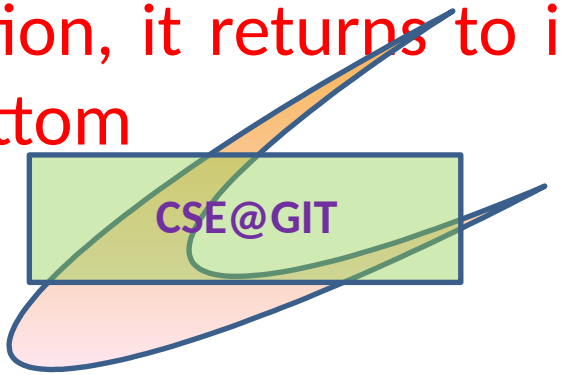
Experiment No. 3

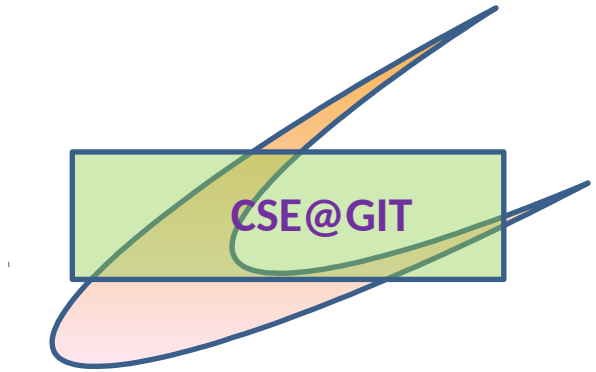
Problem Definition: a) Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.

b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom



Objectives of the Experiment:

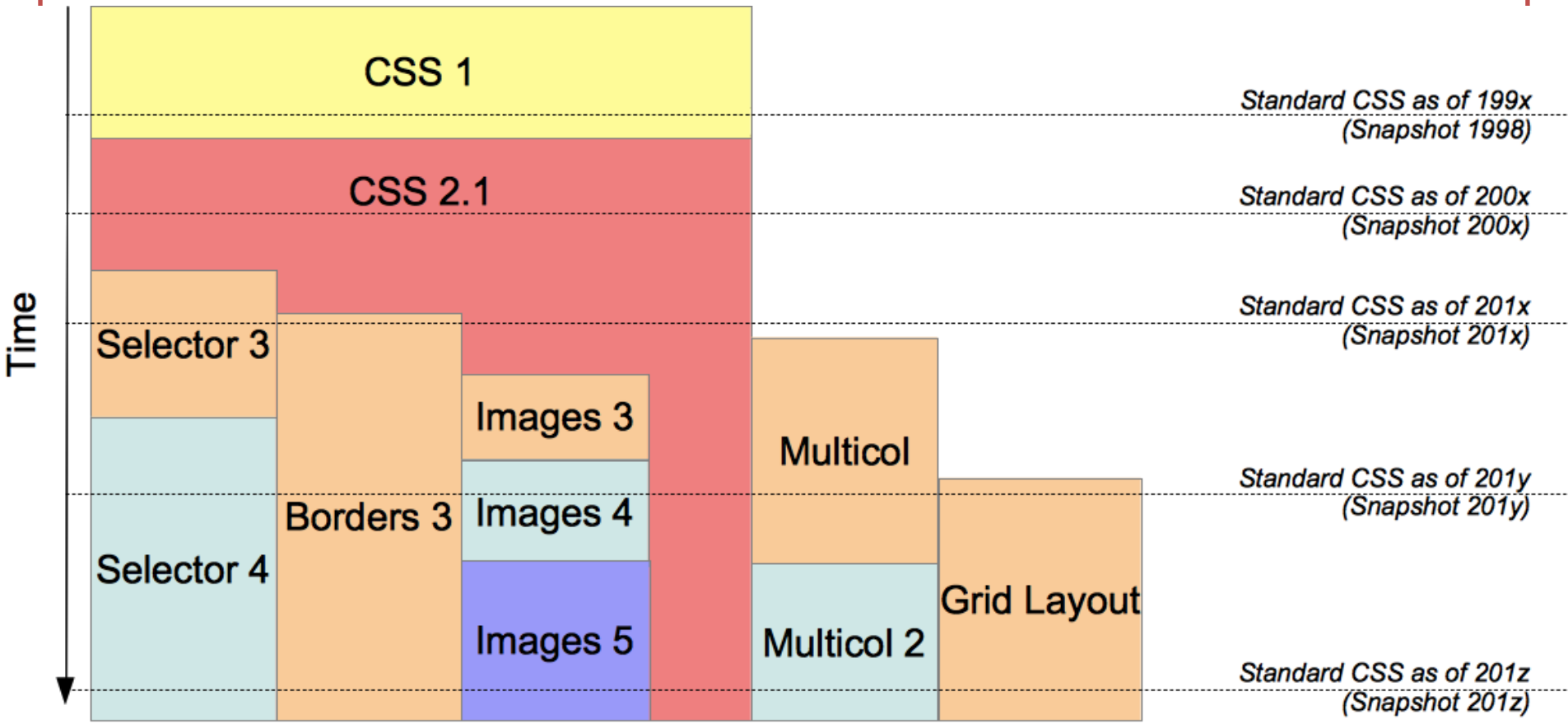
1. To demonstrate the use of CSS
2. To develop an understanding of tags in Z-index
3. To be able to build stacking of elements



Introduction

- Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML
- CSS describes how elements should be rendered on screen
- The CSS1 specification was developed in 1996
- CSS2 was released in 1998
- CSS2.1 is a recommendation
- CSS3 is the latest evolution
- CSSs provide the means to control and change presentation of HTML documents
- CSS is not technically HTML, but can be embedded in HTML documents

Introduction



Introduction

- CSS can be embedded in HTML documents
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents
- Style is specified for a tag by the values of its properties

Levels of Style Sheets

- There are three levels of style sheets
  - **Inline** - specified for a specific occurrence of a tag and apply only to that tag
    - This is fine-grain style, which defeats the purpose of style sheets - uniform style
  - **Document**-level style sheets - apply to the whole document in which they appear
  - **External** style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence
- Browser searches for a style property spec, starting with inline, until it finds one (or there isn't one)

Levels of Style Sheets

- Inline style sheets appear in the tag itself
- Document-level style sheets appear in the head of the document
- External style sheets are in separate files, potentially on any server on the Internet
- Written as text files with the MIME type text/css

Style Specification Formats

- Format depends on the level of the style sheet
- **Inline**:
  - Style sheet appears as the value of the **style** attribute
  - General form: as attribute value pair

```
style = "property_1: value_1;
         property_2: value_2;
         …
         property_n: value_n"
```

```
<element style = "property_1: value_1;
                  property_2: value_2;
                  …
                  property_n: value_n" />
```

