

The object element found in the **UML Collaboration Diagram** is a rectangle which displays the object name, preceding a colon. The object name is underlined. This shows the objects interacting with each other in the system.



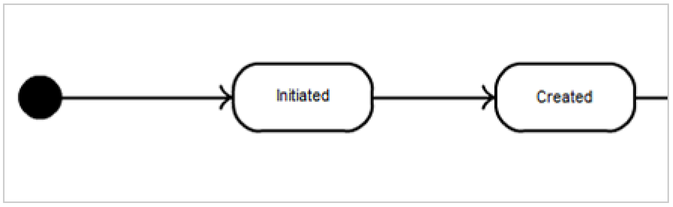
Message Arrow

Message Arrow in the **UML Collaboration Diagram** shows the interaction between the commencing object and the destination object.



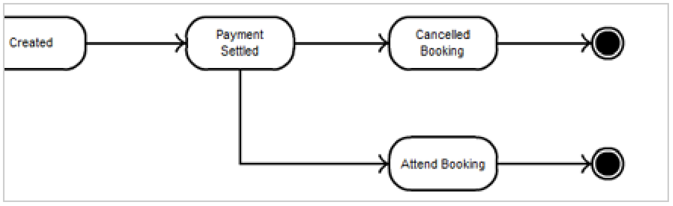
Initial State

The Initial State from the **UML Statechart Diagram** is the state of an object before any transitions. For objects, this could be the state when instantiated. The Initial State from the **UML Activity Diagram** marks the entry point and the initial Activity State. The notation for the Initial State is a small solid filled circle. There can only be one Initial State on a diagram.



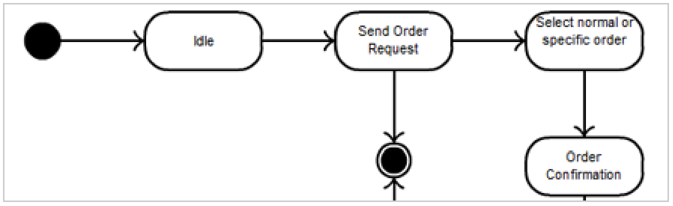
End State

End state from the **UML Statechart Diagram** marks the destruction of the object who's state we are modeling. The Activity End in a **UML Activity Diagram** shows the termination of the activity. The End notation is shown as a circle surrounding a small solid filled circle.



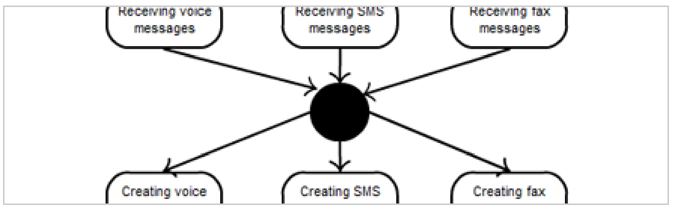
Activity

Activity state in a **UML Statechart Diagram** and **UML Activity Diagram** marks an action by an object. The notation for this is a rounded rectangle.



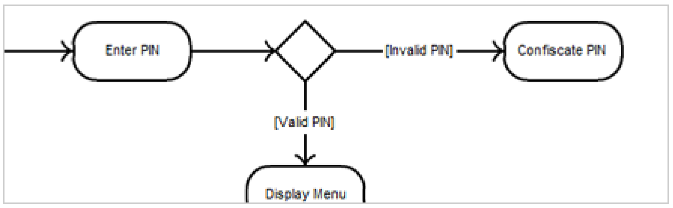
Junction

Junction state in a **UML Statechart Diagram** are vertices that are used to chain together multiple transitions. They are used to construct compound transition paths between states. A junction is represented by a small black circle.



Choice

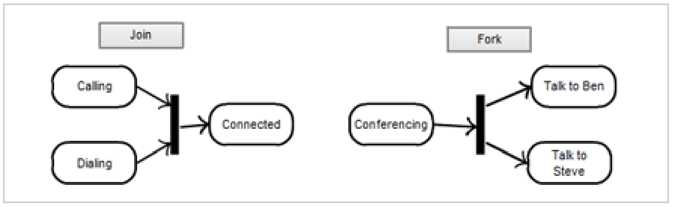
Choice state in a **UML Statechart Diagram** evaluates the guards of the triggers of its outgoing transitions to select only one outgoing transition. The decision on which path to take may be a function of the results of prior actions performed in the same run-to-completion step. A choice pseudostate is shown as a diamond-shaped symbol.



