

---

**AIM:** Write *JavaScript* to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern  
name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

**DESCRIPTION:**

JavaScript is a simple scripting language invented specifically for use in web browsers to make websites more dynamic. On its own, HTML is capable of outputting more-or-less static pages. Once you load them up your view doesn't change much until you click a link to go to a new page. Adding JavaScript to your code allows you to change how the document looks completely, from changing text, to changing colors, to changing the options available in a drop-down list. JavaScript is a *client-side* language.

JavaScripts are integrated into the browsing environment, which means they can get information about the browser and HTML page, and modify this information, thus changing how things are presented on your screen. This access to information gives JavaScript great power to modify the browsing experience. They can also react to *events*, such as when the user clicks their mouse, or points to a certain page element. This is also a very powerful ability.

**Regular Expressions:**

One of the most common situations that come up is having an HTML form for users to enter data. Normally, we might be interested in the visitor's name, phone number and email address, and so forth. However, even if we are very careful about putting some hints next to each required field, some visitors are going to get it wrong, either accidentally or for malicious purposes. Here's where regular expressions come in handy. A regular expression is a way of describing a pattern in a piece of text. In fact, it's an easy way of matching a string to a pattern. We could write a simple regular expression and use it to check, quickly, whether or not any given string is a properly formatted user input. This saves us from difficulties and allows us to write clean and tight code.

A regular expression is a JavaScript object. There are multiple ways of creating them. They can be created statically when the script is first parsed or dynamically at run time. A static regular expression is created as follows:

```
regx=/fish|fow1/;
```

Dynamic patterns are created using the keyword to create an instance of the RegExp class:

```
regx=new RegExp("fish|fow1");
```

**Functions:**

**test(string)-** Tests a string for pattern matches. This method returns a Boolean that indicates whether or not the specified pattern exists within the searched string. This is the most commonly used method for validation. It updates some of the properties of the parent RegExp object following a successful search.

**exec(string)**- Executes a search for a pattern within a string. If the pattern is not found, exec() returns a null value. If it finds one or more matches it returns an array of the match results. It also updates some of the properties of the parent RegExp object

#### PROGRAM:

##### Valid.js

```
function fun()
{

var userv=document.forms[0].user.value;
var pwdv=document.forms[0].pwd.value;
var emailv=document.forms[0].email.value;
var phv=document.forms[0].ph.value;
    var userreg=new RegExp("^[a-zA-Z][a-zA-Z0-9]*$");
    var emailreg=new RegExp("^[a-zA-Z][a-zA-Z0-9_]*@[a-zA-Z][a-zA-Z0-9_]*.[a-zA-Z][a-zA-Z0-9_]{2}.[a-zA-Z][a-zA-Z0-9_]{2}$|^[a-zA-Z][a-zA-Z0-9_]*@[a-zA-Z][a-zA-Z0-9_]*.[a-zA-Z][a-zA-Z0-9_]{3}$");
    var phreg=new RegExp("^[0-9]{10}$");
var ruser=userreg.exec(userv);
var remail=emailreg.exec(emailv);
var rph=phreg.exec(phv);
if(ruser && remail && rph && (pwdv.length > 6))
    {
        alert("All values are valid");
        return true;
    }
    else
    {
        if(!ruser) { alert("username invalid");document.forms[0].user.focus();}
        if(!remail) { alert("password invalid");document.forms[0].user.focus();}
        if(!rph) { alert("phone number invalid");document.forms[0].ph.focus();}
        if(pwdv.length < 6) { alert("password invalid");document.forms[0].pwd.focus();}
        return false;
    }

}
```

##### Register.html

```
<html>
<body>
<center>
<fieldset>
<legend>Registration</legend>
<form action="Database" method="get" onSubmit="return fun()">
<pre>
Name      :<input type="text" name="user" size="10"><br>
Password  :<input type="password" name="pwd" size="10"><br>
E-mail    :<input type="text" name="email" size="10"><br>
Phone Number :<input type="text" name="ph" size="10"><br>
<input type="submit" value="Register">
</pre>
```

```
</form>  
</body>  
<script src="valid.js"></script>  
</html>
```