Style Specification Formats

- Example :

```
<body style="background-color: lightblue" >

    <p style = "font-family: verdana;
                font-size: 20px;">
       Verdana; 20px
    </p>

    <p style = "font-family: Arial;
                font-size: 22px;
                color: white;">
       Arial; 22px
    </p>
```

Format for Document-level

- Style sheet appears as a list of rules that are the content of a **<style>** tag
- The **<style>** tag must include the **type** attribute, set to **"text/css"**
- The list of rules must be placed in an HTML comment, because it is not HTML
- Comments in the rule list must have a different form - use C comments (/*...*/)

Linking an External Stylesheet

- A **<link>** tag is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = "stylesheet"  type = "text/css"
  href = "http://www.wherever.org/termpaper.css">
</link>
```

- External style sheets can be validated
  https://jigsaw.w3.org/css-validator/

General Form, Document Level

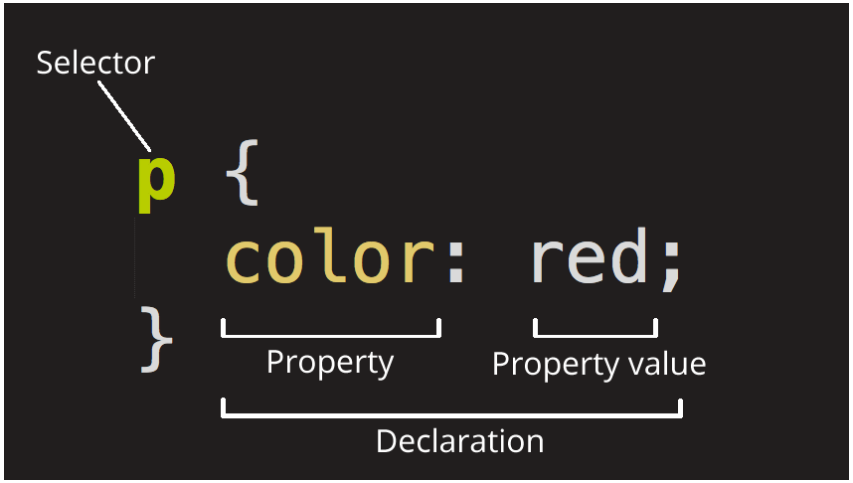
General form:

```
<style type = "text/css">
  /*
      rule list
  */
</style>
```

- Form of the rules:
  - **selector** { list of property/values}
  - Each property/value pair has the form:
    property:value
- Pairs are separated by semicolons ;

Format for Document-level

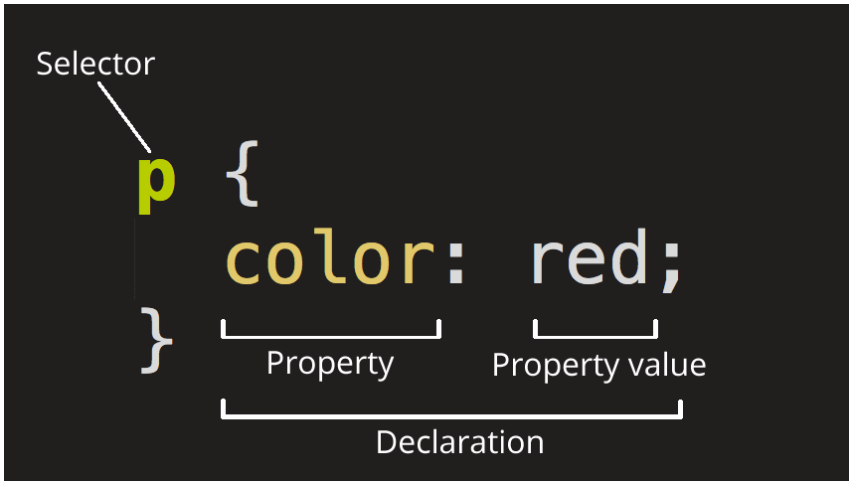
- Example : to select all the paragraph elements on an HTML page and turn the text within them red, you'd write this CSS:



[https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics]

Format for Document-level

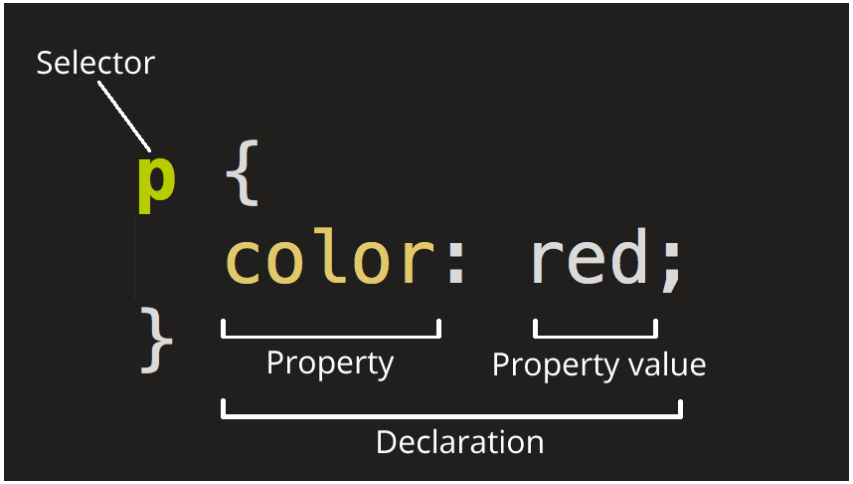
- Example : to select all the paragraph elements on an HTML page and turn the text within them red, you'd write this CSS:
  - **Selector -** HTML element name at the start of the rule set
  - It selects the element(s) to be styled (in this case, p elements)
  - To style a different element, just change the selector



[https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics]

Format for Document-level

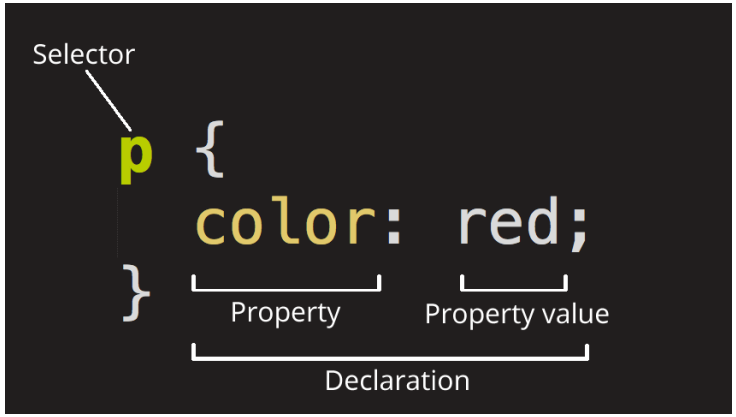
- Example : to select all the paragraph elements on an HTML page and turn the text within them red, you'd write this CSS:
  - **Declaration -** single rule like **color: red;** specifying which of the element's properties you want to style



[https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics]

Format for Document-level

- **Properties** - Ways in which you can style a given HTML element - this case, color is a property of the **<p>** elements
- You choose which properties you want to affect in your rule
- **Property value** - To the right of the property after the **colon**, we have the property value, which chooses one out of many possible appearances for a given property
- There are many color values besides red



[https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics]

Format for Document-level

- Example :

