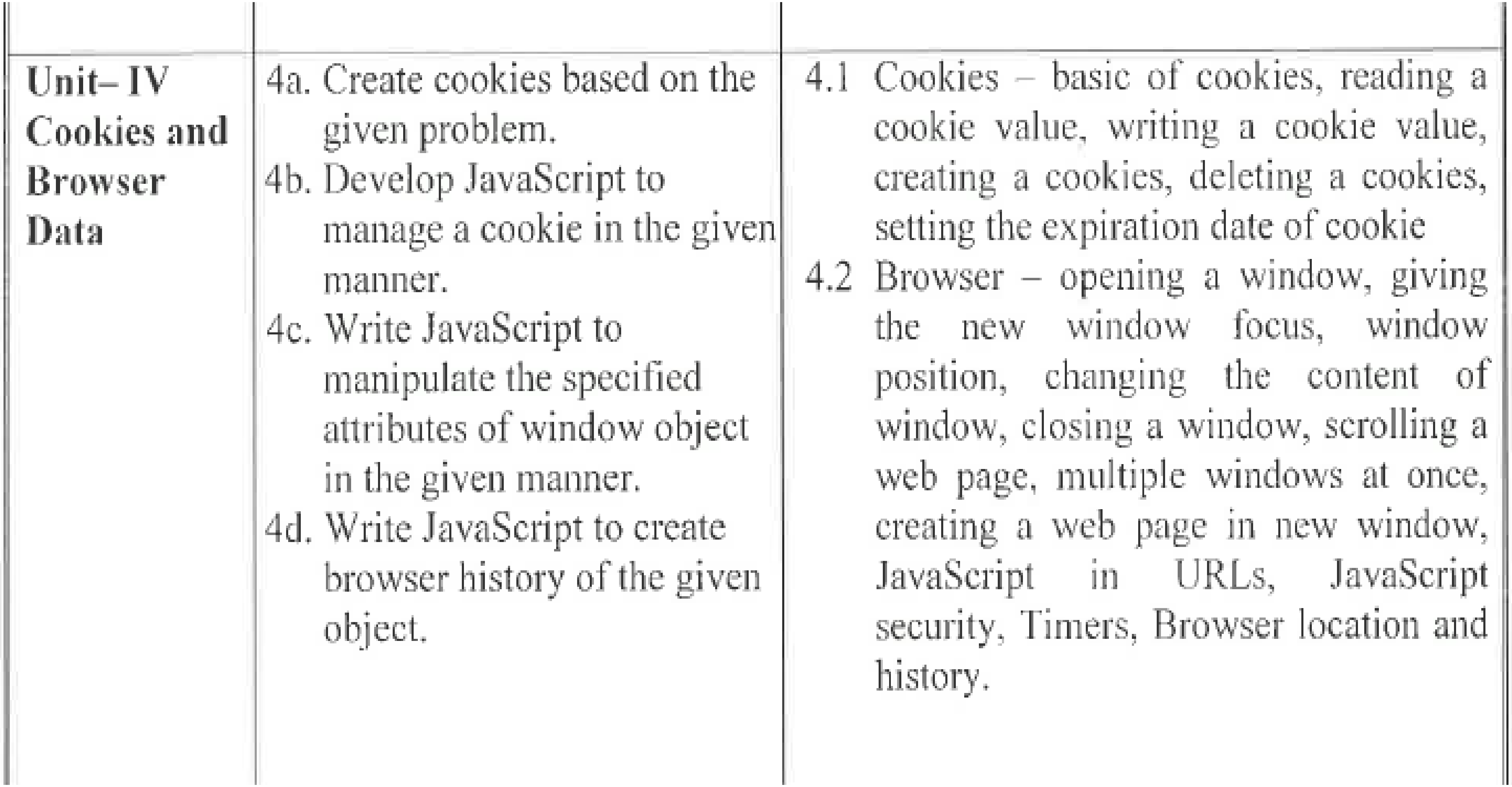
UNIT NO 04

Cookies and Browser data

Marks-8



Unit– IV Cookies and Browser Data	4a. Create cookies based on the given problem. 4b. Develop JavaScript to manage a cookie in the given manner. 4c. Write JavaScript to manipulate the specified attributes of window object in the given manner. 4d. Write JavaScript to create browser history of the given object.	4.1 Cookies – basic of cookies, reading a cookie value, writing a cookie value, creating a cookies, deleting a cookies, setting the expiration date of cookie 4.2 Browser – opening a window, giving the new window focus, window position, changing the content of window, closing a window, scrolling a web page, multiple windows at once, creating a web page in new window, JavaScript in URLs, JavaScript security, Timers, Browser location and history.

4.1 Cookies

Basic of Cookies

- The user information present in web pages can be stored by using cookie.
- Cookie is a small piece of information sent from the website and stored in a file on the use's computer.
- The information sent from the website is specific to a user.
- It is stored by the user's web browser, and web browser can access that information.

4.1 Cookies

Cookie works as follows :

1.Storing information

When user visits the website the basic information such as username and password can be stored in a cookie.

2. Accessing information :

- The stored username and password can be accessed by the web browser which is remembered by the browser in the form of cookie.
- Cookies are saved in form of name-value pair as follows: Username ="student";

It contains following data.

1. A name-value pair
2. An expiry date for that cookie
3. The website/Domain name

4.1 Cookies

Types of Cookies:

There are three types of cookies.

