Regular Expression, Rollover and Frames

Unit - V

Frame

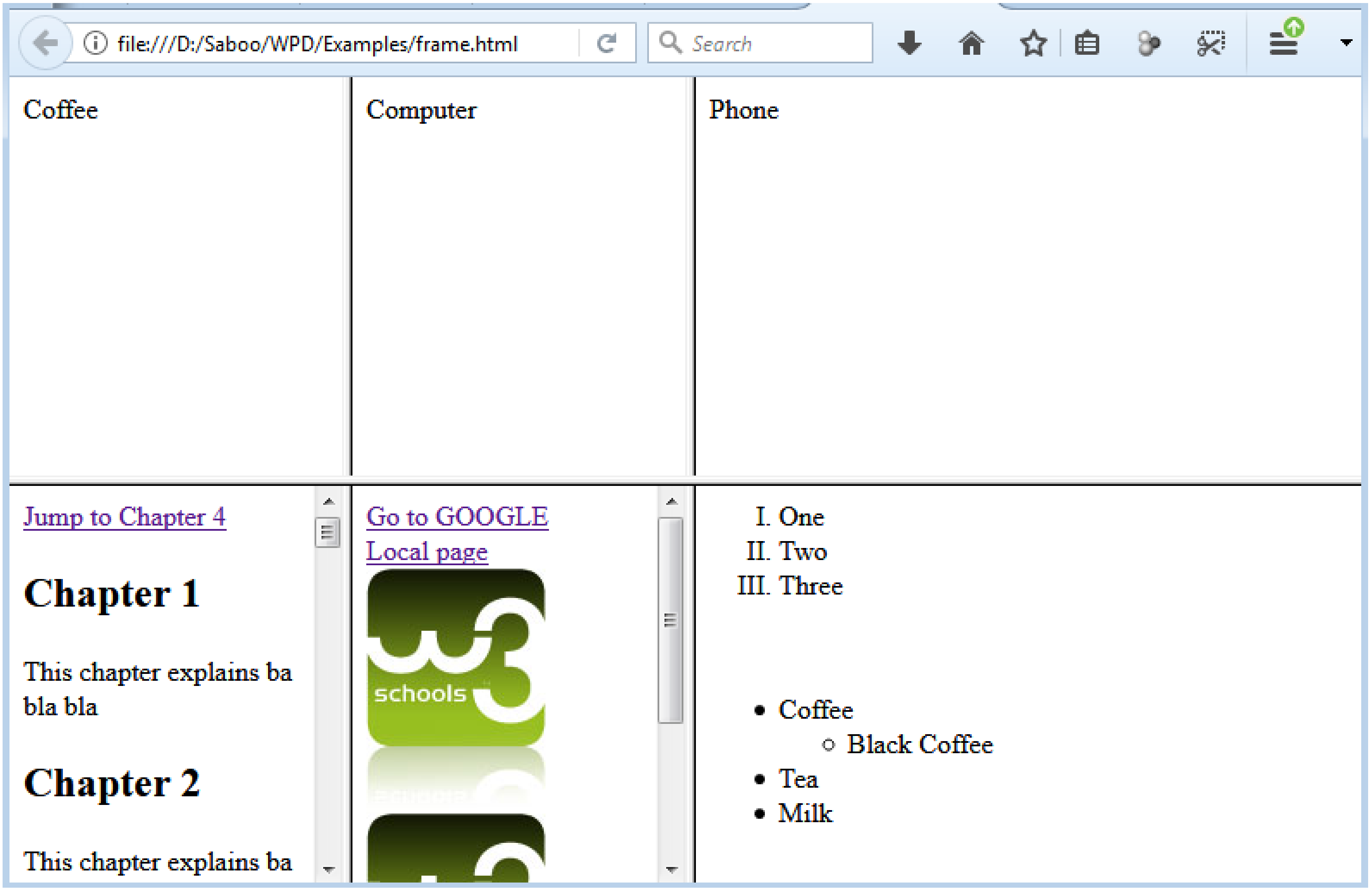
- It defines one particular window (frame) within a <frameset>.
- Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize etc.
- In HTML, <frame> tag has no end tag.