```
<style type="text/css">

  h1 {color:red;}
  p  {color:blue;}

  /* Comments */

</style>
```

Important parts of the syntax

- Each rule set (apart from the selector) must be wrapped in **curly braces {}**
- Within each declaration, you must use a **colon :** to separate the property from its values
- Within each rule set, you must use a **semicolon ;** to separate each declaration from the next one

```
p {
    color: red;
    width: 500px;
    border: 1px solid black;
}
```

Selector Forms: Simple

- The **selector** is a tag name or a list of tag names, separated by commas , example of tag names :
  - h1
  - h3
  - p

- **Selecting multiple elements** – select multiple types of elements and apply a single rule set to all of them
- Include multiple selectors separated by commas

```
p,li,h1 {
          color: red;
        }
```

Selector Forms: Simple

- The **selector** is a tag name or a list of tag names, separated by commas , example of tag names :
  - h1
  - h3
  - p
- **Contextual selectors -** Context is defined as an ancestor/descendent relationship between elements in the document tree
  - Example – apply style only to those **em** elements that are contained by an **h1** element

```
h1 { color: red }
em { color: red }
h1 em { color: blue }
```

Class Selectors

- Used to allow different occurrences of the same tag to use different style specifications
- A style class has a name, which is attached to a tag name

```
p.narrow {property/value list}
p.wide {property/value list}
```

- The class you want on a particular occurrence of a tag is specified with the **class** attribute of the tag
- For example,

```
<p class = "narrow">...</p>
<p class = "wide">...</p>
```

Different types of selector

Selector name	What does it select	Example
Element selector (sometimes called a tag or type selector)	All HTML element(s) of the specified type.	p Selects <p>
ID selector	The element on the page with the specified ID (on a given HTML page, you're only allowed one element per ID).	#my-id Selects <p id="my-id"> or <a id="my-id">
Class selector	The element(s) on the page with the specified class (multiple class instances can appear on a page).	.my-class Selects <p class="my-class"> and <a class="my-class">
Attribute selector	The element(s) on the page with the specified attribute.	img[src] Selects <img src="myimage.png"> but not <img>
Pseudo-class selector	The specified element(s), but only when in the specified state, e.g. being hovered over.	a:hover Selects <a>, but only when the mouse pointer is hovering over the link.

[https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics]

Document Object Model

- The Document Object Model (DOM) connects web pages to scripts or programming languages
- DOM model represents a document with a logical tree
- Each branch of the tree ends in a node, and each node contains objects
- DOM methods allow programmatic access to the tree; with them you can change the document's structure, style or content
- Nodes can have event handlers attached to them
- Once an event is triggered, the event handlers get execute

[https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model]

Document Object Model

- DOM specifications describe an abstract model of a document
- Interfaces describe **methods** and **properties**, tree structure
- Different languages will **bind** the interfaces to specific implementations
- The internal representation may not be tree-like
- In JavaScript, **data** are represented as properties and **operations** as methods
- DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript

Document Object Model

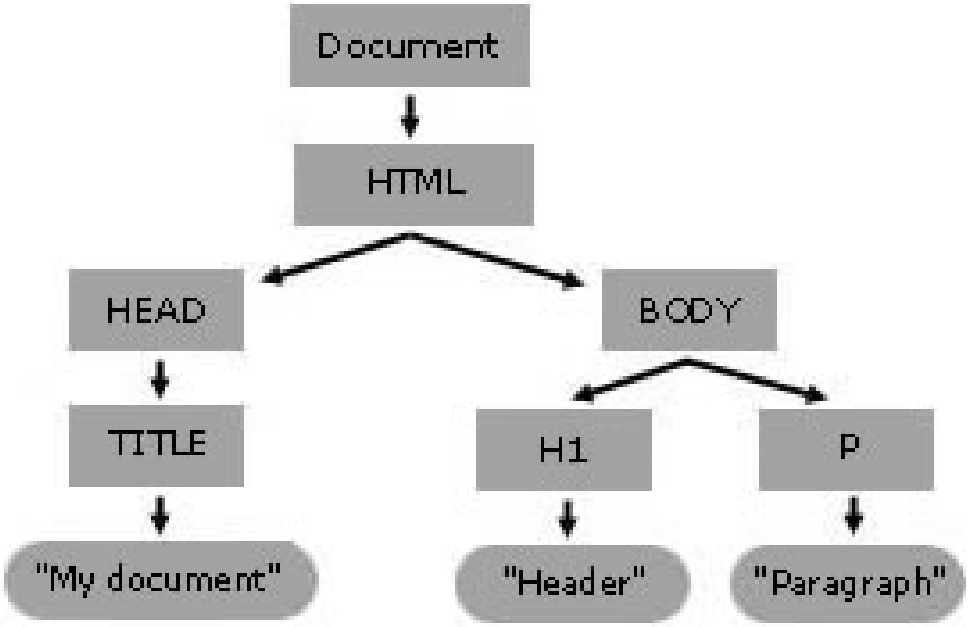
- Example : standard DOM specifies that the method : **getElementsByTagName** in the code below must return a list of all the **<p>** elements/tags in the document

```
<script>

  var paragraphs = document.getElementsByTagName('p');
  // paragraphs[0] is the first <p> element
  // paragraphs[1] is the second <p> element, etc.
  alert("Hi");
  alert(paragraphs.length);

</script>
```

[https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction]

Document Object Model

- When Mozilla parses a document, it builds a content tree and then uses it to display the document



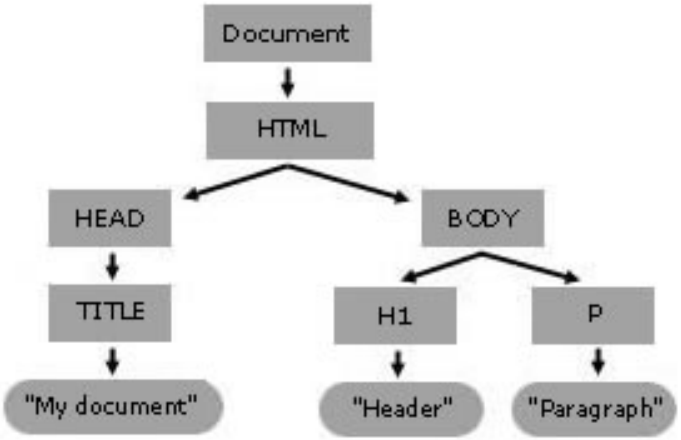
[https://developer.mozilla.org/en-US/docs/Web/API/Document_object_model/Using_the_W3C_DOM_Level_1_Core]

Document Object Model

- Suppose the script was palced in head tag, then what will be displayed by alert?