Fork / Join

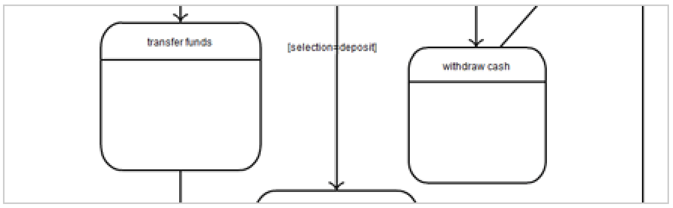
Fork vertices in the **UML Statechart Diagram** serve to split an incoming transition into two or more transitions terminating on orthogonal target vertices. The segments outgoing from a fork vertex must not have guards or triggers. Join vertices serve to merge several transitions emanating from source vertices in different orthogonal regions. The transitions entering a join vertex cannot have guards or triggers.

A Fork notation in a **UML Activity Diagram** is a control node that splits a flow into multiple concurrent flows. This will have one incoming edge and multiple outgoing edges. A join node is a control node that synchronizes multiple flows.This will have multiple incoming edges and one outgoing edge.



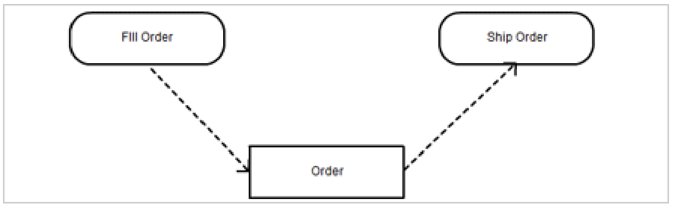
Composite State

A composite state in a **UML Statechart Diagram** is a state that has substates (nested states).



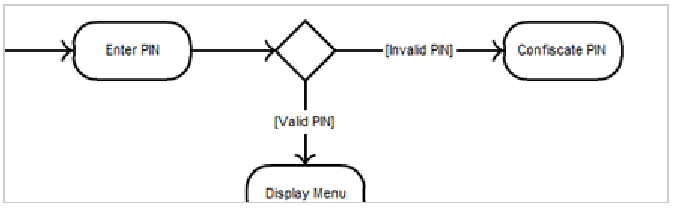
Object

The Object notation in a **UML Activity Diagram** is an activity node that is used to define the object flow in an activity.



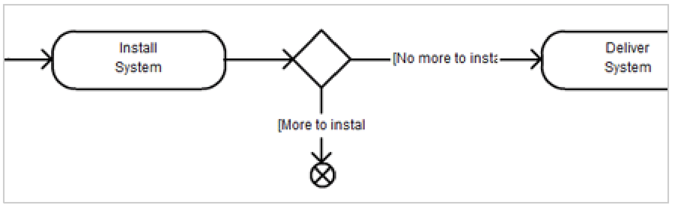
Decision

Decision notation in a **UML Activity Diagram** is a control node that accepts tokens on one or two incoming edges and selects one outgoing edge from one or more outgoing flows.



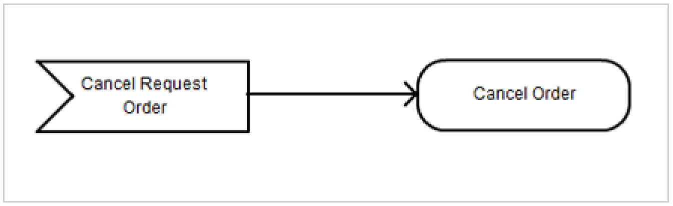
Flow End

Flow End node in a **UML Activity Diagram** is a control final node that terminates a flow. It destroys all tokens that arrive at it but has no effect on other flows in the activity. This is a small circle with a X inside.



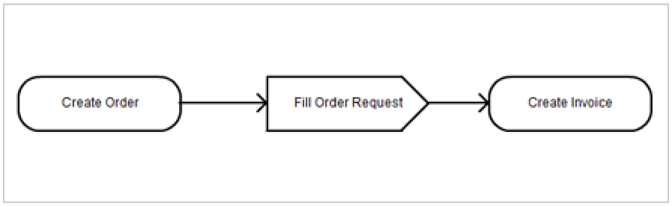
Signal Receipt

Signal Receipt notation also called the Accept event action in a **UML Activity Diagram** is an action that waits for a specific event to occur. This is drawn as a concave pentagon.