Frame

Attribute	Value	Description
frameborder	0 1	Specifies whether or not to display a border around a frame
longdesc	URL	Specifies a page that contains a long description of the content of a frame
marginheight	pixels	Specifies the top and bottom margins of a frame
Marginwidth	pixels	Specifies the left and right margin of a frame
Name	text	Specifies the name of a frame
noresize	noresize	Specifies that a frame is not resizable
scrolling	yes no auto	Specifies whether or not to display scroll bars in a frame
src	URL	Specifies the URL of the document to show in a frame

Frame

```
<html>

<frameset cols="25%,25%,50%"
           rows="50%,50%">
  <frame src="coffee.html">
  <frame src="computer.html">
  <frame src="phone.html">
  <frame src="Bookmark.html">
  <frame src="anchor.html">
  <frame src="list.html">
</frameset>

</html>
```



Coffee

Computer

Phone

Jump to Chapter 4

Chapter 1

This chapter explains ba bla bla

Chapter 2

This chapter explains ba

Go to GOOGLE
Local page

I. One
II. Two
III. Three

- Coffee
  - Black Coffee
- Tea
- Milk

Frame

- <frameset>:
  - <frameset> element holds one or more <frame> elements.
  - Each <frame> element can hold a separate document.
  - The <frameset > element specifies how many columns or rows there will be in the frameset, and how much percentage/pixels of space will occupy each of them.

Frame

- <frameset>: Attributes

Attribute	Value	Description
cols	pixels%	Specifies the number and size of columns in a frame
rows	pixels%	Specifies the number and size of rows in a frame

Focus to a Child Window

- Focus() method is used to set focus to the new child window.

  Syntax: *win_obj.focus();*

- E.g.

```
var myWindow = window.open("", "", "width=200, height=100");
myWindow.document.write("<p>A new window!</p>");
myWindow.focus();
```

Accessing elements of another child window

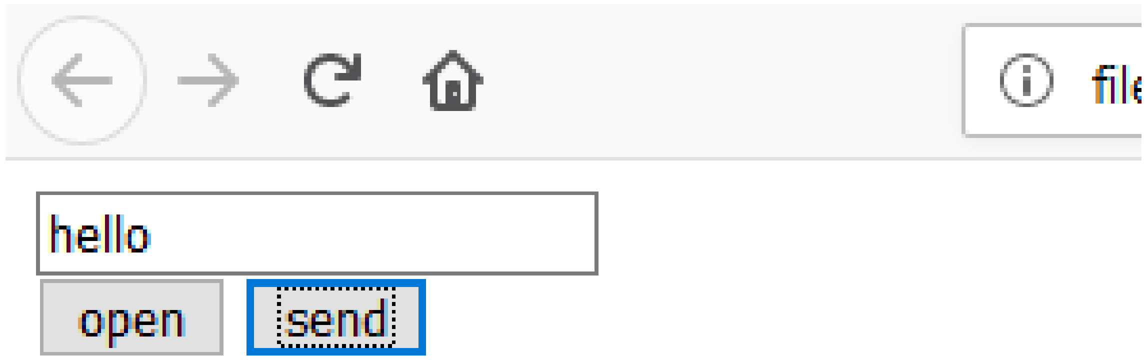
```
<html>
<head>      <title>
Accessing child window elements.
</title>
</head><body>
<input type="text" id="txt1" value=""/><br>
<input type="button" id="btn1" value="open" onclick="openchild()">
<input type="button" id="btn2" value="send" onclick="sendval()">
<script>
var popchild;
function openchild()
{popchild=window.open("child.html","child window");}
function sendval()
{   if(popchild != null   && !popchild.closed)
{var p=popchild.document.getElementById("p1");
p.innerHTML=document.getElementById("txt1").value;
popchild.focus();}
Else{
alert("Child window has been closed.");}
}</script></body></html>
```

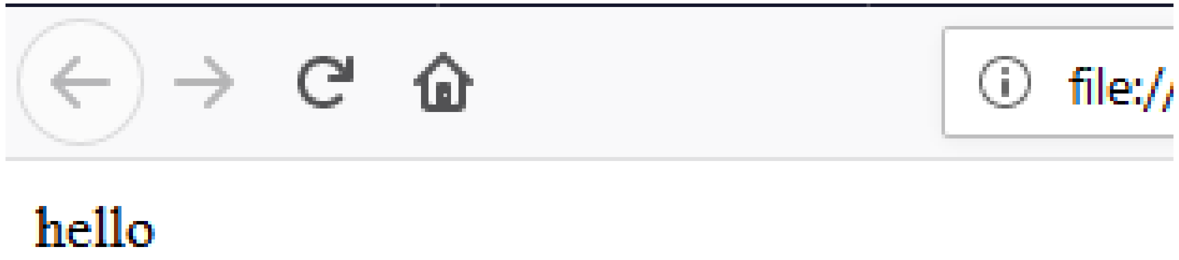
Parent.html

Accessing elements of another child window

```
<html>
    <head>
            <title>
                    Child window
            </title>
    </head>
    <body>
            <p id="p1"></p>
    </body>
</html>
```

Child.html





hello

Rollover

- Rollover is a JavaScript technique used by Web developers to produce an effect in which the appearance of a graphical image changes when the user rolls the mouse pointer over it.
- Rollover also refers to a button on a Web page that allows interactivity between the user and the Web page.
- It causes the button to react by either replacing the source image at the button with another image or redirecting it to a different Web page.