```
<script>

   var markup = document.documentElement.outerHTML;
   alert(markup);

</script>
```

- Suppose same script was placed in body tag, then?



[https://developer.mozilla.org/en-US/docs/Web/API/Element/outerHTML]

Document Object Model Levels

- DOM Levels
- DOM 0: informal, early browsers
- DOM 1: XHMTL/XML structure
- DOM 2: event model, style interface, traversal
- DOM 3: content model, validation

[https://www.w3.org/DOM/ , https://developer.mozilla.org/fr/docs/DOM_Levels]

Document Object Model Levels

- If you want to utilize full power of DOM, DOM builds succcessfully only for structured document, hence
  - For each **start** tag – there should be **close/end** tag
  - Nest properly
  - Case sensitive, <head> is **different** than <HEAD>
  - Use "" to enclose Attribute values

Document Object Model Levels – Level 1

- **DOM Level 1** specification is separated into two parts:
  - Core and
  - HTML
- **Core Level 1** provides a low-level set of fundamental interfaces that can represent any structured document (example XML document)
- **HTML Level 1** provides additional, higher-level interfaces that are used with interfaces defined in Core Level 1 to provide a more convenient view of an HTML document
- Interfaces introduced in DOM1 include Document, Node, Attr, Element, and Text interfaces
- All **interfaces** contain attributes and/or methods that can be used to interact with XML and HTML documents

Document Object Model Levels – Level 2

- **DOM Level 2** has six different specifications:
  - DOM2 Core
  - Views
  - Events
  - Style
  - Traversal and Range
  - DOM2 HTML

Document Object Model Levels – Level 2

- **DOM Level 2** has six different specifications:
  - DOM2 Core :
    - extends the functionality of the DOM1 Core
    - Contains specialized interfaces dedicated to XML
    - Introduces **getElementById**
  - Views
    - Allows programs and scripts to dynamically access and update the content of a representation of a document
    - Interfaces are AbstractView and DocumentView
  - Events
  - Style
  - Traversal and Range
  - DOM2 HTML

Document Object Model Levels – Level 2

- **DOM Level 2** has six different specifications:
  - DOM2 Core
  - Views
  - Events : Interfaces make your life easier when dealing with events, Generic event system to programs and script
    - addEventListener , handleEvent
    - EventListener, DocumentEvent, MouseEvent
    - Does not include an interface for the keyboard events, which is in later versions of the DOM
  - Style
  - Traversal and Range
  - DOM2 HTML

Document Object Model Levels – Level 2

- **DOM Level 2** has six different specifications:
  - DOM2 Core
  - Views
  - Events
  - Style : CSS, allows programs and scripts to dynamically access and update the content of style sheets
    - interfaces for Style Sheets : getComputedStyle
  - Traversal and Range : allow programs and scripts to dynamically traverse and identify a range of content in a document
    - interfaces like NodeIterator and TreeWalker
  - DOM2 HTML

Document Object Model Levels – Level 2

- **DOM Level 2** has six different specifications:
  - DOM2 Core
  - Views
  - Events
  - Style
  - Traversal and Range
  - DOM2 HTML : allows programs and scripts to dynamically access and update the content and structure of HTML documents
    - Extends the interfaces defined in the DOM1 HTML, using the DOM2 Core possibilities

Document Object Model Levels – Level 3

- **DOM Level 3** contains five different specifications:
  - DOM3 Core
  - Load and Save
  - Validation
  - Events
  - Xpath

[https://developer.mozilla.org/fr/docs/DOM_Levels ,
Author: Fabian Guisset]

Document Object Model Levels – Level 3

- **DOM Level 3** contains five different specifications:
  - DOM3 Core : extend the functionality of the DOM1 and DOM2 Core specs
    - methods and properties like adoptNode() and textContent
  - Load and Save : allows programs and scripts to dynamically load the content of an XML document into a DOM document, and serialize a DOM document into an XML document
  - Validation
  - Events
  - Xpath

Document Object Model Levels – Level 3

- **DOM Level 3** contains five different specifications:
  - DOM3 Core
  - Load and Save
  - Validation : allows programs and scripts to dynamically update the content and the structure of documents while ensuring that the document remains valid, or to ensure that the document becomes valid
  - Events : extension of the DOM2 Events specification, keyboard events and how to handle them
  - Xpath : provides simple functionalities to access a DOM tree using XPath 1.0.

Events and Event Handling

- **Event-driven programming** is a style of programming in which pieces of code, **event handlers**, are written to be activated when certain events occur
- Events represent activity in the environment including, especially, user actions such as moving the mouse or typing on the keyboard
- Events are fired inside the browser window, and tend to be attached to a specific item that resides in it
- Might be a single element or set , HTML document loaded in the current tab, or the entire browser window

Events and Event Handling

- Examples of events :
  - The user clicking the mouse over a certain element, or hovering the cursor over a certain element
  - The user pressing a key on the keyboard
  - The user resizing or closing the browser window
  - A web page finishing loading
  - A form being submitted
  - A video being played, or paused, or finishing play
  - An error occuring

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events]

Events and Event Handling

- Each event has an **event handler**, which is block of code (usually a user-defined JavaScript function) that will be run when the event fires
- Events are represented by JavaScript objects
- When such a block of code is defined to be run in response to an event firing, we say we are **registering an event handler**
- Event handlers are sometimes called event listeners
- Assign an event attribute an event handler
- Assign a DOM node an event handler

Events, Attributes and Tags

Event	*Tag Attribute*
blur	onblur
change	onchange
click	onclick
focus	onfocus
load	onload
mousedown	onmousedown
mousemove	onmousemove
mouseout	onmouseout
mouseover	onmouseover
mouseup	onmouseup
select	onselect
submit	onsubmit
unload	onunload

Events and Event Handling

- What will be displayed ?

```
<html>
 <head>
  <title>Button click event</title>

 </head>
 <body>

   <button>
      Click for change !
   </button>

 </body>
</html>
```

Events and Event Handling

- Following changes are made to button, now what will happen ?