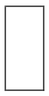


Signal Sending

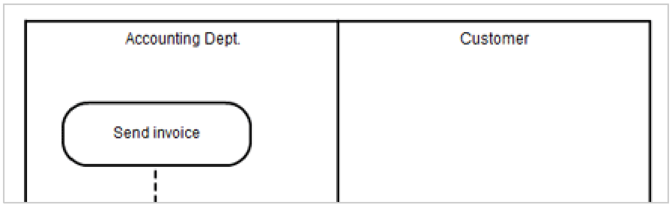
Signal Sending in **UML Activity Diagram** is an action that creates a signal instance from its inputs, and transmits it to the target object, where it may cause the firing of a state machine transition or the execution of an activity.



Activity Partition

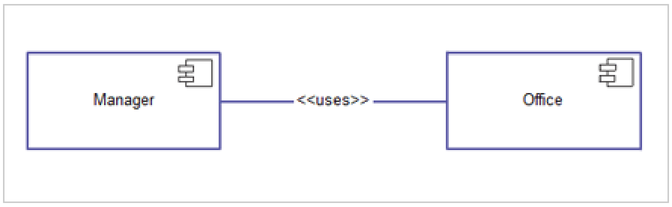


Activity Partition in a **UML Activity Diagram** is either horizontal/vertical swimlane. The partitions are used to separate actions within an activity diagram.

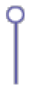


Component

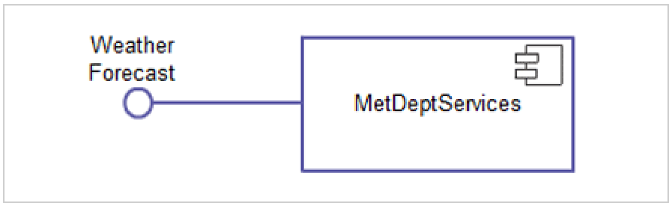
A Component **UML Component Diagram** represents a modular part of a system. A Component element in a **UML Deployment Diagram** represents a distributable piece of implementation of a system.



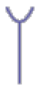
Provided Interface



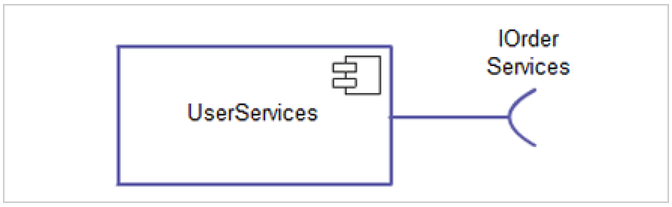
A Provided Interface of a component in a **UML Component Diagram** describes the services that the component offers to its environment. This is modeled using the lollipop notation.



Required Interface

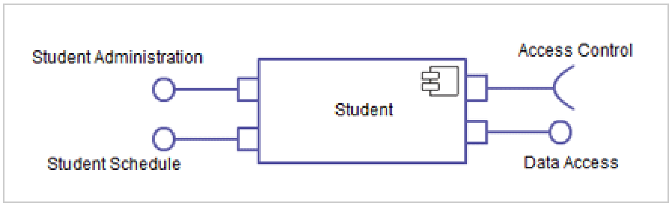


A Required Interface of a component in a **UML Component Diagram** declares the services that the component expects from its environment. This is modeled using the socket notation.



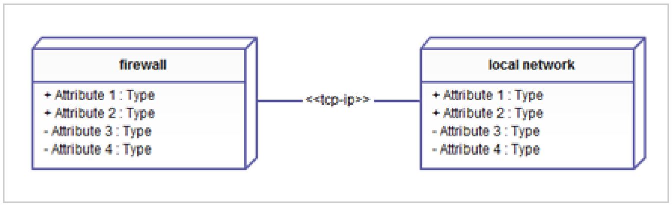
Provided Interface & Required Interface with Port

A Provided Interface with Port in a **UML Component Diagram** specifies a distinct interaction point between the component and its environment. Ports are depicted as small squares on the sides of components.