Rollover

- Rollover is triggered when the mouse moves over the primary image, causing the secondary image to appear. The primary image reappears when the mouse is moved away.

- Occasionally, rollover is referred to as synonym for mouseover.

- Rollover can be accomplished using text, buttons or images, which can be made to appear when the mouse is rolled over an image. The user needs two images/buttons to perform rollover action.

Rollover

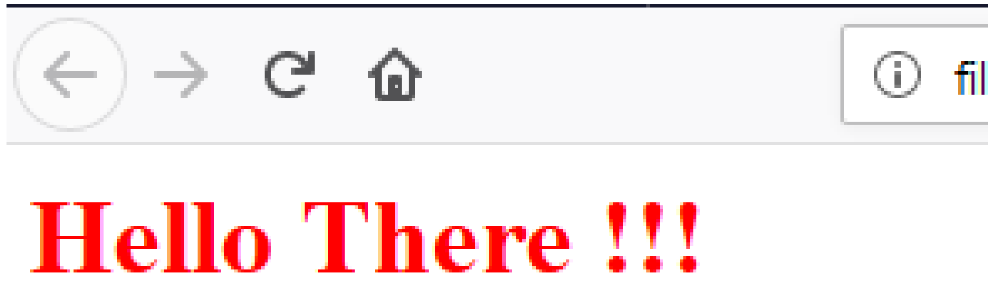
- Syntax:

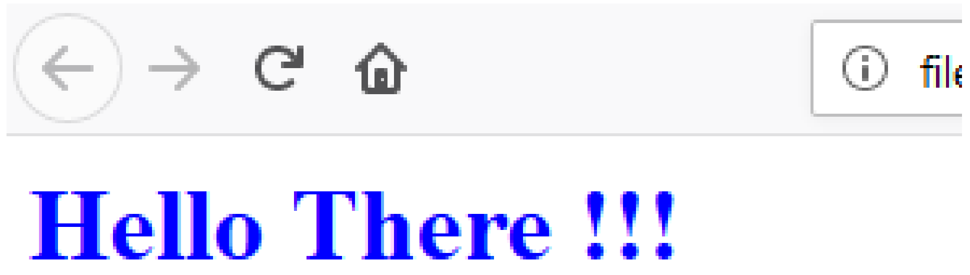
  <element onmouseover="myScript" onmouseout="myScript">

- The *onmousemove* event occurs every time the mouse pointer is moved over the div element.
- The *onmouseenter* event only occurs when the mouse pointer enters the div element.
- The *onmouseover* event occurs when the mouse pointer enters the div element, and its child elements.

Rollover

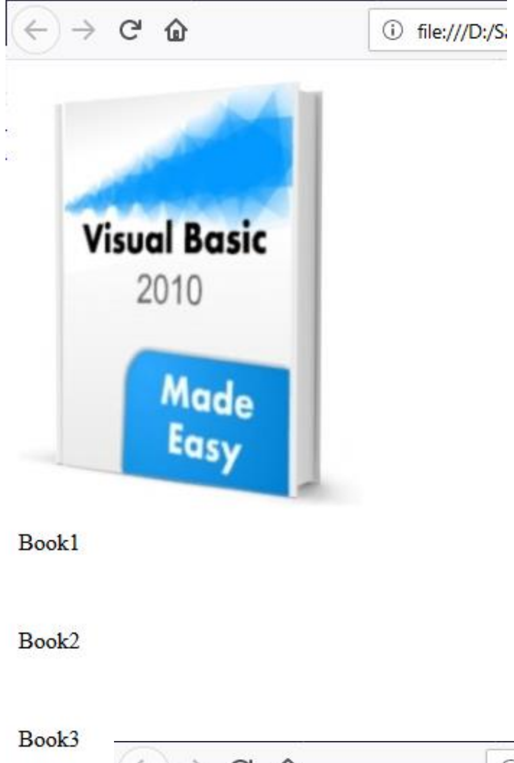
```
<html>
  <head><title>
                Mouse Rollover events
  </title></head>
  <body>
        <h1 id="head1" onmouseover="mouseover()"
        onmouseout="mouseout()">
                Hello There !!!</h1>
        <script>
        function mouseover()
        {document.getElementById("head1").style.color="red";
        }
        function mouseout()
        {document.getElementById("head1").style.color="blue";
        }</script></body></html>
```


Hello There !!!