```
<button onclick="whatDoesThisFunctionDo()">
   Click for change !
</button>
```

Events and Event Handling

- And if the function call is :

```
<button onclick="whatDoesThisFunctionDo()">
   Click for change !
</button>

<script>
 function random(number)
  {
    return Math.floor(Math.random()*(number+1));
  }
 function whatDoesThisFunctionDo()
  {
    var randomColour = 'rgb(' + random(255) + ','
              + random(255) + ',' + random(255) + ')' ;
    document.body.style.backgroundColor = randomColour;
  }
</script>
```

Element Positioning

- CSS provides tools to position elements in a web page
- Property **position** specifies the position mode
- **Value** is absolute, relative or static

```
.className
  {
    position: relative;
  }
```

- Properties **left**, **right**, **top** and **bottom** specify element position

```
#id
  {
    position: absolute;
    top: 20px;
    right: -20px;
  }
```

Element Positioning

- A positive value of top pushes the element down
- A positive value of left pushes the element to the right
- A negative value of top pushes the element up
- A negative value of left pushes the element to the left
- Absolute position specifies where an element appears relative to the containing element
- Final location of positioned elements is determine based on position and values of top or bottom, and left or right

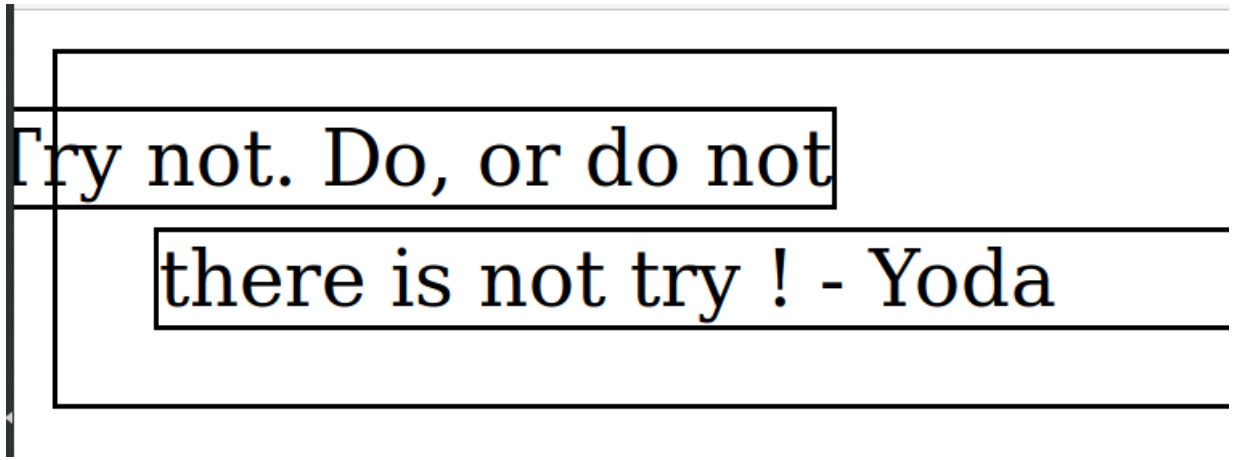
[CSS basics - Learn web development | MDN :
https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics
position - CSS | MDN :
https://developer.mozilla.org/en-US/docs/Web/CSS/position]

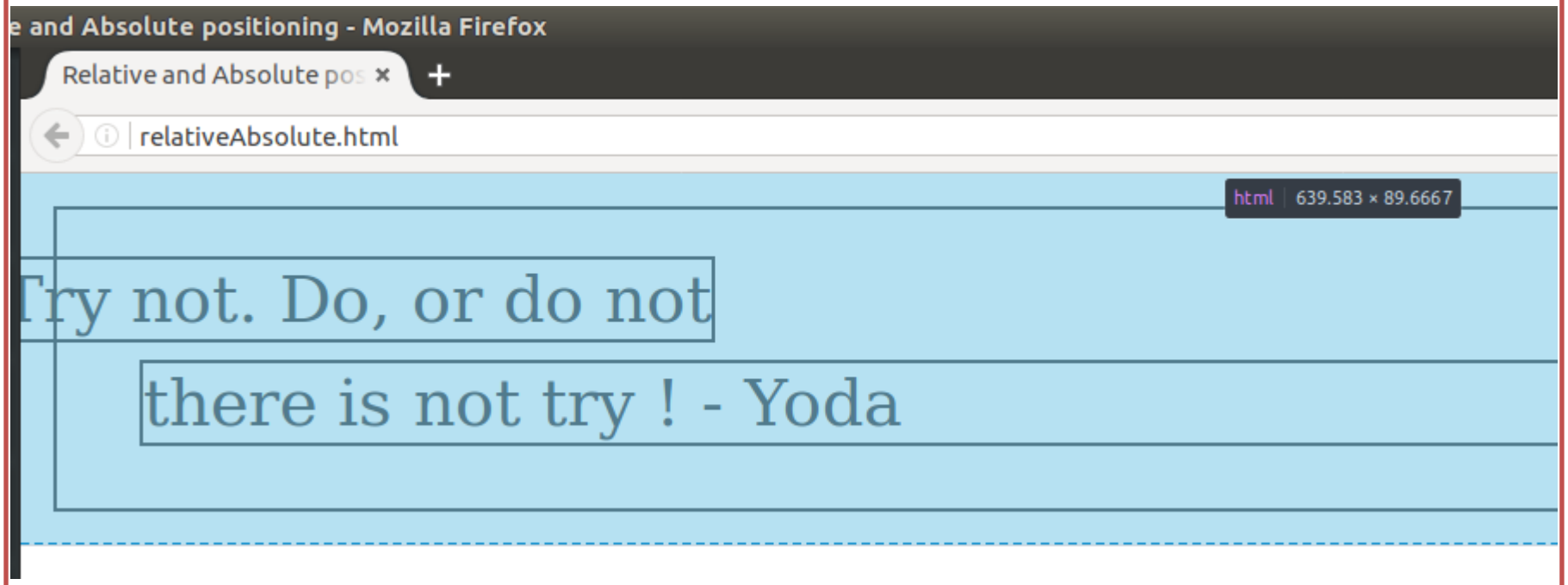
Absolute and Relative Positioning

```
.className
  {
    position: relative;
    top: 20px;
    left: 20px;
  }
#id
  {
    position: absolute;
    top: 20px;
    left: -6px;
  }
body, p
  {
    border:1px solid;
  }
```

```
<body>
  <p id="id"> Try not. Do, or do not  </p>
  <p class="className"> there is not try !  - Yoda </p>
</body>
```

Absolute / Relative Positioning



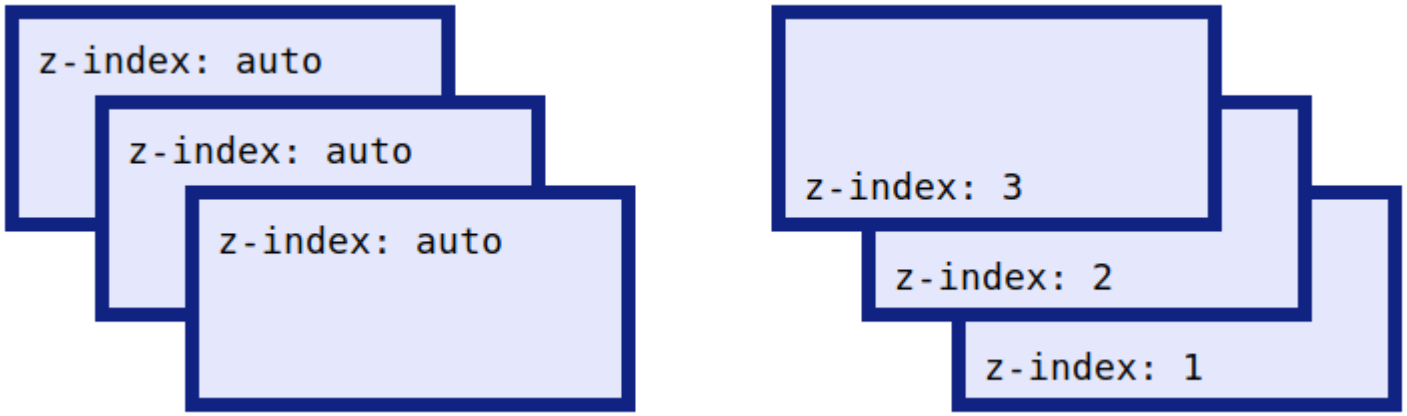


Stacking Elements

- Property = **z-index**
- The z-index style property can be used to govern the layering of elements in the display
- If two elements both cover a part of the window, the element with the higher z-index value will cover the other one
- Think of a artist painting the document content on the screen.
- Elements with lower z-index are painted before those with higher z-index
- Manipulating the z-index property dynamically
- Absolute position specifies where an element appears relative to the containing element

Stacking Elements

- Effect of z-index
- On the left three boxes have been overlaped using absolute positioning
- By default, the elements are stacked following the order they're declared in the HTML
- On the right the same markup, but have reversed the default order using z-index **property** and assigning values



[https://developer.mozilla.org/en-US/docs/Web/CSS/z-index]

Z-Index

