JAVASCRIPT REGULAR EXPRESSIONS

Presented by
Eng./Abanoub Nabil
Teaching assistant at ITI

JavaScript Regular Expressions

- 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.
- A Regular Expression lets you **build patterns** using a set of special characters.

- **For example, *Form Validation*** is one of the most common requirements. You don't know what exact values the user will enter, **but** you do know the format they need to use.

JavaScript Regular Expressions

- **RegExp Syntax** Using literal RegExp:

```
/pattern/modifiers;
```

- **RegExp Syntax Using** Explicitly using the RegExp object:

```
new RegExp("pattern" "modifiers");
```

JavaScript Regular Expressions

- **Example** Using literal RegExp:

```
var myRegExp = /j.*t/i;
```

- **Example** Explicitly using the RegExp object:

```
var searchPattern = new RegExp("j.*t", "i");
```

JavaScript Regular Expressions

- **In the example above:**
- **j.*t** is the regular expression pattern. **It means**, "Match any string that starts with j, ends with t and has zero or more characters in between".
- **The asterisk** * means "zero or more of the preceding“.
- **The dot (.)** means "any character“.

JavaScript Regular Expressions

Regular Expression Object Methods:

1-test()

-returns a boolean (true when there's a match, false otherwise)

```
var reg=/i.*I/ ;
var t= reg.test("iTI")
console.log(t)
```

JavaScript Regular Expressions

Regular Expression Object Methods:

2-exec()

-returns first matched strings.

```
var reg=/j.*t/i;
var str="Jscrip is the same of javascript";
var res= reg.exec(str);
console.log(res)
```

JavaScript Regular Expressions

Regular Expression Object Methods:

3-match()

- returns an array of matches.

```
var str = "Is this is it?";
 var patt1 = /is/g;
 var result = str.match(patt1);
console.log(result)
```

JavaScript Regular Expressions

Regular Expression Object Methods:

4-search()

- returns the position of the first match.

```
var str = "Is this is it?";
var patt1 = /is/g;
var result = str.search(patt1);
console.log(result) //5
```

JavaScript Regular Expressions

Regular Expression Modifiers

i	Perform case-insensitive matching

Example

```
var str = "Visit W3Schools";
var patt1 = /w3schools/i;
var result = str.match(patt1);
console.log(result);
```

JavaScript Regular Expressions

Regular Expression Modifiers

g	Perform a global match (find all matches rather than stopping after the first match)

Example

```
var str = "Is this all there is?";
var patt1 = /is/g;
var result = str.match(patt1);
console.log(result);
```

JavaScript Regular Expressions

Regular Expression Modifiers

m	Perform multiline matching

Example

```
var str = "\nIs th\nis it?";
var patt1 = /^is/m;
var result = str.match(patt1);
console.log(result)
```

JavaScript Regular Expressions

- Regular Expression Patterns

[abc]	Find any of the characters between the brackets

Example

```
var str = "Is this all there is?";
var patt1 = /[h]/g;
var result = str.match(patt1);
console.log(result)
```

JavaScript Regular Expressions

- Regular Expression Patterns

[0-9]	Find any of the digits between the brackets

Example

```
var str = "123456789";
 var patt1 = /[1-4]/g;
 var result = str.match(patt1);
 console.log(result)
```

JavaScript Regular Expressions

- Regular Expression Patterns

(x|y)	Find any of the alternatives separated with |

Example

```
var str = "re, green, red, green, gren, gr, blue, yellow";
 var patt1 = /(red|green)/g;
 var result = str.match(patt1);
 console.log(result)
```

JavaScript Regular Expressions

- **Metacharacters** are characters with a special meaning:

\d	Find a digit

Example

```
var str = "Give 100%!";
 var patt1 = /\d/g;
 var result = str.match(patt1);
 console.log(result)
```

JavaScript Regular Expressions

- **Metacharacters** are characters with a special meaning:

\s	Find a whitespace character

Example

```
var str = "Is this all there is?";
  var patt1 = /\s/g;
  var result = str.match(patt1);
  console.log(result)
```

JavaScript Regular Expressions

- **Metacharacters** are characters with a special meaning:

\b	Find a match at the beginning of a word like this: \bWORD, or at the end of a word like this: WORD\b

Example

```
var str = "HELLO, LOOK AT YOU!";
var patt1 = /\bLO/;
var result = str.search(patt1);
console.log(result)
```

```
var str = "HELLO, LOOK AT YOU!";
var patt1 = /LO\b/;
var result = str.search(patt1);
console.log(result)
```

JavaScript Regular Expressions

- **Quantifiers** define quantities:

n+	Matches any string that contains at least one *n*

Example

```
var str = "Hellooo World! Hello W3Schools!";
var patt1 = /o+/g;
var result = str.match(patt1);
console.log(result) //ooo,o,o,oo
```

JavaScript Regular Expressions

- **Quantifiers** define quantities:

n*	Matches any string that contains zero or more occurrences of *n*

Example

```
//do a global search for an "l", followed by zero or more "o" characters
 var str = "Hellooo World! Hello W3Schools!";
  var patt1 = /lo*/g;
  var result = str.match(patt1)
console.log(result) // l,looo,l,l,lo,l
```