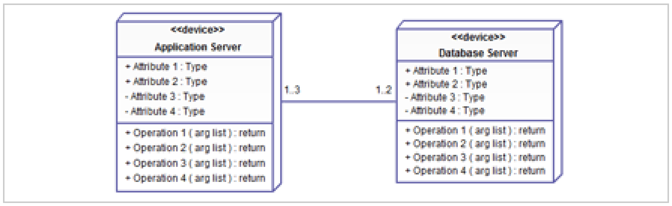
Node

A Node element in a **UML Deployment Diagram** is anything that performs work in the system. This can be either a hardware like personal computers; or a software like the operating system, database server and so forth.



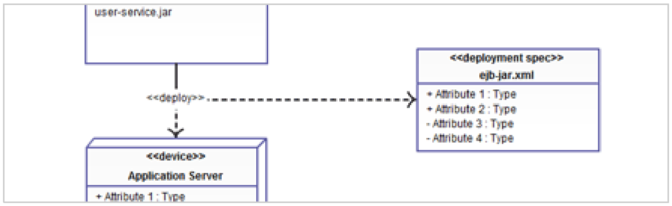
Device

A Device element in a **UML Deployment Diagram** is a type of node that represents a physical computational resource in a system, such as an application server.


<<device>>
Application Server
+ Attribute 1 : Type
+ Attribute 2 : Type
- Attribute 3 : Type
- Attribute 4 : Type
+ Operation 1 (arg list) : return
+ Operation 2 (arg list) : return
+ Operation 3 (arg list) : return
+ Operation 4 (arg list) : return
1..3
1..2
<<device>>
Database Server
+ Attribute 1 : Type
+ Attribute 2 : Type
- Attribute 3 : Type
- Attribute 4 : Type
+ Operation 1 (arg list) : return
+ Operation 2 (arg list) : return
+ Operation 3 (arg list) : return
+ Operation 4 (arg list) : return

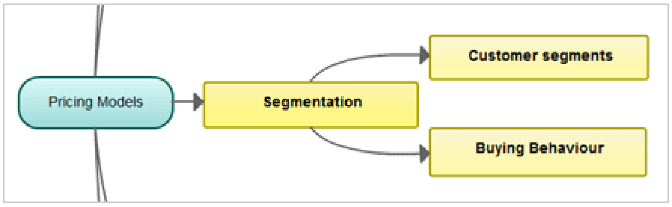
Deployment Specification

A Deployment Specification element in a **UML Deployment Diagram** is a configuration file, such as an XML document or a text file that defines how an artifact is deployed on a node.



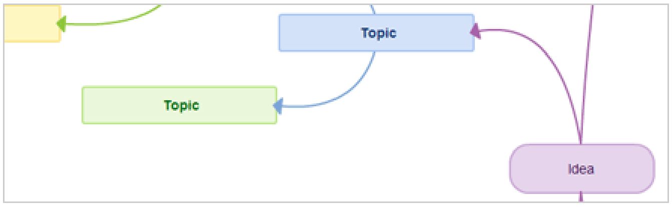
Idea

Every mind map has an idea, where the graphical representation starts from. There can be only one main idea in any mind map.



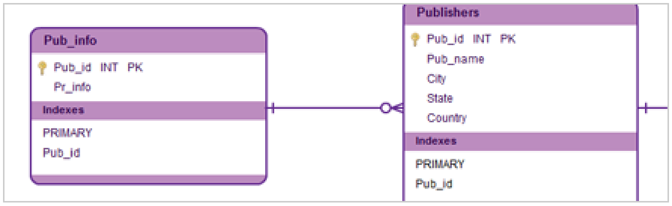
Topic

Attached to the Idea are Topics. You can have as many topics as you like on a mind map.



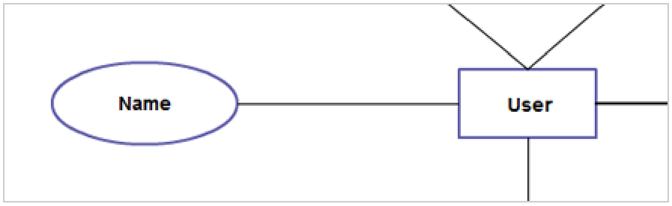
Table

Table object in database design is used to represent 'things' in the real world. real-world objects might be a customer, an inventory item or an invoice. Tables are made of rows.