```
<html>
 <head>
  <title>z-index</title>
 </head>
 <body>
  <img src="U_R_Rao.jpg" />
  <img src="Atoms_for_Peace.jpg"  />
 </body>
</html>
```

[Author HPNadig] [Author IAEA Imagebank]

Z-Index

```
<html>
 <head>
  <title>z-index</title>
 </head>
 <body>
  <img src="U_R_Rao.jpg" style="position:absolute;
                                top:40px;left:40px;"/>
  <img src="Atoms_for_Peace.jpg"  />
 </body>
</html>
```

Z-Index

```
<html>
 <head>
  <title>z-index</title>
 </head>
 <body>
  <img src="U_R_Rao.jpg" style="position:absolute;
            top:40px;left:40px;"/>
  <img src="Atoms_for_Peace.jpg" style="position:absolute;
            top:120px;left:120px;"/>
 </body>
</html>
```

Z-Index

```
<html>
 <head>
  <title>z-index</title>
 </head>
 <body>
  <img src="U_R_Rao.jpg" style="position:absolute;
            z-index:1;  top:40px;left:40px;"/>
  <img src="Atoms_for_Peace.jpg" style="position:absolute;
           top:120px;left:120px;"/>
 </body>
</html>
```

Z-Index

```
<html>
 <head>
  <title>z-index</title>
 </head>
 <body>
  <img src="U_R_Rao.jpg" style="position:absolute;
            z-index:1;  top:40px;left:40px;"/>
  <img src="Atoms_for_P
           top:120px;le
 </body>
</html>
```

3a. Pseudo Code / Outline of the Algorithm

```
p
 {
   border:solid thick black;
   padding:10px;
   width:300px;
   height:200px;
   position:absolute;
 }
.plane1
 {
   background-color:red;
   top:100px;
   left:200px;
   z-index:0;
 }
```

```
.plane2
 {
   background-color:green;
   top:140px;
   left:220px;
   z-index:0;
 }
.plane3
 {
   background-color:yellow;
   top:180px;
   left:240px;
   z-index:0;
 }
```

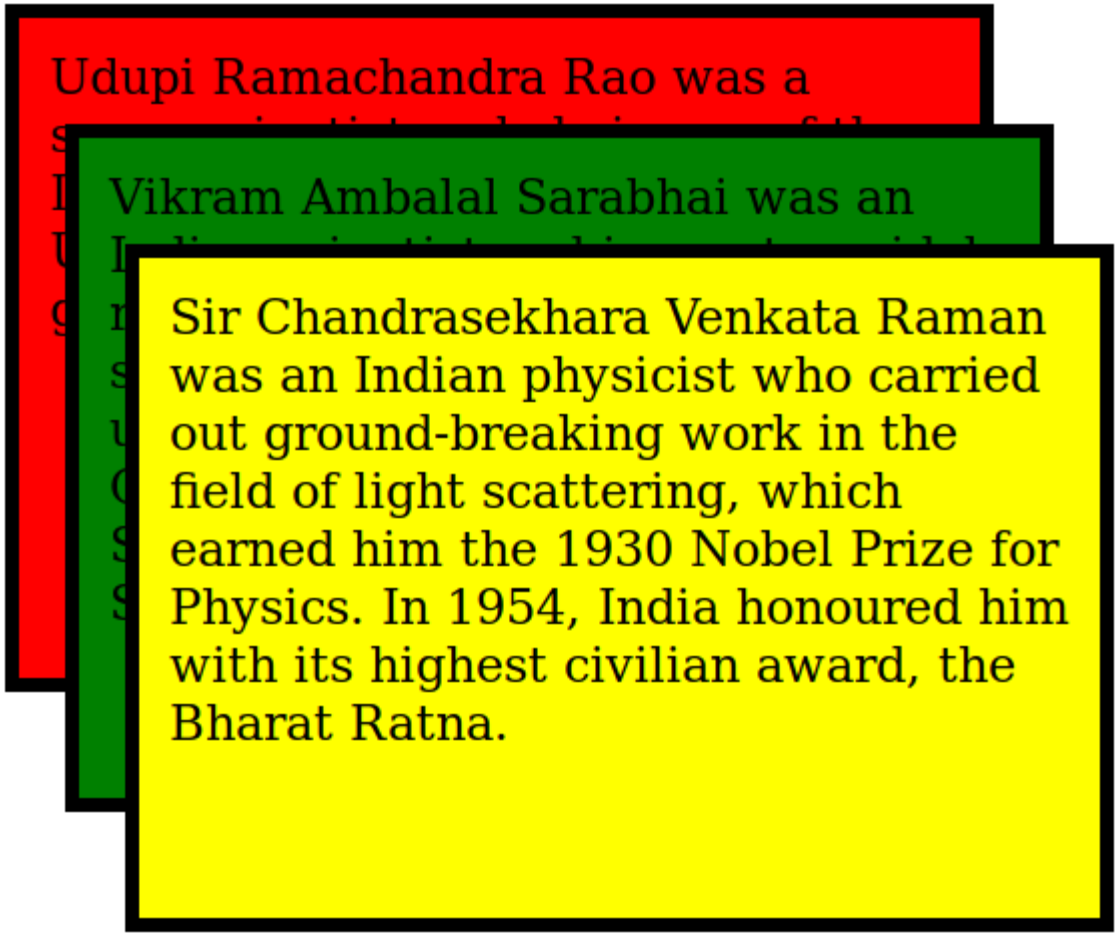
3a. Pseudo Code / Outline of the Algorithm