(a) Session cookie (b) Persistent cookie (c) Third party cookie

(a) Session Cookies

Session cookies are temporary cookies that remember user's online activities throughout a session. i.e. it remember the information when a user logged in until logout or when user close the web page. When a user gets logged out the session cookie gets expired.

(b) Persistent Cookies

Persistent cookies assigned an expiration date. It does not get expired until it reaches to the expiration date. It do not expire with users logged out and closing web page. It is stored in subfolder in folder of browser's installation. JavaScript can create, read and delete cookies using ***document.cookie*** property.

4.1 Cookies

Creating a Cookie

To create a cookie we need to assign a value to cookie by using document.cookie as follows :

document.cookie = "username = student“;

You can also add an expiry date to the cookie in UTC time format, as follows:

document.cookie ="username = student;expires = Thu, 4 Jul 2019 11:30:00 UTC“

4.1 Cookies

Reading a Cookie Value

- When a cookie is created it can be read by the browser.
- The value of the *document.cookie* object is the cookie and it is in the string format.
- The *document.cookie* string will keep a list of name=value pairs separated by semicolons (:), where name is the name of a cookie and value is its associated string value to that name.
- After adding a new name value pair to the *document.cooki*e, that new pair will concatenate to the string of *document.cookie.*
- Using split() function we can get the name and values.
- we get two elements; first is name and second is value.
- We can access those elements as array[o] and array[1] name and value respectively.

4.1 Cookies

Writing a Cookie Value

- Writing a cookie is same as creating a new one.
- If we want to change the value or expiry date of the cookie, we have to rewrite the cookie as follows:

- **document.cookie = "username = student“;**

- **document.cookie = "username = admin“;**

4.1 Cookies

Deleting a Cookie

- If the cookie is temporary then the cookie will deleted automatically after closing the browser.
- if we want to delete the temporary cookie without closing the browser, we must update the expiry date of the cookie to the past date. Also to delete the persistent cookie we must update the expiry date of the cookie to the past date manually.
- Then automatically cookie gets expired and it will be deleted by the browser.
- We used to set the date 1 January 1970 in the format: "Thu, 01 Jan 1970 00:00-01 GMT as follows.

```
document.cookie ="cookie_name=cookie_value; expires =Thu, 01 Jan 1970 00:00:01 GMT”;
```

4.1 Cookies

Setting the Expiration Date of Cookie

- You can extend the life of a cookie by setting a new expiration date to the cookie.
- This can be done by setting the 'expires' attribute to a new date and time.
- The new date object is obtained by using *'new Date()'.*
- The Date has following function to get the month of the date to set the month of the date and to format the date string.

1. **getMonth() :** this function returns the month of the current date object.
2. **setMonth() :** this function set the month to the current date object
3. **toUTCString() :** this function formats the date in the standard format to set to the expiration date of the cookie.

4.2 Browser

- We can perform various operation with browser window in JavaScript.
- Following are the operations
  - ✓ Opening a window
  - ✓ Giving new window focus
  - ✓ Window position
  - ✓ Changing content of widow
  - ✓ Closing the window
  - ✓ Scrolling window
  - ✓ Multiple window

4.2 Browser

- **Opening a window-**
- **open() function**

This function is used to open a new window from current window. It is invoked on window object.

syntax

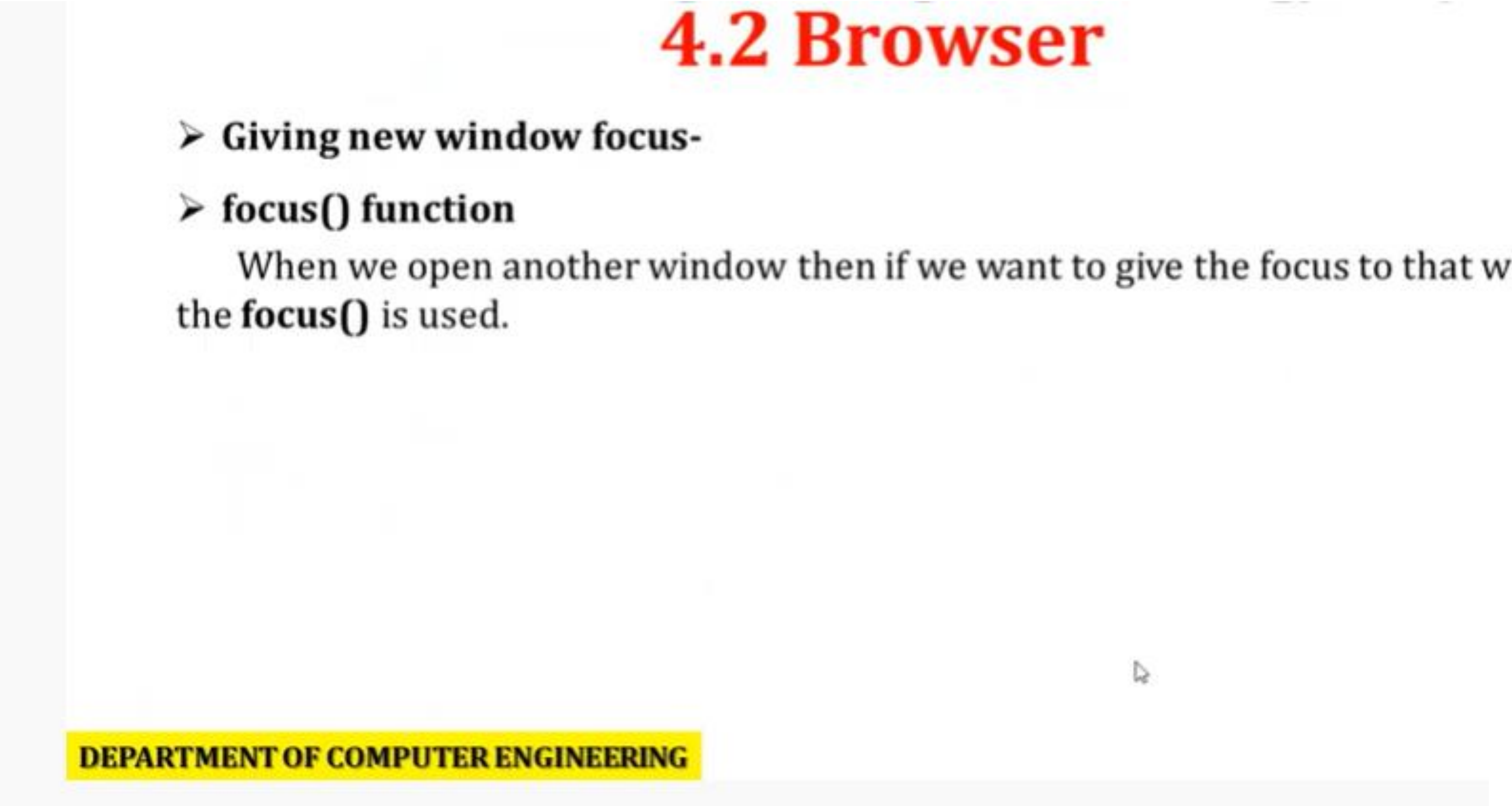
window.open(url,name,style);

