# Web Programming

Javascript

Module 2

# Agenda

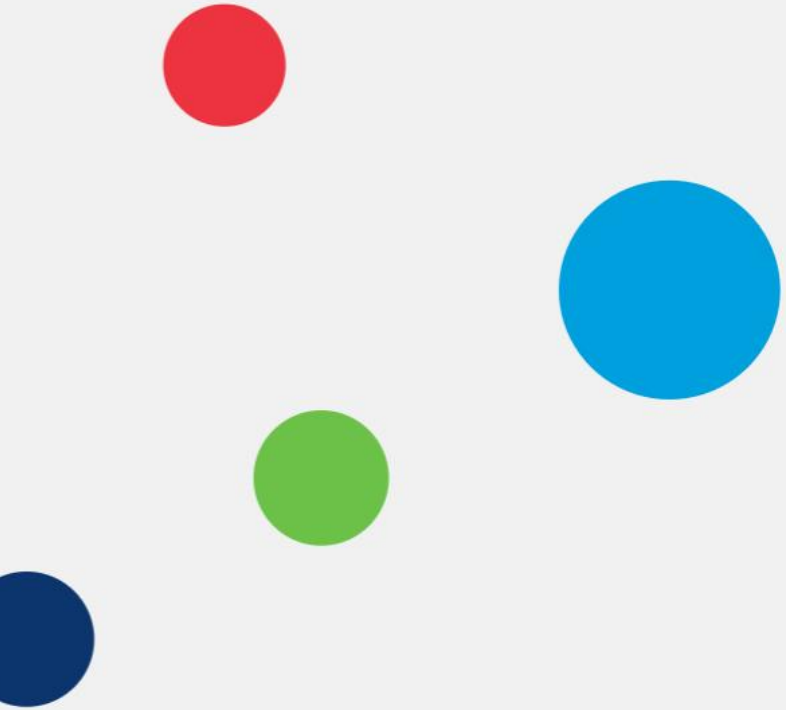
1. JavaScript Regular Expressions

# Objectives

At the end of this module you will be able to:

- Understand the use of regular expressions
- Validate user inputs using regular expressions

# JavaScript Regular Expressions



# Regular Expression in Javascript

- The most difficult part of creating user interface for a web application is user validation
- Developing user interfaces that will be accessed by different browsers is much more painful, due to lack of useful validation functions in Javascript
- Luckily, Javascript (version 1.2 and above) has incorporated regular expressions, using which we can perform validations easily
- Regular expression are tools for performing pattern matching
- We can perform complex task that requires lengthy procedures with just few lines using regular expressions

# Use of Patterns

- Regular expressions are implemented in Javascript in the following way:
- var regexp = /pattern/
- To use regular expressions to validate a String you need to define a pattern String that defines the search criteria
- Use a relevant String method to denote actions like search or test
- Patterns are defined using String literal characters or meta characters

# Metacharacters

- Metacharacters are characters with special meaning

| Metacharacter | What it means |
|---|---|
| \d | Find a digit |
| \D | Find a non-digit character |
| \w | Find a word character |
| \W | Find a non-word character |
| \s | Find a whitespace character |
| \S | Find a non-whitespace character |
| \b | Find a match at the beginning or end of a word |
| \B | Find a match not at the beginning or end of a word |

# Example 1

- The following example demonstrates how to use patterns that will check whether the key pressed is a digit or not :

```
<html>
<body>
<script language="javascript">
function onlyNumbers(e){
  var keynum
  var keychar
  var numcheck
  if(window.event) {
    keynum = e.keyCode
  }
```

Contd..

# Example 1 (Contd.).

```
  keychar = String.fromCharCode(keynum)
  numcheck = /\d/
  return numcheck.test(keychar)
}
</script>

<form>
<input type="text" onkeypress="return onlyNumbers(event)" />
</form>
</body>
</html>
```

# Example 1 (Contd.).

- When the user presses any key, an event is generated. The key on which the event is generated, is stored in the variable keynum by capturing the keycode.
- The javascript String method ***fromCharCode()*** is used to convert the unicode value to character and it is stored in the variable keychar.
- The variable ***numcheck*** defines a pattern for searching. Here \d is the metacharacter, which is used to find a digit.
- The String method ***test()*** is used to match the pattern(here it is trying to match the character obtained from the keypress with \d, i.e a digit).
- Thus, the function ***onlyNumbers()*** will return the value of the keypress only if it is a digit(0 – 9). It will not return any other character.

# Brackets

- Brackets are used to find a range of characters

| Expression | What it means |
|---|---|
| [xyz] | Find any character between the brackets |
| [^xyz] | Find any character not between the brackets |
| [A-Z] | Find any character from uppercase A to uppercase Z |
| [a-z] | Find any character from lowercase a to lowercase z |
| [A-z] | Find any character from uppercase A to lowercase z |
| [0-9] | Find any digit from 0 to 9 |
| [Bandra\|Andheri\|Borivli] | Find any of the alternatives specified |

# Example 2

- The following example demonstrates how to use patterns that will check whether the key pressed is an alphabet(Lower case or upper case):

```
<html>
<body>
<script language="javascript">
function onlyCharacters(e){
  var keynum
  var keychar
  var charcheck
  if(window.event) {
    keynum = e.keyCode
  }
```



Contd..

# Example 2 (Contd.).

```
  keychar = String.fromCharCode(keynum)
  charcheck = /[A-Za-z]/
  return charcheck.test(keychar)
}
</script>

<form>
<input type="text" onkeypress="return onlyCharacters(event)" />
</form>
</body>
</html>
```

# Quiz

What happens when you move the mouse pointer on the image displayed?

```
<html>
 <body bgcolor="pink">
<img name=ash src="ash2.bmp" onmousemove=vh() onmouseout=hv()>
<script>
function vh() {
  document.ash.src="ash3.bmp";
}
function hv(){
  document.ash.src="ash2.bmp";
}
</script>
</body></html>
```

# Quiz (Contd.).

What happens when you move the mouse pointer on the image displayed? What happens when you move the mouse pointer out of the image?

```
<html>
<img name=ash src="ash2.bmp" onmousemove=vh() onmouseout=hv()>
<script>
function vh(){
  document.ash.height=document.ash.height+1;
  document.ash.width=document.ash.width+1;
}
function hv(){
  document.ash.height=document.ash.height-100;
  document.ash.width=document.ash.width-100;
}
</script></body></html>
```

# Summary

In this module, you were able to:

- Understand the use of regular expressions
- Validate user inputs using regular expressions



# Thank You