JavaScript Regular Expressions

- **Quantifiers** define quantities:

n?	Matches any string that contains zero or one occurrences of *n*

Example

```
// to do a global search for a "1", followed by zero or one "0" characters.
 var str = "1, 100 or 1000?";
  var patt1 = /10?/g;
  var result = str.match(patt1);)
console.log(result) // 1,10,10
```

JavaScript Regular Expressions

Character	Description	Example
.	Any character	/a.*a/ matches "aa", "aba", "a9qa", "a!?_a",
^	Start	/^a/ matches “apple”, “abcde”..
$	End	/z$/ matches “abcz”, “az”..
|	Or	/abc|def|g/ matches lines with "abc", "def", or "g"
[ ]	Match any one character between the brackets	/[a-z]/ matches any lowercase letter
[^]	Match any one character not between the brackets	/[^abcd]/ matches any character but not a, b, c, or d

JavaScript Regular Expressions

-

^The	matches any string that **starts with** The -> Try it!
end$	matches a string that **ends with** end
^The end$	**exact string match** (starts and ends with The end)
roar	matches any string that **has the text** roar **in it**

JavaScript Regular Expressions


```
abc*          matches a string that has ab followed by zero or more c -
> Try it!

abc+          matches a string that has ab followed by one or more c

abc?          matches a string that has ab followed by zero or one c

abc{2}        matches a string that has ab followed by 2 c

abc{2,}       matches a string that has ab followed by 2 or more c

abc{2,5}      matches a string that has ab followed by 2 up to 5 c

a(bc)*        matches a string that has a followed by zero or more
copies of the sequence bc

a(bc){2,5}    matches a string that has a followed by 2 up to 5 copies
of the sequence bc
```

JavaScript Regular Expressions

- a(b|c) matches a string that has a followed by b or c *(and captures b or c)* -> Try it!

  a[bc] same as previous, *but without capturing b or c*

[abc] matches a string that has either an a or a b or a c -> is the same as a|b|c -> Try it!

[a-c] same as previous

[a-fA-F0-9] a string that represents a single hexadecimal digit, case insensitively -> Try it!

[0-9]% a string that has a character from 0 to 9 before a % sign

[^a-zA-Z] a string that has not a letter from a to z or from A to Z. In this case the ^ is used as negation of the expression -> Try it!

JavaScript Regular Expressions

- a(b|c) matches a string that has a followed by b or c *(and captures b or c)* -> Try it!

  a[bc] same as previous, *but without capturing b or c*

[abc] matches a string that has either an a or a b or a c -> is the same as a|b|c -> Try it!

[a-c] same as previous

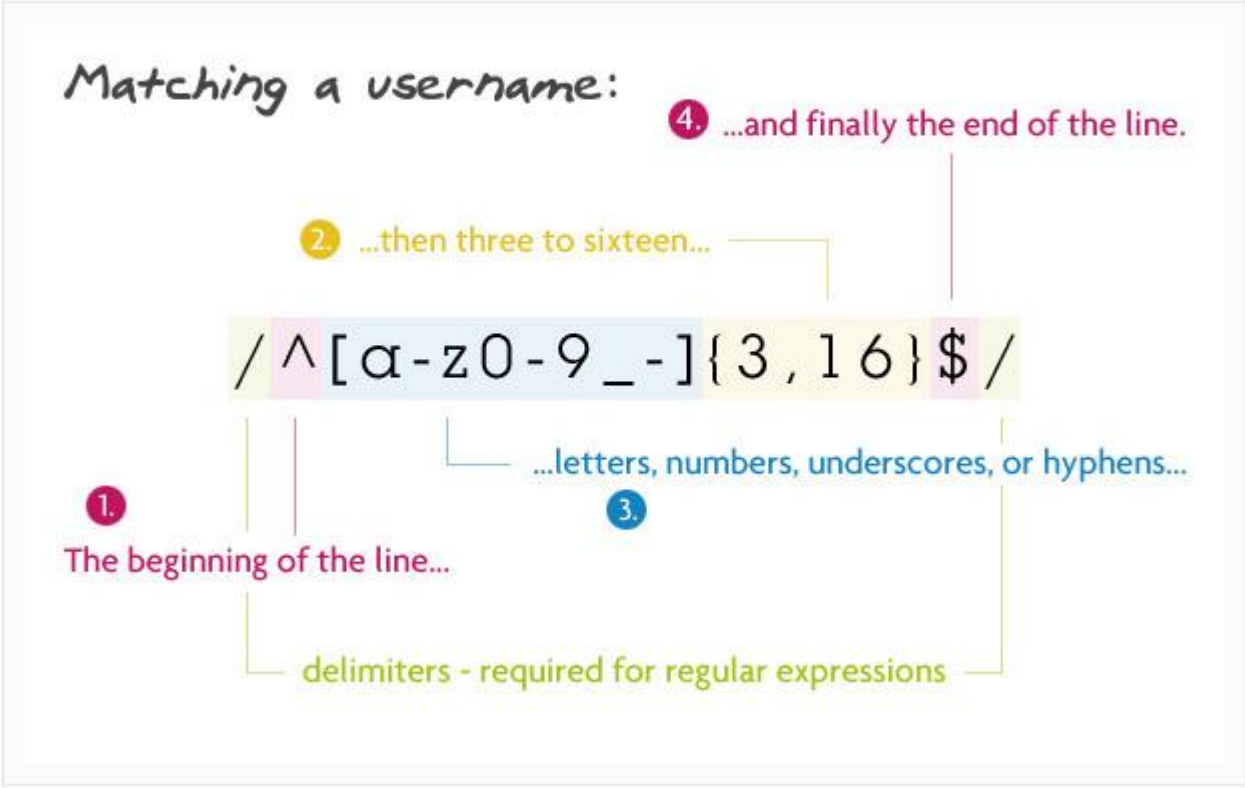
[a-fA-F0-9] a string that represents a single hexadecimal digit, case insensitively -> Try it!