Hello There !!!

Rollover

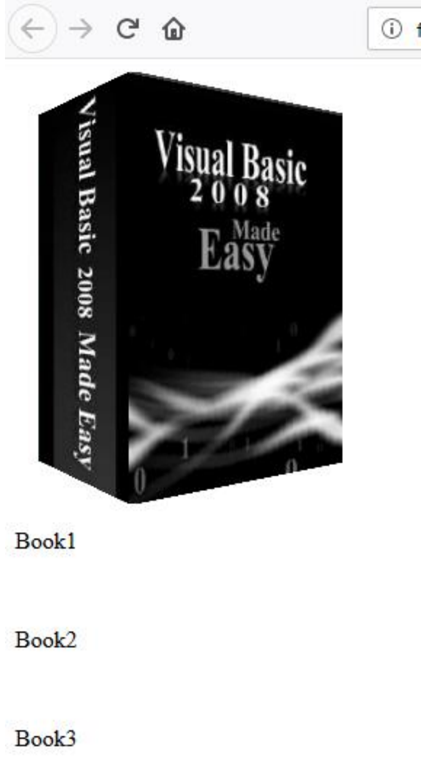
```
<html>
  <head><title>
                         Image rollover using text
  </title></head>
  <body>
        <img src="book1.jpg" name="displaybook"/>
        <p id="p1" onmouseover="showbook(0)">Book1</p><br>
        <p id="p2" onmouseover="showbook(1)">Book2</p><br>
        <p id="p3" onmouseover="showbook(2)">Book3</p><br>
        <script>
        var mybooks=["book1.jpg","book2.jpg","book3.jpg"];
        function showbook(book)
        {document.displaybook.src=mybooks[book];
        }</script>
  </body></html>
```