url-it specifies the URL of the new page going to be open in new window.

Name-it is used to set the name of the window.it is optional.

Style-style of the window has various parameters such as

scrollbar,toolbar,height,width,location,etc, it is optional.



4.2 Browser

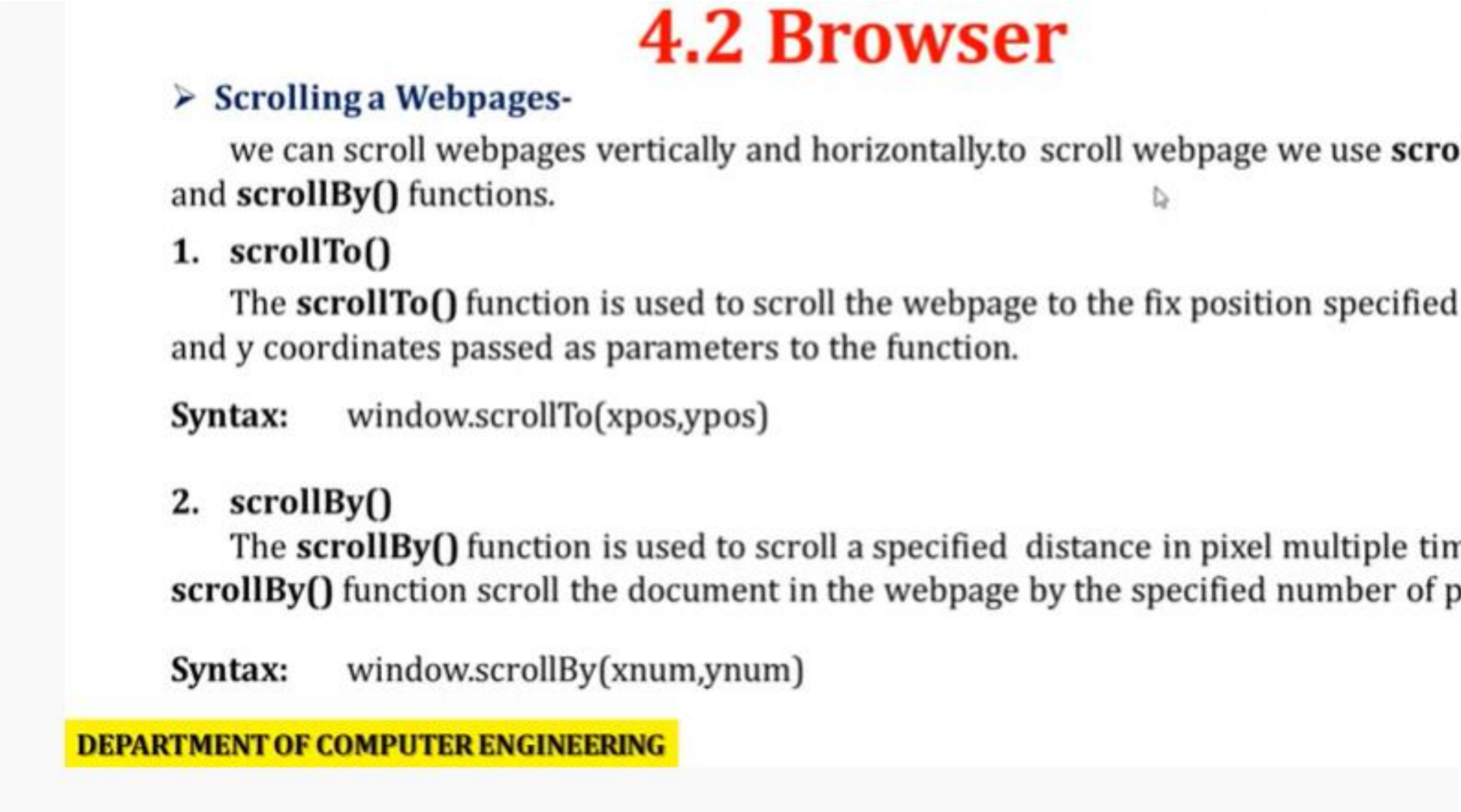
- **Giving new window focus-**
- **focus() function**

When we open another window then if we want to give the focus to that window the **focus()** is used.

