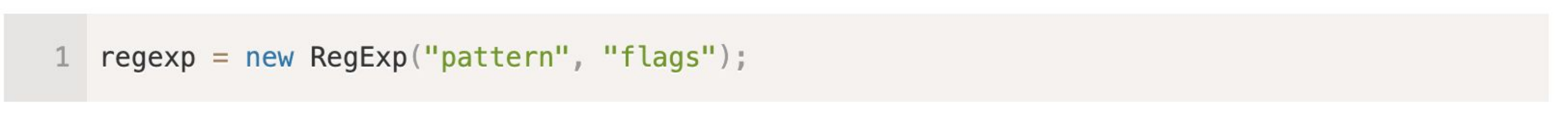
Advanced JavaScript

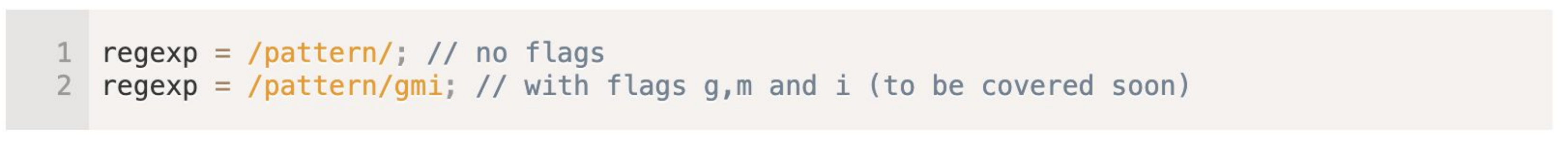
•••

Lecture 3

Regular expressions - Definition



```
regexp = new RegExp("pattern", "flags");
```



```
regexp = /pattern/; // no flags
regexp = /pattern/gmi; // with flags g,m and i (to be covered soon)
```

Definition (continue)

- A regular expression consists of a pattern and optional flags: `g, i, m, u, s, y`.
- The method `str.match(regexp)` looks for matches: all of them if there’s `g` flag, otherwise, only the first one.
- The method `str.replace(regexp, replacement)` replaces matches found using regexp with replacement: all of them if there’s `g` flag, otherwise only the first one. The method `regexp.test(str)` returns `true` if there’s at least one match, otherwise, it returns `false`.

Regular Expression - Flags

i With this flag the search is case-insensitive: no difference between A and a (see the example below).

g With this flag the search looks for all matches, without it – only the first match is returned.

m Multiline mode (covered in the chapter Multiline mode of anchors ^ $, flag "m").

s Enables “dotall” mode, that allows a dot . to match newline character \n (covered in the chapter Character classes).

u Enables full unicode support. The flag enables correct processing of surrogate pairs. More about that in the chapter Unicode: flag "u" and class \p{...}.

y “Sticky” mode: searching at the exact position in the text (covered in the chapter Sticky flag "y", searching at position)

Character classes

`\d` A digit: a character from 0 to 9

`\w` A “worldly” character: either a letter of Latin alphabet or a digit or an underscore _. Non-Latin letters (like cyrillic or hindi) do not belong to `\w`.

`\s` A space symbol: includes spaces, tabs \t, newlines \n and few other rare characters, such as \v, \f and \r.

Character Inverse classes

`\D`

Non-digit: any character except \d, for instance a letter.

`\W`

Non-wordly character: anything but \w, e.g a non-latin letter or a space.

`\S`

Non-space: any character except \s, for instance a letter.

Anchors

```
^
```

String starts with a specific pattern

```
$
```

String ends with specific pattern

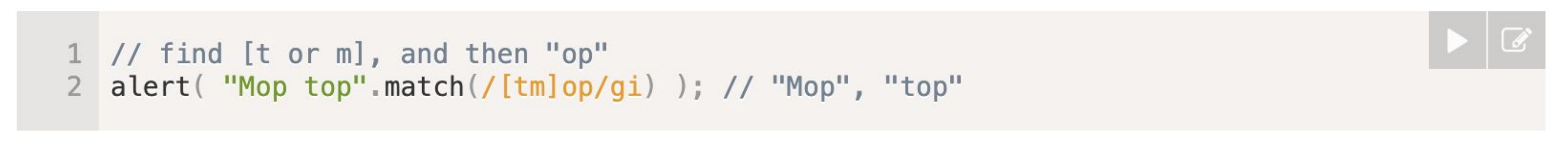
```
^......$
```

Matching full string from start to end with specific pattern

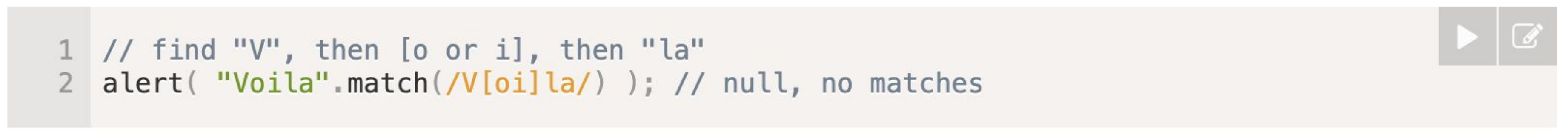
Sets

[abc]

Several characters or character classes inside square brackets […] mean to “search for any character among given”.



```
// find [t or m], and then "op"
alert( "Mop top".match(/[tm]op/gi) ); // "Mop", "top"
```



```
// find "V", then [o or i], then "la"
alert( "Voila".match(/V[oi]la/) ); // null, no matches
```

Ranges

```
[a-w]
```

Using sets brackets to create a range of characters, the range above searches from “a” until character “w”

```
[a-z0-9]
```

Range searches for two ranges any character included in any of these two ranges would be accepted

Quantifiers

Quantity {n}

\d{5} Exact size of numbers should be “5” not accepted less or more

\d{1,4} Range size accepted between “1” until “4” not accepted below the minimum or above the maximum

\d{2,} Range size with at least “2” and no limit of maximum size

Quantifiers (continue)

+ ⇒ {1,}

? ⇒ {0,1}

* ⇒ {0,}

Lab

- Create a signup form that have the following fields: Email address and Password , user name.
- User Name must accept only letters (numbers or special characters are not allowed).
- Email field must contains @ character and end with . and domain name . ex: test@gamil.com .
- Password field must contains at least One capital letter , one small letter and one number.
  Special characters are allowed.
  Password length must be at least 8 characters.
  If the Password length less than 8 characters show message “your password is very weak” , otherwise is accepted.

Lab

- Signup button must be dimmed if we the user has any validation errors , otherwise submit the form successfully.

- Bounce:
    - Store the user data on your code.
    - If the user try to sign up again with the same email address , show message “this email is already registered you can login easly” and show a login button.
    - When user clicked the login button , you must hide the signup form and show another form that contains 2 fields only (email and password) with email validation only.
    - If the user enter his data and matches any record that you stored before , alert logged in successfully otherwise alert email or password is wrong“