Book1

Book2

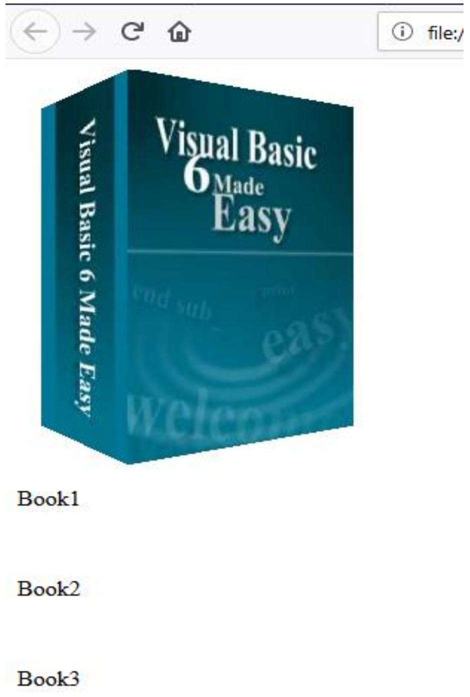
Book3



Book1

Book2

Book3



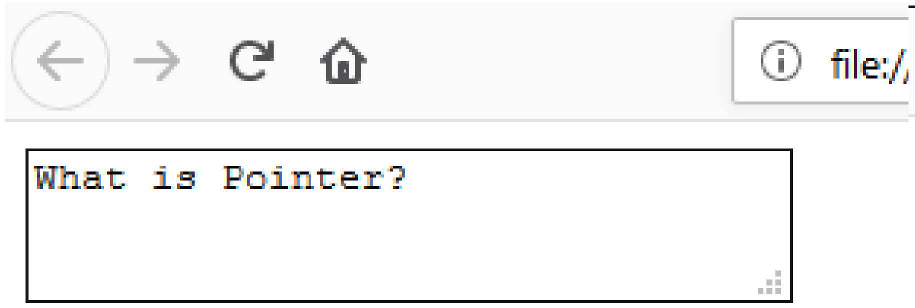
Book1

Book2

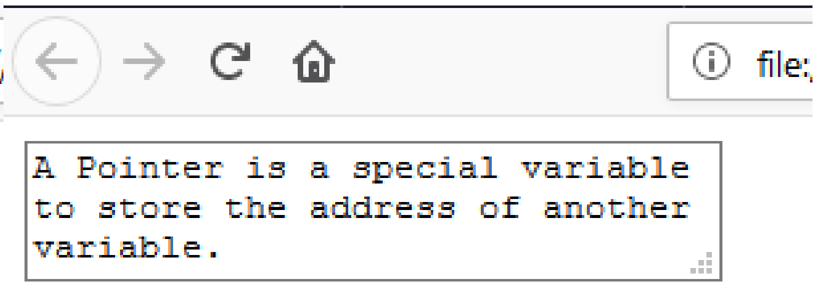
Book3

Rollover

```
<html>
  <head><title>
        Rollover Text
  </title></head>
  <body>
  <textarea rows="2" cols="70" name="rollovertext"
  onmouseover="this.value='What is Pointer?'" onmouseout="this.value='A Pointer
  is a special variable to store the address of another variable.'">
  </textarea>
  </body>
</html>
```



```
What is Pointer?
```



```
A Pointer is a special variable
to store the address of another
variable.
```

Regular Expression

- A regular expression is an object that describes a pattern of characters.

- Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

Syntax:

/pattern/modifiers;

- E.g.: var patt = /w3schools/i

Where:	w3schools/i	is a regular expression
	w3schools	is a pattern to be used in search
	i	is a modifier (modifies search to be case-insensitive)

Regular Expression

- Modifiers:
  - g : performs a global match, case-sensitive(find all matches rather than stopping after the first match)
  - i : performs case-insensitive matching and returns the first occurrence.
  - m: By default, all matching or search operation is done as case sensitive and on single line. To perform search or matching on text containing new line character (\n) use modifier (*m*). Performs multiline matching.

Regular Expression

- Brackets: they are used to find a range of characters.
  - [abc] : find any character between the bracket. i.e. between a, b and c.
  - [^abc]   : find any character NOT between the brackets.
  - [0-9]: Find any character between the brackets (any digit).
  - (x|y): Find any of the alternatives specified.

Regular Expression

- Metacharacters: Characters with special meaning.
  - . : Find any single character, except newline or line terminator.
  - \w : Find a word character.
  - \W: Find a Non word character.
  - \d : Find a digit.
  - \D : Find a Non digit character.
  - \s : Find a white space character.
  - \S : Find a Non white space character.
  - \b : Find a match at the beginning of a word: \bDICE
    or at the end of the word: DICE\b

Regular Expression

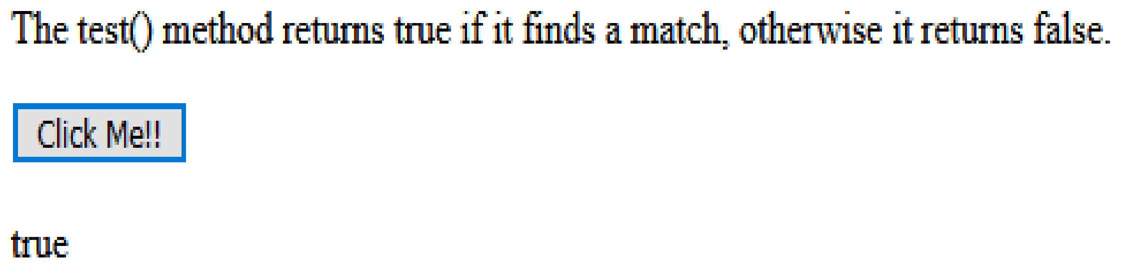
- Quantifiers:
  - n+ : Matches any string that contains at least one n. (1 or more)
  - n* : Matches any string that contains zero or more occurrences of n.
  - n?: Matches any string that contains zero or one occurrences of n.
  - n{X} : Matches any string that contains a sequence of X n's.
  - n{X,Y} : Matches any string that contains a sequence of X to Y n's.
  - n{X} : Matches any string that contains a sequence of at least X n's.
  - n$: Matches any string with n at the end of it.
  - ^n : Matches any string with n at the beginning of it.