4.2 Browser

- **Closing the window -**
- **close() function**

The **close() function** is used to close the window.by using **close()** functions we can only close the which we open with the help of **open()** function.



4.2 Browser

➢ **Scrolling a Webpages-**

we can scroll webpages vertically and horizontally.to scroll webpage we use **scrollTo()** and **scrollBy()** functions.

1. **scrollTo()**

The **scrollTo()** function is used to scroll the webpage to the fix position specified by x and y coordinates passed as parameters to the function.

Syntax: window.scrollTo(xpos,ypos)

2. **scrollBy()**

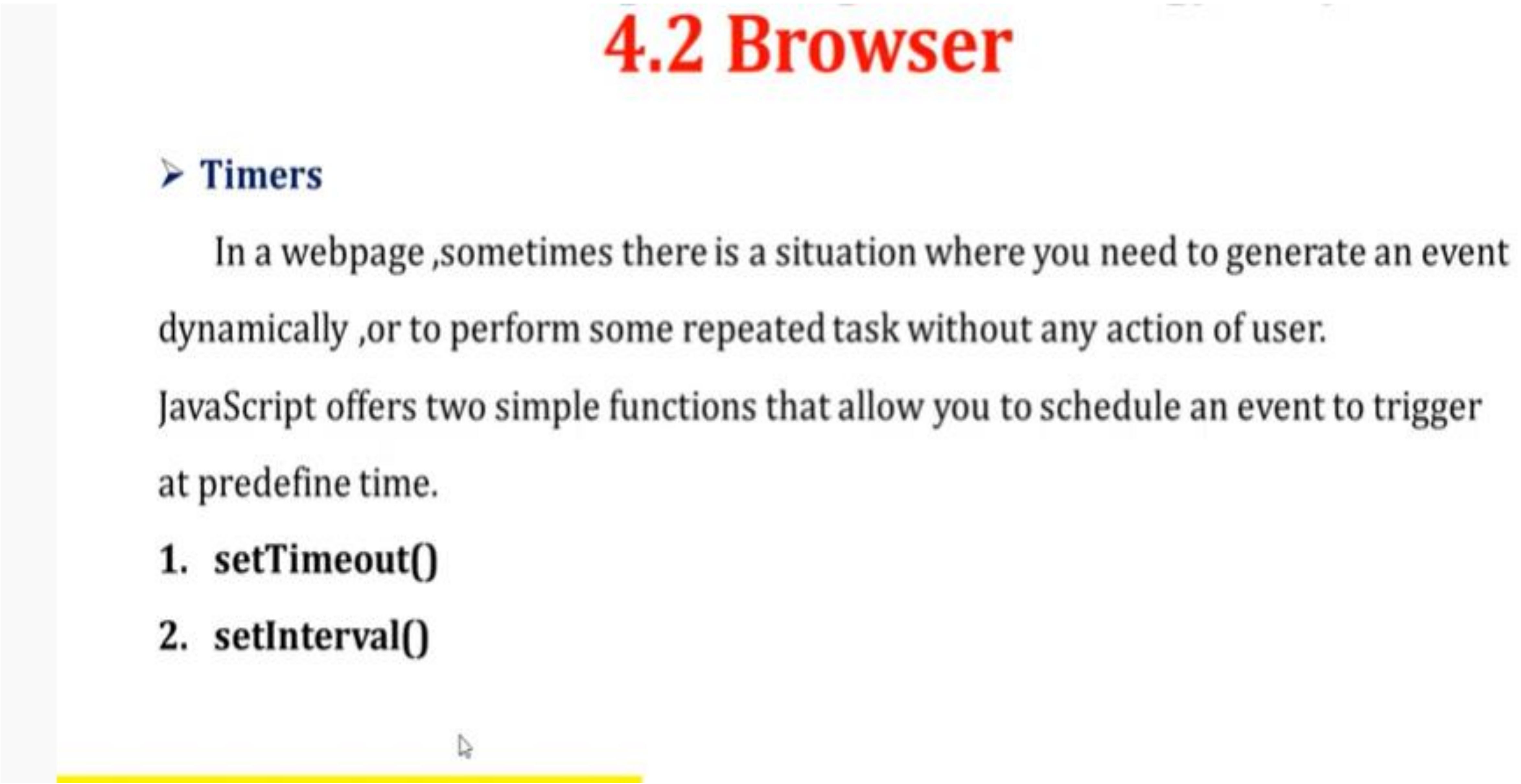
The **scrollBy()** function is used to scroll a specified distance in pixel multiple times. The **scrollBy()** function scroll the document in the webpage by the specified number of pixel.