```
<script type="text/javascript">
  var topLayer="layer3";
  function mover(toTop)
   {
     document.getElementById(topLayer).style.zIndex="5";
     document.getElementById(toTop).style.zIndex="10";
     topLayer=toTop;
   }
</script>
```

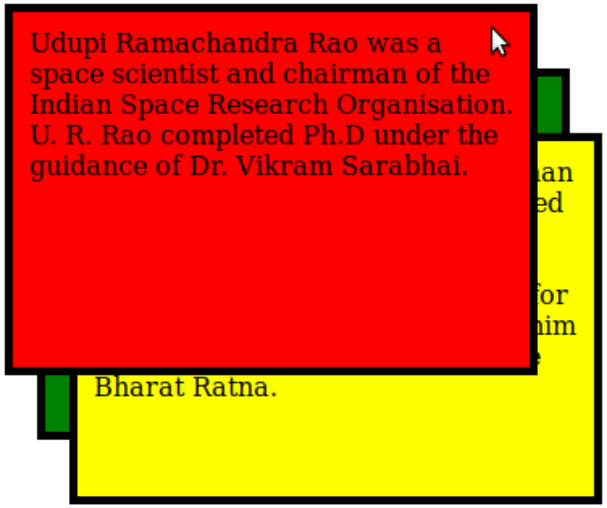
3a. Pseudo Code / Outline of the Algorithm

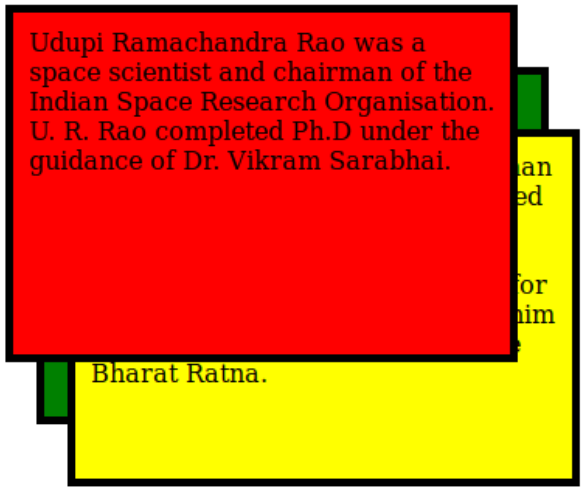
```
<p class="plane1" id="layer1" onMouseOver="mover('layer1');">
Udupi Ramachandra Rao was a space scientist and chairman of the
Indian Space Research Organisation. U. R. Rao completed Ph.D under
the guidance of Dr. Vikram Sarabhai. </p>

<p class="plane2" id="layer2" onMouseOver="mover('layer2');">
Vikram Ambalal Sarabhai was an Indian scientist and innovator widely
regarded as the father of India's space programme, completed Ph.D
under the guidance of Dr. Chandrasekhara Venkata Raman. Sarabhai
received the Shanti Swarup Bhatnagar Medal in 1962. </p>

<p class="plane3" id="layer3" onMouseOver="mover('layer3');">Sir
Chandrasekhara Venkata Raman was an Indian physicist who carried out
ground-breaking work in the field of light scattering, which earned
him the 1930 Nobel Prize for Physics. In 1954, India honoured him with
its highest civilian award, the Bharat Ratna. </p>
```

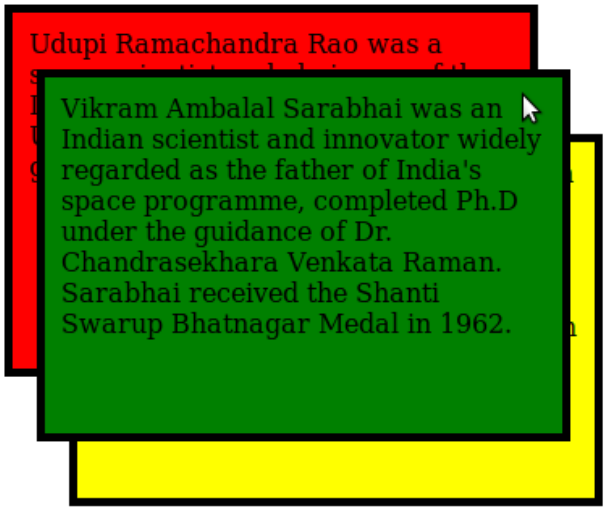
Sample Run

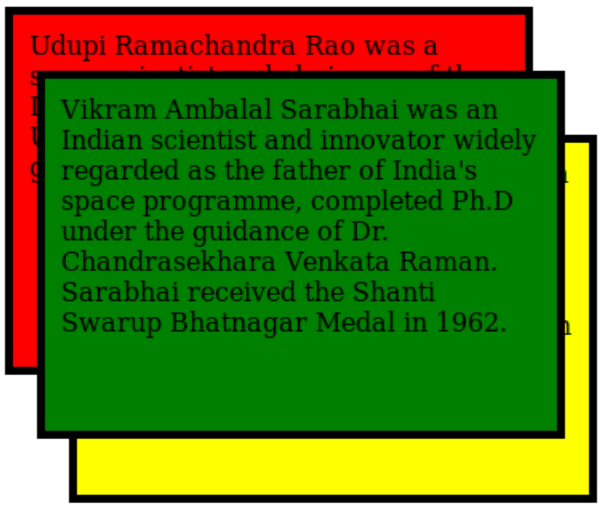

Udupi Ramachandra Rao was a
Vikram Ambalal Sarabhai was an
Sir Chandrasekhara Venkata Raman was an Indian physicist who carried out ground-breaking work in the field of light scattering, which earned him the 1930 Nobel Prize for Physics. In 1954, India honoured him with its highest civilian award, the Bharat Ratna.

Sample Run


Udupi Ramachandra Rao was a space scientist and chairman of the Indian Space Research Organisation. U. R. Rao completed Ph.D under the guidance of Dr. Vikram Sarabhai.
Bharat Ratna.


Udupi Ramachandra Rao was a space scientist and chairman of the Indian Space Research Organisation. U. R. Rao completed Ph.D under the guidance of Dr. Vikram Sarabhai.
Bharat Ratna.


Udupi Ramachandra Rao was a
Vikram Ambalal Sarabhai was an Indian scientist and innovator widely regarded as the father of India's space programme, completed Ph.D under the guidance of Dr. Chandrasekhara Venkata Raman. Sarabhai received the Shanti Swarup Bhatnagar Medal in 1962.


Udupi Ramachandra Rao was a
Vikram Ambalal Sarabhai was an Indian scientist and innovator widely regarded as the father of India's space programme, completed Ph.D under the guidance of Dr. Chandrasekhara Venkata Raman. Sarabhai received the Shanti Swarup Bhatnagar Medal in 1962.

3b. Pseudo Code / Outline of the Algorithm