Regular Expression

- Methods:
  - exec() : Tests for a match in a string. Returns the first match. (pattern.exec())
  - test() : Tests for a match in a string. Returns True or False. (pattern.test())
  - match() : The match() method searches a string for a match against a regular expression, and returns the matches, as an Array object. (text.match(regexp))

Regular Expression

```
<html>
<body>
<p>The test() method returns true if it finds a match, otherwise it returns false.</p>
<button onclick="myFunction()">Click Me!!</button>
<p id="demo"></p>
<script>
function myFunction() {
  var str = "There is a price for everything, nothing is free in this world";
  var patt = new RegExp("nothing");
  var res = patt.test(str);
  document.getElementById("demo").innerHTML = res;
}
</script>
</body>
</html>
```

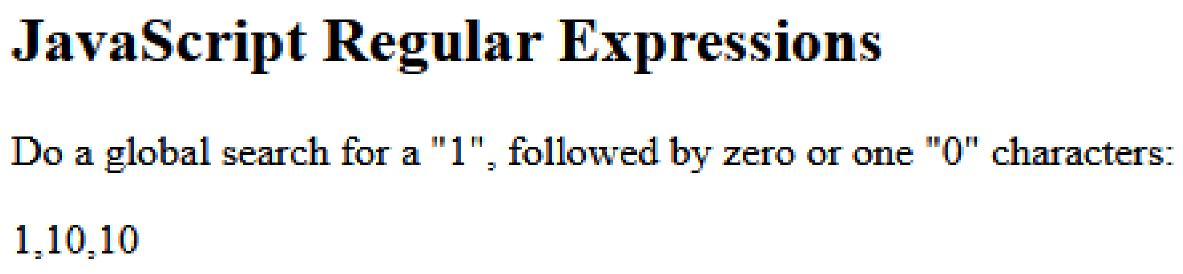


```
The test() method returns true if it finds a match, otherwise it returns false.
```

```
true
```

Regular Expression

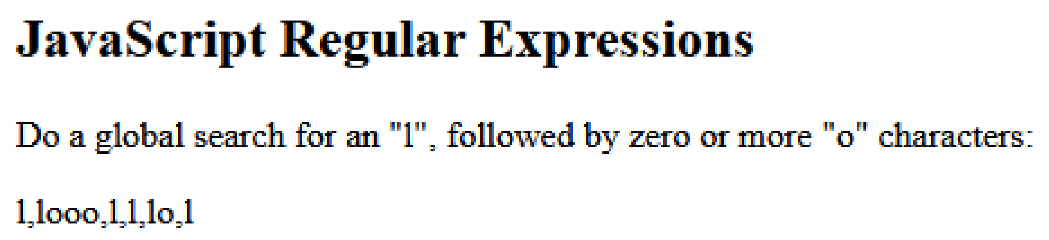
```
<body>
<h2>JavaScript Regular Expressions</h2>
<p>Do a global search for a "1", followed by zero or one "0" characters:</p>
<p id="demo"></p>
<script>
let text = "1, 100 or 1000?";
let result = text.match(/10?/g);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```



```
JavaScript Regular Expressions

Do a global search for a "1", followed by zero or one "0" characters:

1,10,10
```

Regular Expression

```
<html>
<body>

<h2>JavaScript Regular Expressions</h2>

<p>Do a global search for an "l", followed by zero or more "o" characters:</p>
<p id="demo"></p>
<script>
let text = "Hellooo World! Hello W3Schools!";
let result = text.match(/lo*/g);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```



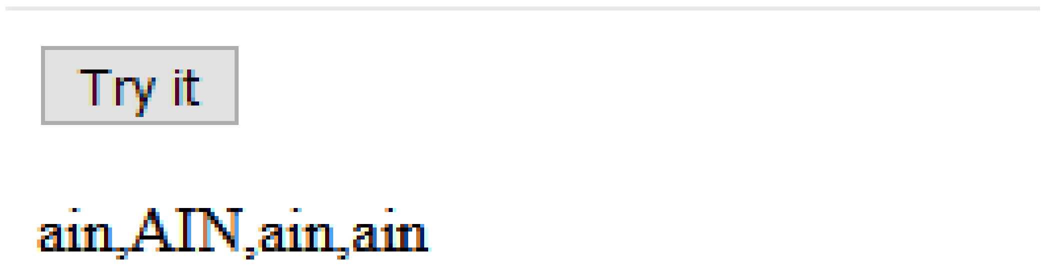
JavaScript Regular Expressions

Do a global search for an "l", followed by zero or more "o" characters:

```
l,looo,l,l,lo,l
```

Regular Expression

```
<html>
<body>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
  var str = "The rain in SPAIN stays mainly in the plain";
  var res = str.match(/ain/gi);
  document.getElementById("demo").innerHTML = res;
}
</script>
</body>
</html>
```



```
ain,AIN,ain,ain
```