Syntax: window.scrollBy(xnum,ynum)

4.2 Browser

- **Multiple window at a Glance-**

we can open multiple windows by using ***window.open()*** function multiple times. also we can use for loop to open multiple windows. The last open window will get the focus by default.

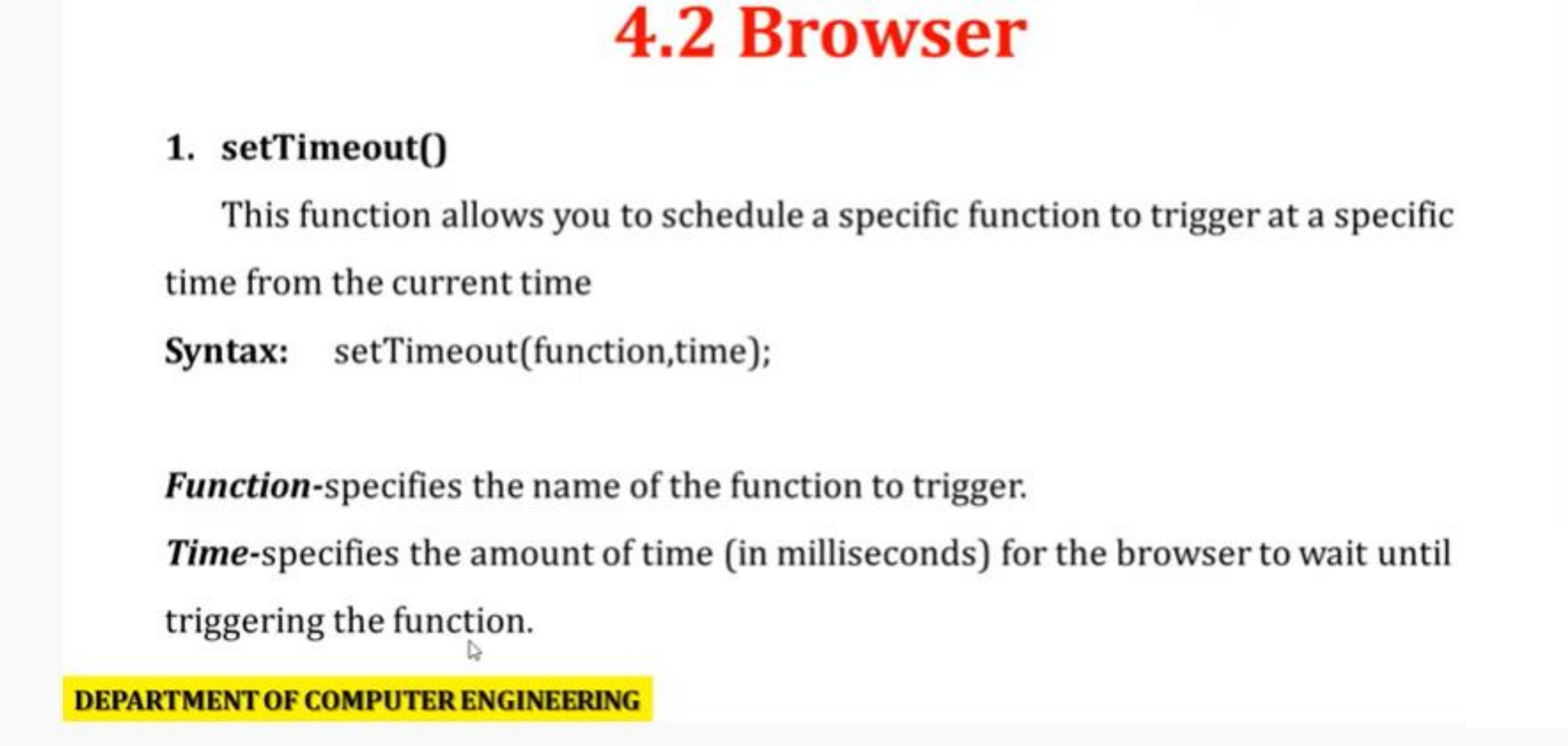


4.2 Browser

- **Timers**

In a webpage ,sometimes there is a situation where you need to generate an event dynamically ,or to perform some repeated task without any action of user. JavaScript offers two simple functions that allow you to schedule an event to trigger at predefine time.