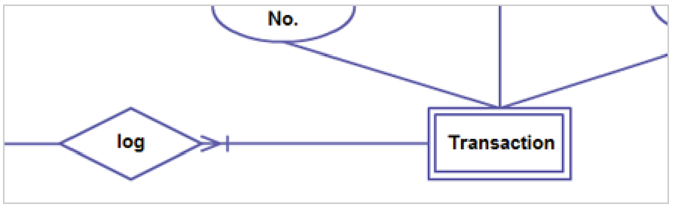
Entity

In Entity Relationship model, an entity notation is a thing of the real world which can be distinguished from other aspects of the real world. An entity may be a physical object such as a house, an event such as a house sale, or a concept such as a customer transaction.



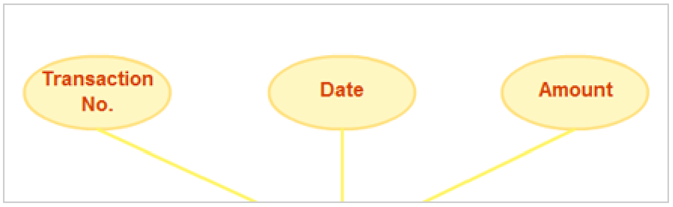
Weak Entity

Weak Entity notation cannot be uniquely identified by its attributes alone. It must use a foreign key in conjunction with its attributes to create a primary key. This notation is represented by a double-outlined rectangle. For ex., consider a database for tracking employees. Dependent persons (Jake and Tom) exist in the employee database because their father (an employee) exists in the Employee table.



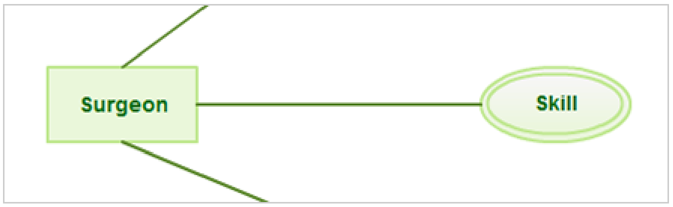
Key Attribute

A key attribute is the unique characteristic of the entity. For ex. Name and hire date are attributes of the entity Employee.



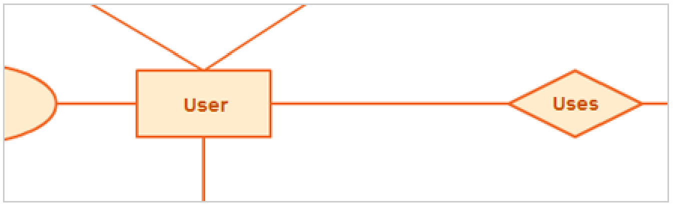
Multivalued Attribute

A multivalued attribute can have more than one value at a time for an attribute. For ex., skills of a surgeon is a multivalued attribute since a surgeon can have more than one skill.

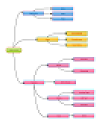


Relationship

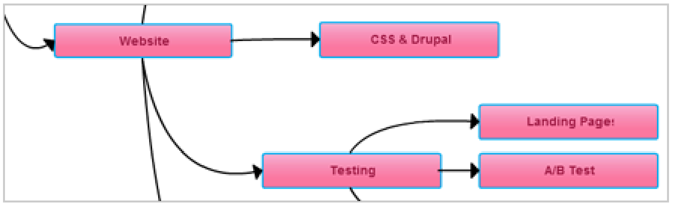
Relationships in Entity Relationship Models show how two entities share information in the database structure.



Mind Map

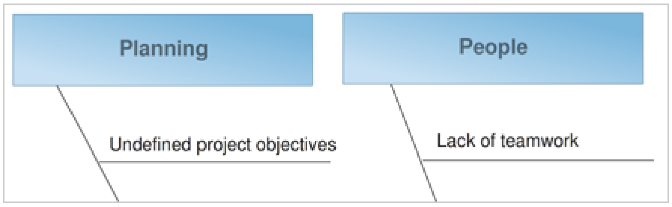


Mind Map is a graphical way of organizing information, helping students to better analyze and generate new ideas. Check this for more **Mind Map related examples and use cases**.

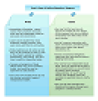


Fishbone Diagram

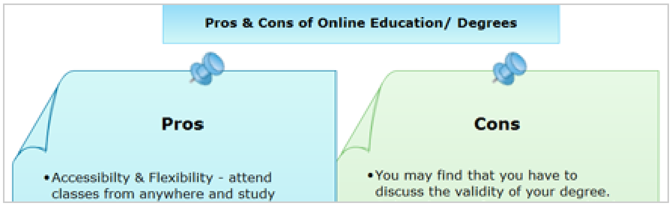
Fishbone diagram is generally used to identify the cause and effect of any given situation.