```
p
 {
   border:solid black;
   position:absolute;
   padding:10px;
   width:250px;
   height:200px;
 }
.plane1
 {
   background-color:red;
   top:100px;
   left:200px;
   z-index:1;
 }
.plane2
 {
   background-color:green;
   top:140px;
   left:220px;
   z-index:2;
 }
.plane3
 {
   background-color:yellow;
   top:180px;
   left:240px;
   z-index:3;
 }
```

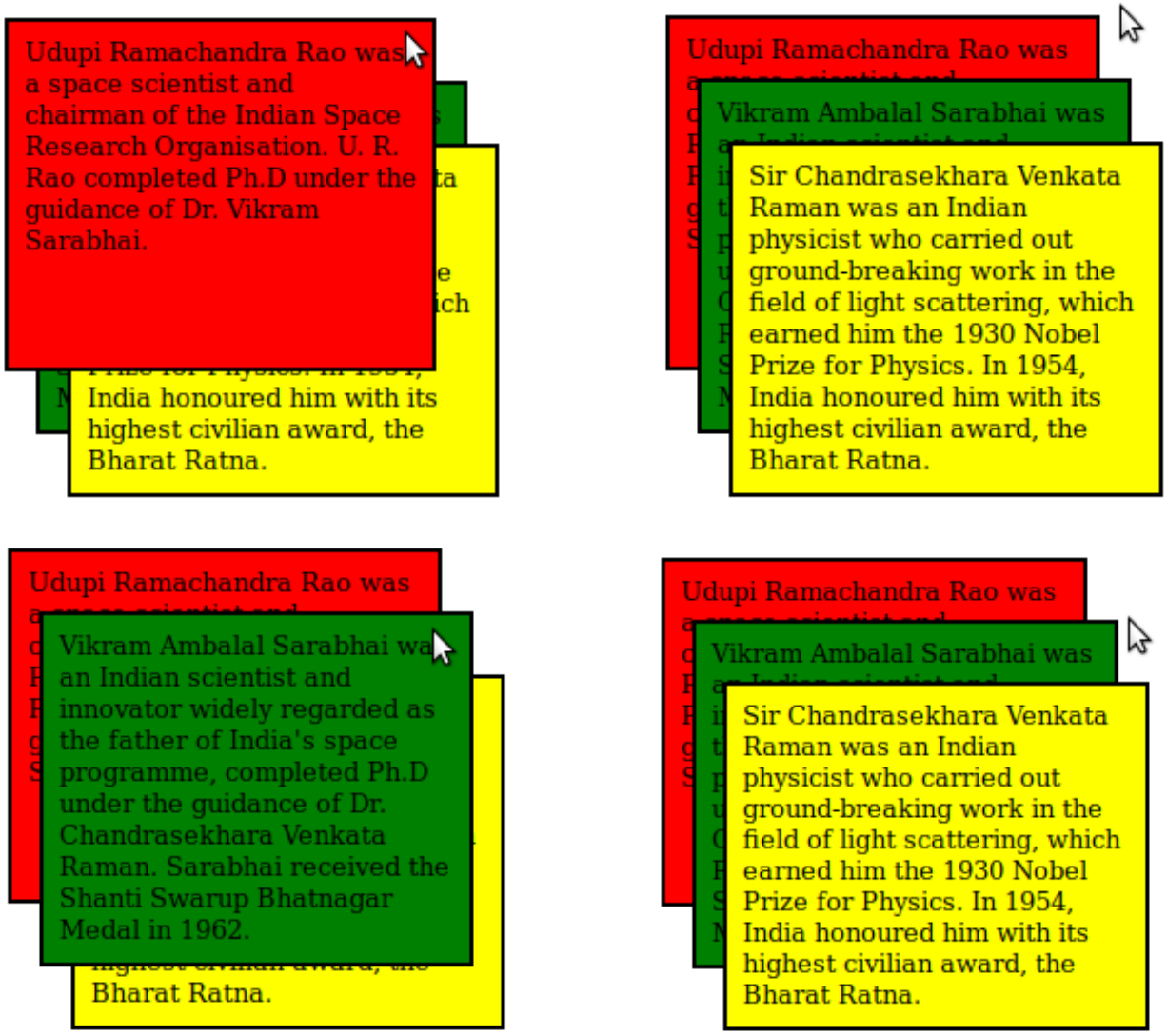
3b. Pseudo Code / Outline of the Algorithm

```
<script type="text/javascript">
  var topLayer="layer3";
  var origpos;
  function mover(toTop,pos)
    {
      document.getElementById(toTop).style.zIndex="4";
      topLayer=toTop;
      origpos=pos;
    }
  function moveBack()
    {
      document.getElementById(topLayer).style.zIndex=origpos;
    }
</script>
```

3b. Pseudo Code / Outline of the Algorithm

```
<p class="plane1" id="layer1" onMouseOver="mover('layer1','1');"
                              onMouseOut="moveBack();" />
Udupi Ramachandra Rao was a space scientist and chairman of the
Indian Space Research Organisation. U. R. Rao completed Ph.D under
the guidance of Dr. Vikram Sarabhai. </p>

<p class="plane2" id="layer2" onMouseOver="mover('layer2','2');"
                              onMouseOut="moveBack();" />
Vikram Ambalal Sarabhai was an Indian scientist and innovator widely
regarded as the father of India's space programme, completed Ph.D
under the guidance of Dr. Chandrasekhara Venkata Raman. Sarabhai
received the Shanti Swarup Bhatnagar Medal in 1962. </p>

<p class="plane3" id="layer3" onMouseOver="mover('layer3','3');"
                              onMouseOut="moveBack();" />
Sir Chandrasekhara Venkata Raman was an Indian physicist who carried
out ground-breaking work in the field of light scattering, which
earned him the 1930 Nobel Prize for Physics. In 1954, India honoured
him with its highest civilian award, the Bharat Ratna. </p>
```

Sample Run


Udupi Ramachandra Rao was a space scientist and chairman of the Indian Space Research Organisation. U. R. Rao completed Ph.D under the guidance of Dr. Vikram Sarabhai.
India honoured him with its highest civilian award, the Bharat Ratna.
Udupi Ramachandra Rao was
Vikram Ambalal Sarabhai was
Sir Chandrasekhara Venkata Raman was an Indian physicist who carried out ground-breaking work in the field of light scattering, which earned him the 1930 Nobel Prize for Physics. In 1954, India honoured him with its highest civilian award, the Bharat Ratna.
Udupi Ramachandra Rao was
Vikram Ambalal Sarabhai was an Indian scientist and innovator widely regarded as the father of India's space programme, completed Ph.D under the guidance of Dr. Chandrasekhara Venkata Raman. Sarabhai received the Shanti Swarup Bhatnagar Medal in 1962.
Bharat Ratna.
Udupi Ramachandra Rao was
Vikram Ambalal Sarabhai was
Sir Chandrasekhara Venkata Raman was an Indian physicist who carried out ground-breaking work in the field of light scattering, which earned him the 1930 Nobel Prize for Physics. In 1954, India honoured him with its highest civilian award, the Bharat Ratna.

Learning Outcomes of the Experiment

At the end of the session, students should be able to :

1)

2)