1. **setTimeout()**
2. **setInterval()**



4.2 Browser

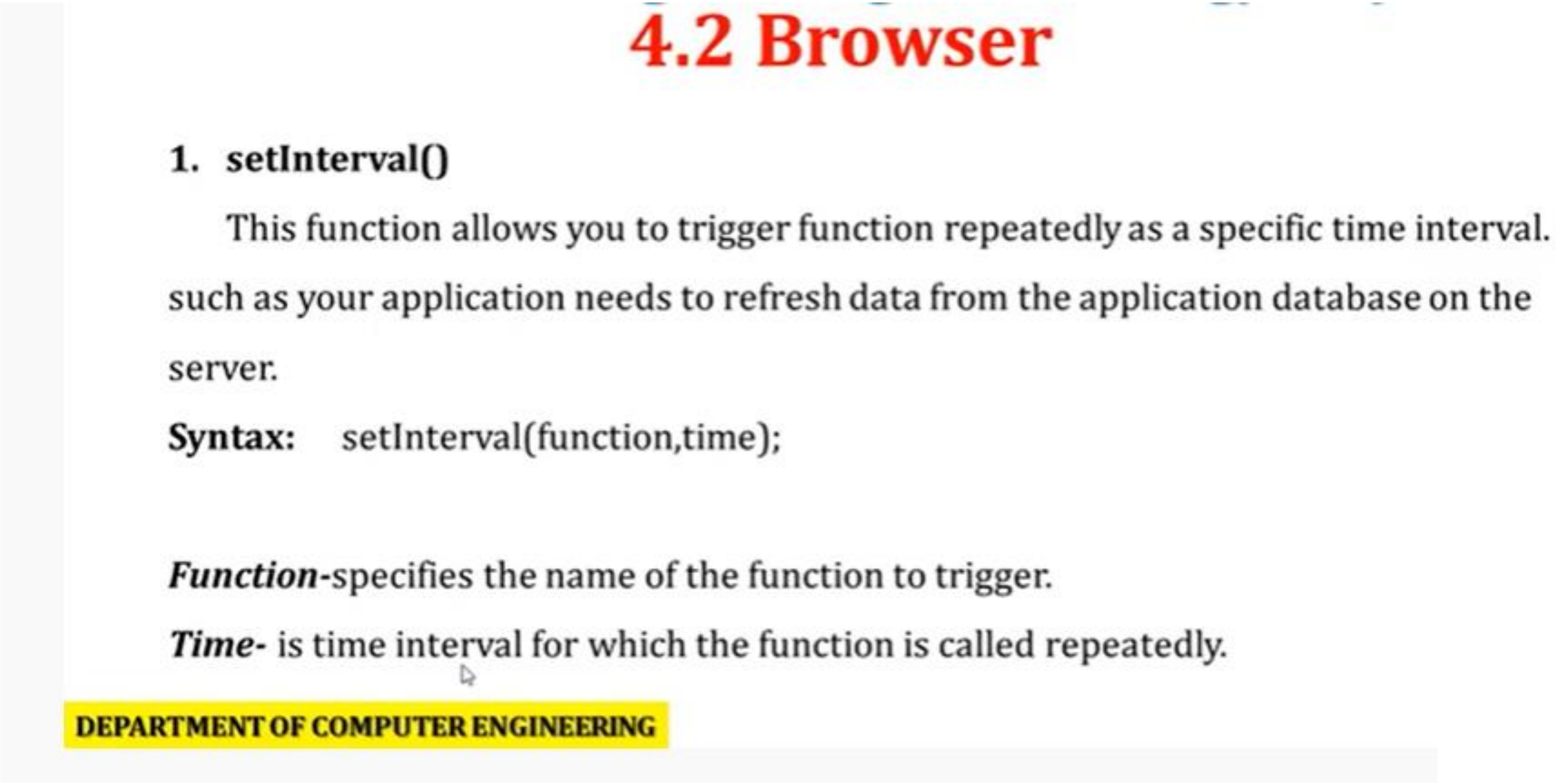
1. **setTimeout()**

This function allows you to schedule a specific function to trigger at a specific time from the current time

Syntax: setTimeout(function,time);

Function-specifies the name of the function to trigger.

Time-specifies the amount of time (in milliseconds) for the browser to wait until triggering the function.