T Chart

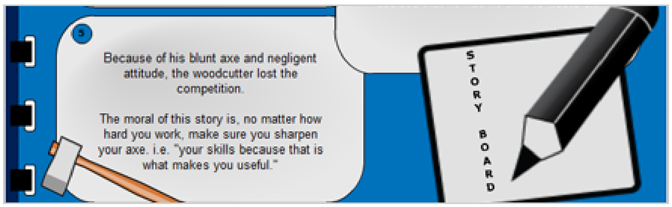


T Chart is used for listing out two different viewpoints of the same topic. For ex. The pros and cons of Online Education, the benefits and drawbacks of Social Media, and so on.

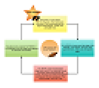

Pros & Cons of Online Education/ Degrees
Pros
Cons
•Accessibilty & Flexibility - attend classes from anywhere and study
•You may find that you have to discuss the validity of your degree.

Storyboard

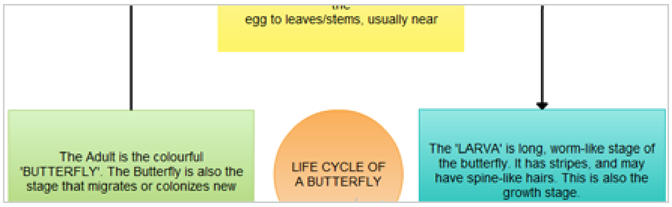
A Storyboard helps student take an idea and translate it into a visual story in a series of illustrations or images.


5
Because of his blunt axe and negligent attitude, the woodcutter lost the competition.
The moral of this story is, no matter how hard you work, make sure you sharpen your axe. i.e. "your skills because that is what makes you useful."
STORY BOARD

Cycle Diagram

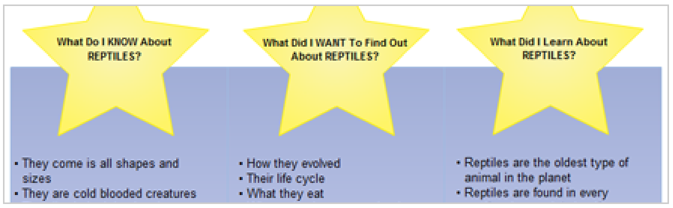


Cycle Diagram is a type of graphic organizer that shows the circular relationships between items.


egg to leaves/stems, usually near
The Adult is the colourful 'BUTTERFLY'. The Butterfly is also the stage that migrates or colonizes new
LIFE CYCLE OF A BUTTERFLY
The 'LARVA' is long, worm-like stage of the butterfly. It has stripes, and may have spine-like hairs. This is also the growth stage.

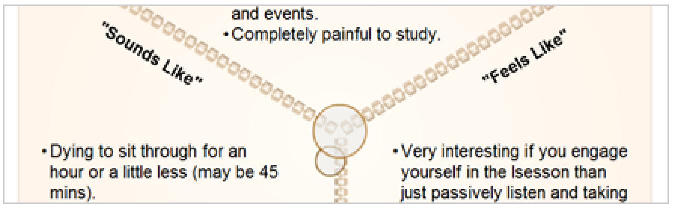
KWL Chart

A KWL (Know-Want-Learn) Chart can be used with any content area to get students think about what they **know** about a topic, what they **want to know** about that topic, and what they have **learned** at the end of the unit.


What Do I KNOW About REPTILES?
What Did I WANT To Find Out About REPTILES?
What Did I Learn About REPTILES?
• They come is all shapes and sizes
• They are cold blooded creatures
• How they evolved
• Their life cycle
• What they eat
• Reptiles are the oldest type of animal in the planet
• Reptiles are found in every

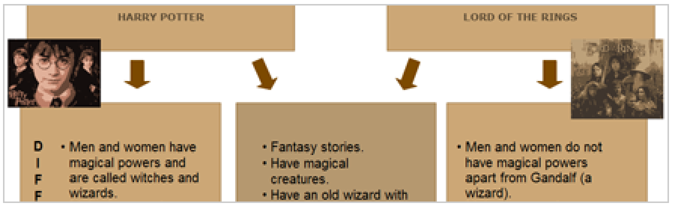
Y Chart

Y Chart is a tool that helps students organize their ideas around three dimensions.