[0-9]% a string that has a character from 0 to 9 before a % sign

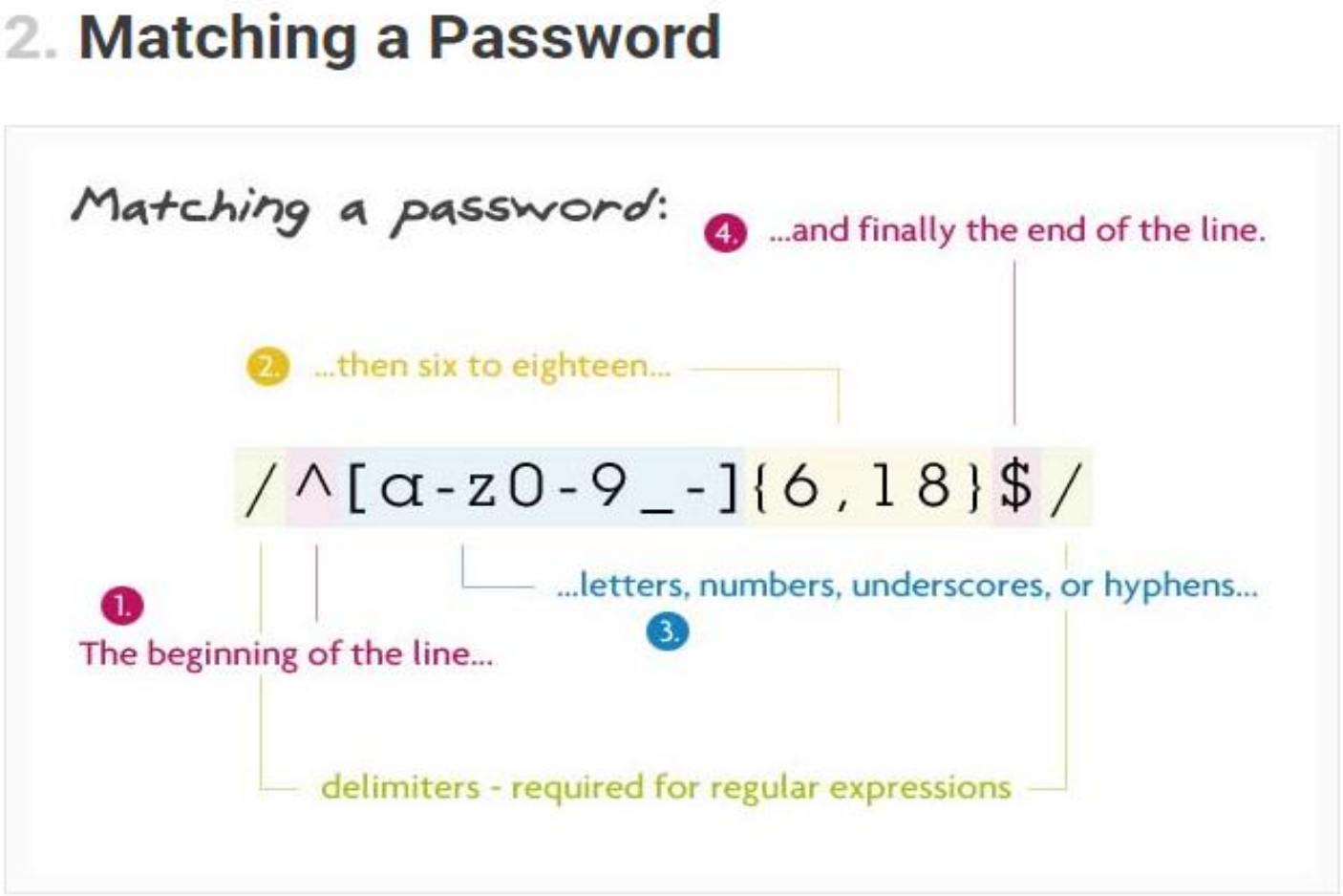
[^a-zA-Z] a string that has not a letter from a to z or from A to Z. In this case the ^ is used as negation of the expression -> Try it!

JavaScript Regular Expressions

- 1. Matching a Username



JavaScript Regular Expressions



JavaScript Regular Expressions

- ## 5. Matching an Email