4.2 Browser

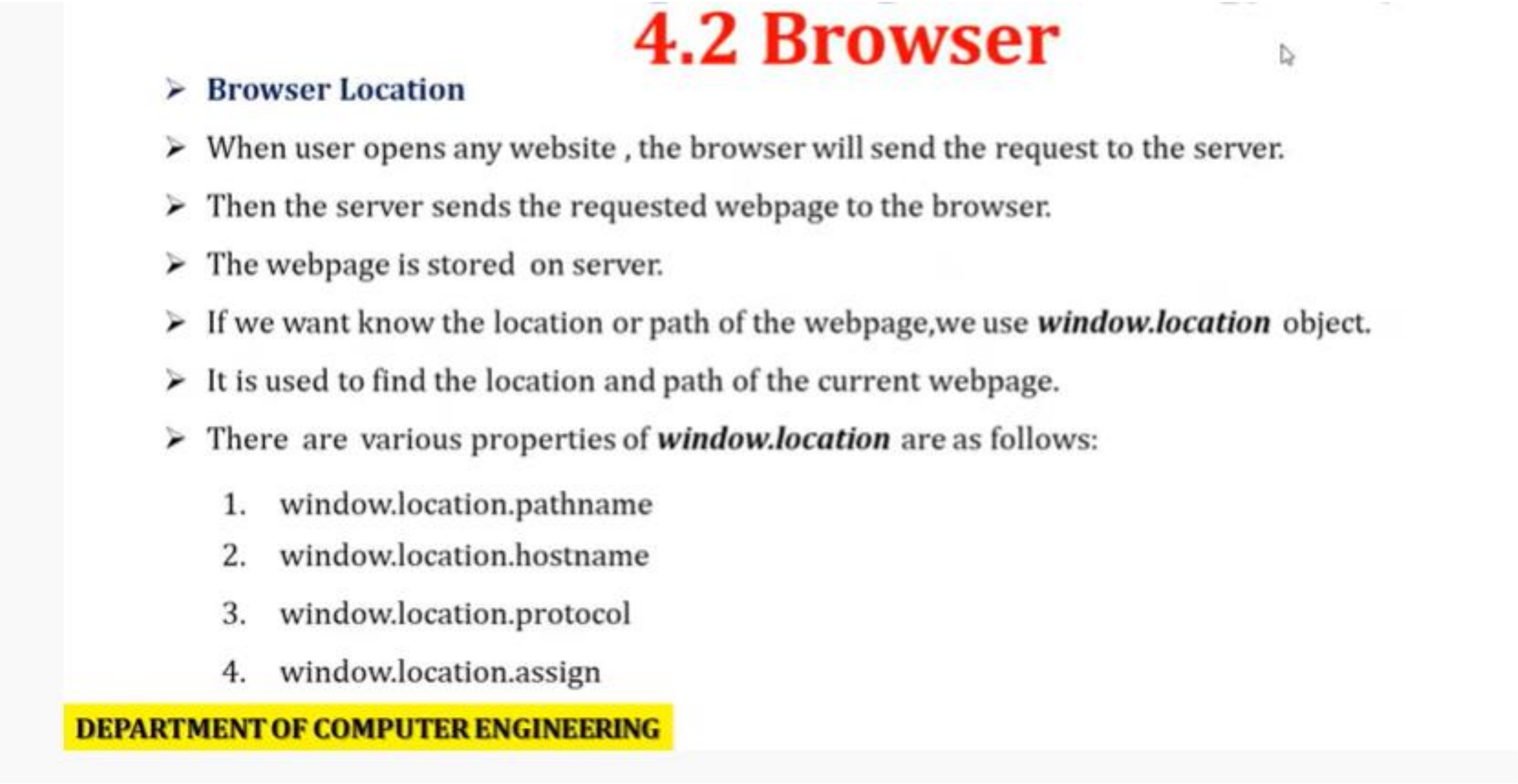
1. **setInterval()**

This function allows you to trigger function repeatedly as a specific time interval. such as your application needs to refresh data from the application database on the server.

Syntax: setInterval(function,time);

Function-specifies the name of the function to trigger.

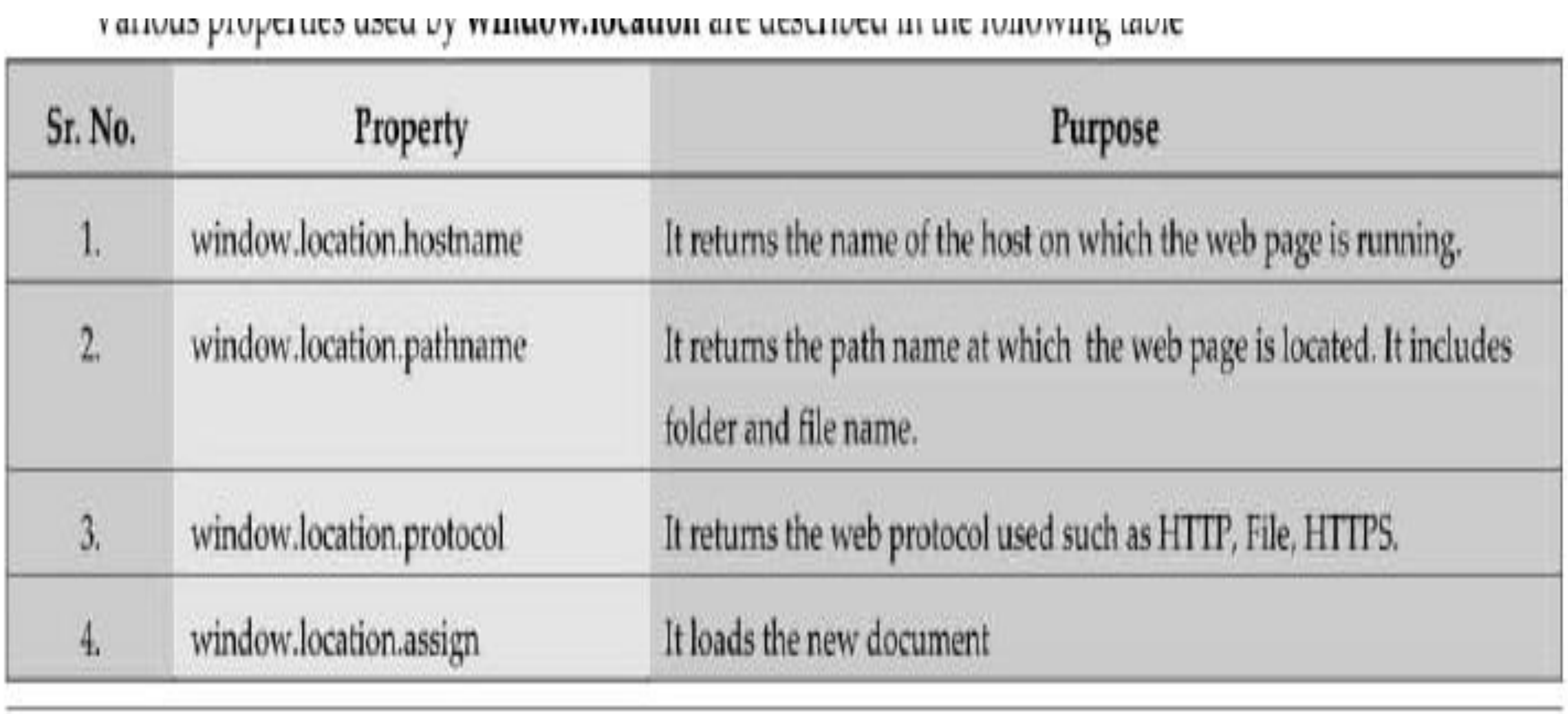
Time- is time interval for which the function is called repeatedly.



