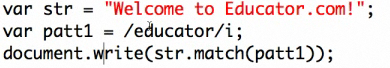
**REGULAR EXPRESSIONS**

1. var reExample = /pattern/;

2. var reExample = new RegExp("pattern");



Modifier i , g

\$\$\$      // Matches "Show me the $$$!"

[12345]     // Matches "1" and "3", but not "a" or "12"

1-5]       // 1,2,3,4, or 5

[0-9a-zA-Z] // Matches any letter or digit

[^a-zA-Z]   // Matches anything except a letter

bana?na     // Matches "banana" and "banna", // but not "banaana"

ba(na)+na   // Matches "banana" and "banananana", // but not "bana" or "banaana".

^(ba|na)+$  // Matches "banana", "nababa", "baba"

var phoneRE = /^\(\d{3}\) \d{3}-\d{4}$/;

var digit5=/^\d{5}$/ //regular expression defining a 5 digit number

if (document.myform.myinput.value.search(digit5)==-1) //if match failed

alert("Please enter a valid 5 digit number inside form")

^ begin string

\d{5} 5 digits

$ end string

**var RE\_SSN = /^[0-9]{3}[\- ]?[0-9]{2}[\- ]?[0-9]{4}$/;**

**function checkSsn(ssn){**

**if (RE\_SSN.test(ssn)) {**

**alert("VALID SSN");**

**} else {**

**alert("INVALID SSN");**

**}**

**}**

^\d{3}([\- ]?)\d{2}([\- ]?)\d{4}$

* 123-45-6789
* 123 45 6789
* 123456789
* 123-45 6789
* 123 45-6789
* 123-456789

var RE\_EMAIL = /^(\w+\.)\*\w+@(\w+\.)+[A-Za-z]+$/;

1. The caret (^) says to start at the beginning. This prevents the user from entering invalid characters at the beginning of the email address.
2. (\w+[\-\.])\* allows for a sequence of word characters followed by a dot or a dash. The \* indicates that the pattern can be repeated zero or more times. Successful patterns include "ndunn.", "ndunn-", "nat.s.", and "nat-s-".
3. \w+ allows for one or more word characters.
4. @ allows for a single @ symbol.
5. (\w+\.)+ allows for a sequence of word characters followed by a dot. The + indicates that the pattern can be repeated one or more times. This is the domain name without the last portion (e.g, without the "com" or "gov").
6. [A-Za-z]+ allows for one or more letters. This is the "com" or "gov" portion of the email address.
7. The dollar sign ($) says to end here. This prevents the user from entering invalid characters at the end of the email address.

var RE\_PASSWORD = /^[A-Za-z\d]{6,8}$/;

1. The caret (^) says to start at the beginning. This prevents the user from entering invalid characters at the beginning of the password.
2. [A-Za-z\d]{6,8} allows for a six- to eight-character sequence of letters and digits.
3. The dollar sign ($) says to end here. This prevents the user from entering invalid characters at the end of the password.