and events.
• Completely painful to study.
"Sounds Like"
"Feels Like"
• Dying to sit through for an hour or a little less (may be 45 mins).
• Very interesting if you engage yourself in the lsesson than just passively listen and taking

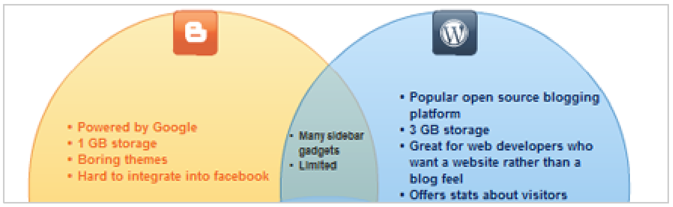
Compare & Contrast Chart

A compare and contrast chart is used to note down the similarities and differences between two pieces of literary work.

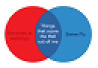

HARRY POTTER
LORD OF THE RINGS
D I F F
• Men and women have magical powers and are called witches and wizards.
• Fantasy stories.
• Have magical creatures.
• Have an old wizard with
• Men and women do not have magical powers apart from Gandalf (a wizard).

Venn

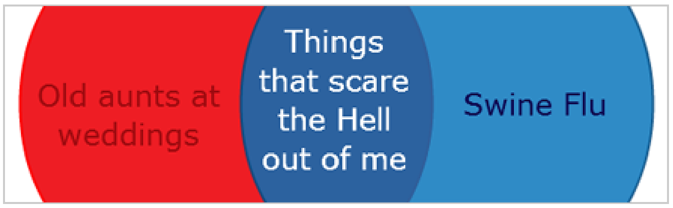
Venn diagrams show all possible logical relations between a finite collection of sets.


• Powered by Google
• 1 GB storage
• Boring themes
• Hard to integrate into facebook
• Many sidebar gadgets
• Limited
• Popular open source blogging platform
• 3 GB storage
• Great for web developers who want a website rather than a blog feel
• Offers stats about visitors

2-Set Venn

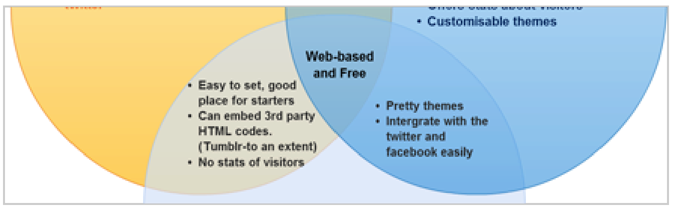


2-set Venn diagram is made up of 2 overlapping circles. And they are used to visualize the relationship between two sets.

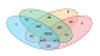


3-set Venn

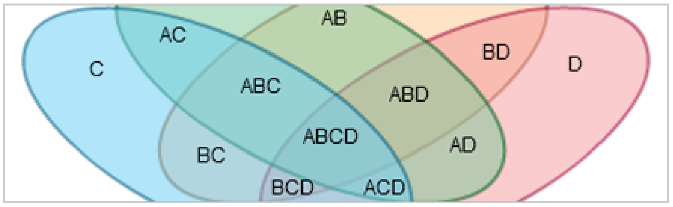
3-set Venn diagram is made up of 3 overlapping circles. And they are used to visualize the relationship between three sets.


• Customisable themes
Web-based and Free
• Easy to set, good place for starters
• Can embed 3rd party HTML codes. (Tumblr-to an extent)
• No stats of visitors
• Pretty themes
• Intergrate with the twitter and facebook easily

4-set Venn



4-set Venn diagram is drawn to visualize the relationship between four sets.



Creately

Creately Blog
Support
Terms of Service
Privacy
Resellers
Press Kit

Business / User Interface Diagrams

Online Flowchart Software
Organizational Chart Software
Mind Mapping Software
SWOT Analysis Software
Online Wireframe Software
Site Map drawing Software
Gantt Charts Drawing Software
Visio Alternative Online

Software and System Diagrams

UML Diagrams Creator
UML Sequence Diagrams
Use Case Diagrams
Class Diagrams Creator
Database Design Software
Venn Diagram Maker
Network Diagram Software
Educational Diagrams Creator

Creately Blog

15 Tools to Launch Your Startup + Bonus Tips
Why People Miss Deadlines and How to Avoid Missing Them
Achieving User Experience in Website Builders
New Creately UI Experience for Confluence and JIRA Users

More