4.2 Browser

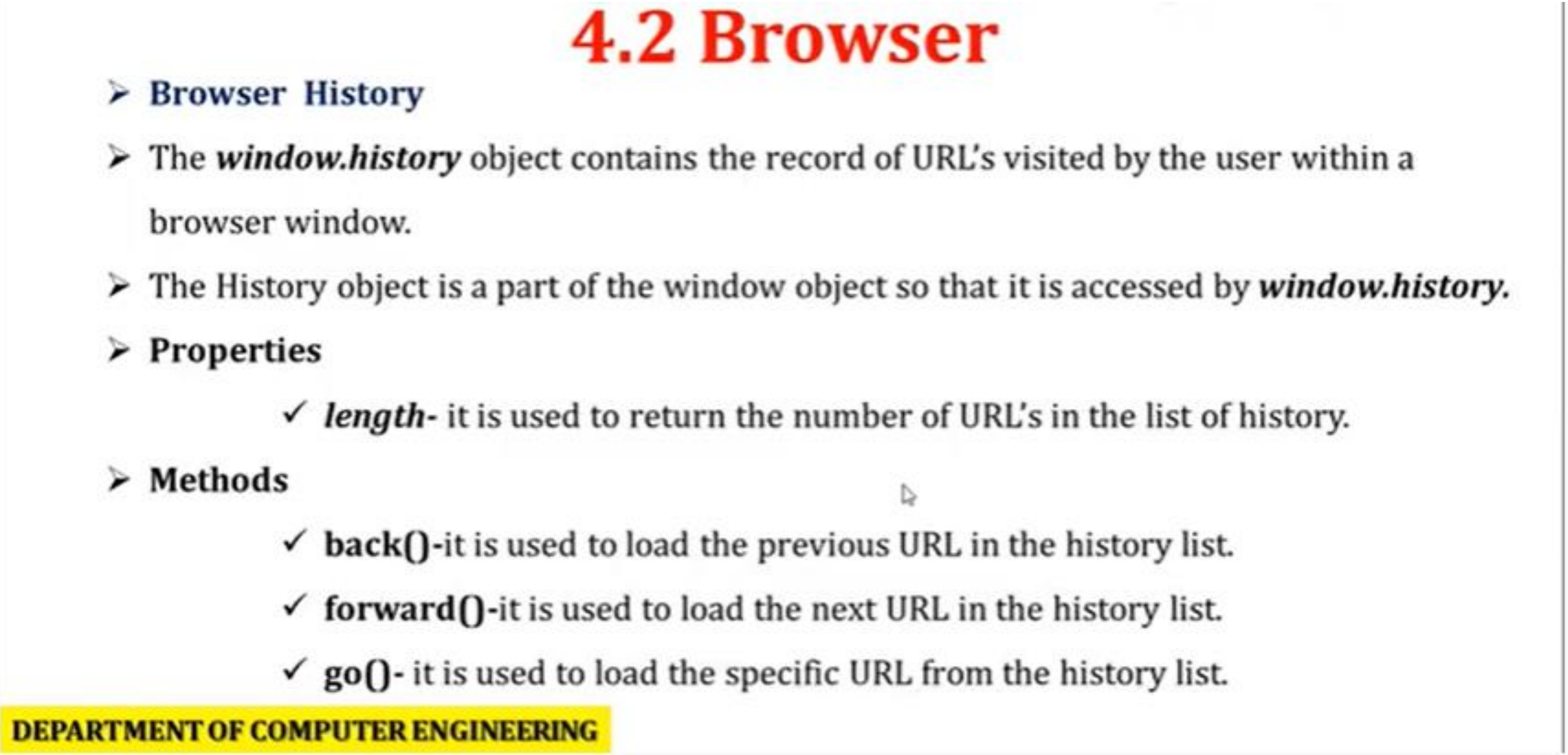
- **Browser Location**
- When user opens any website , the browser will send the request to the server.
- Then the server sends the requested webpage to the browser.
- The webpage is stored on server.
- If we want know the location or path of the webpage,we use ***window.location*** object.
- It is used to find the location and path of the current webpage.
- There are various properties of ***window.location*** are as follows:

  1. window.location.pathname
  2. window.location.hostname
  3. window.location.protocol
  4. window.location.assign



Various properties used by **window.location** are described in the following table

Sr. No.	Property	Purpose
1.	window.location.hostname	It returns the name of the host on which the web page is running.
2.	window.location.pathname	It returns the path name at which the web page is located. It includes folder and file name.
3.	window.location.protocol	It returns the web protocol used such as HTTP, File, HTTPS.
4.	window.location.assign	It loads the new document



4.2 Browser

- **Browser History**
- The ***window.history*** object contains the record of URL's visited by the user within a browser window.
- The History object is a part of the window object so that it is accessed by ***window.history.***
- **Properties**
  - ✓ ***length***- it is used to return the number of URL's in the list of history.
- **Methods**
  - ✓ **back()**-it is used to load the previous URL in the history list.
  - ✓ **forward()**-it is used to load the next URL in the history list.
  - ✓ **go()**- it is used to load the specific URL from the history list.