TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN & TT

BÀI GIẢNG

CT188

NHẬP MÔN LẬP TRÌNH WEB

4
JavaScript (JS)

dung

JavaScript Engine

DataTypes

Statements

CallBack Function

DOM

JavaScript là gì?

JavaScript (JS) là ngôn trình thông (interpreted programming language)

Là trong công lõi Word Wide Web

xây các cho website có tác phía client

Mã nhúng vào các trang web và thi trình web

Các trình tích máy thi mã JS là JavaScript Engine

ra

phát Brendan Eich, nhân viên cty Netscape 1995

Phiên tiên có tên LiveScript phát hành trình web Netscape Navigator

1996 Microsoft phát hành Internet Explorer (IE) tranh Netscape và ra ngôn JS tên Jscript

Khác Jscript và JS gây khó cho các nhà phát website trên 2 trình IE và Netscape

ra

Netscape tiêu lên ECMA* yêu
các nhà phát trình tuân các tiêu
chung

này chính phát hành
ngôn ECMAScript (ES) tiên vào tháng 6
1997

Quá trình tiêu hóa trong
theo

Phát hành ES2 vào tháng 6 1998

ES3 vào tháng 12 1999.

Công trên ES4 vào 2000.

** European Computer Manufacturers Association là một tổ chức đặt ra các tiêu chuẩn cho các hệ thống thông tin và truyền thông*

ra

Trong khi) Microsoft ngày càng
trên trình sau khi ra
Internet Explorer (IE).

2000, IE 95%.

này JScript thành tiêu cho ngôn
phía máy khách.

Ban Microsoft tham gia vào quá trình tiêu
hóa và ngôn JScript
mình, cùng tác trong
công ECMA. Do) ES4 .

này ngôn mãi cho
2004 khi Mozilla, Netscape
phát hành Firefox

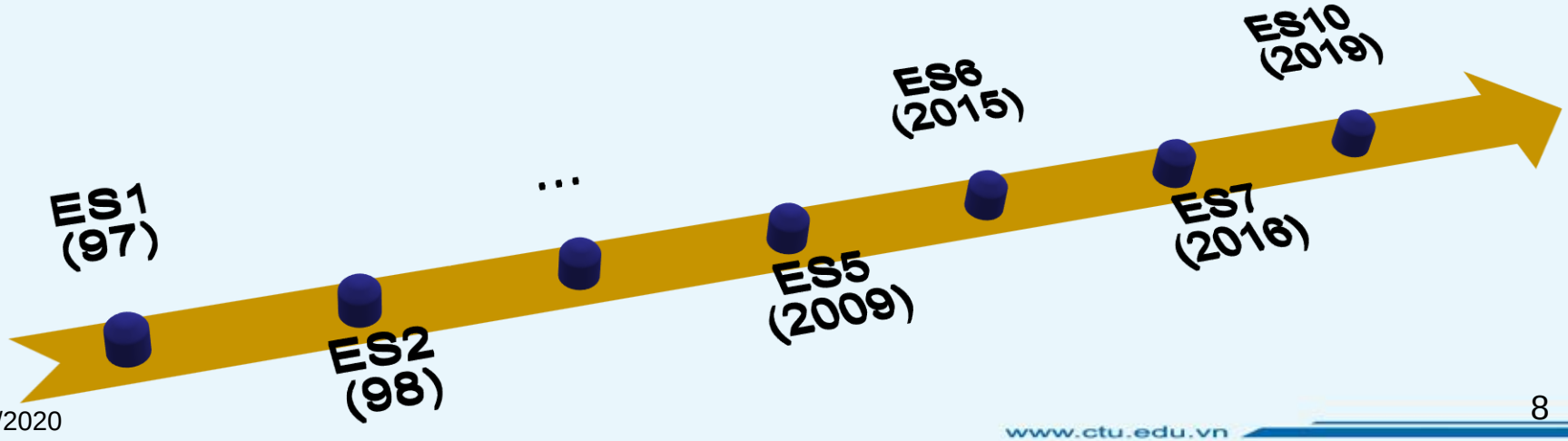
ra

2005, Mozilla gia ECMA International hoàn ES4 không thành công khi không có tác Microsoft

2008 Google ra trình Chrome Engine V8 nhanh nâng JS lên cao

ES5 phát hành 2009 sau khi các công ty,

ES6 phát hành 2015 tiêu thêm vào



JavaScript Engine là gì?

Là trình phân tích cú pháp và thi mã JS tích trong các trình

Phiên tiên phát Brendan Eich tên **SpiderMonkey**

SpiderMonkey **Firefox Browser**

V8 engine (2008) trong **Google Chrome** giá các engine

Webkit engine phát **Apple** và trong trình Safari

Carakan trong **Opera**, sau này sang V8

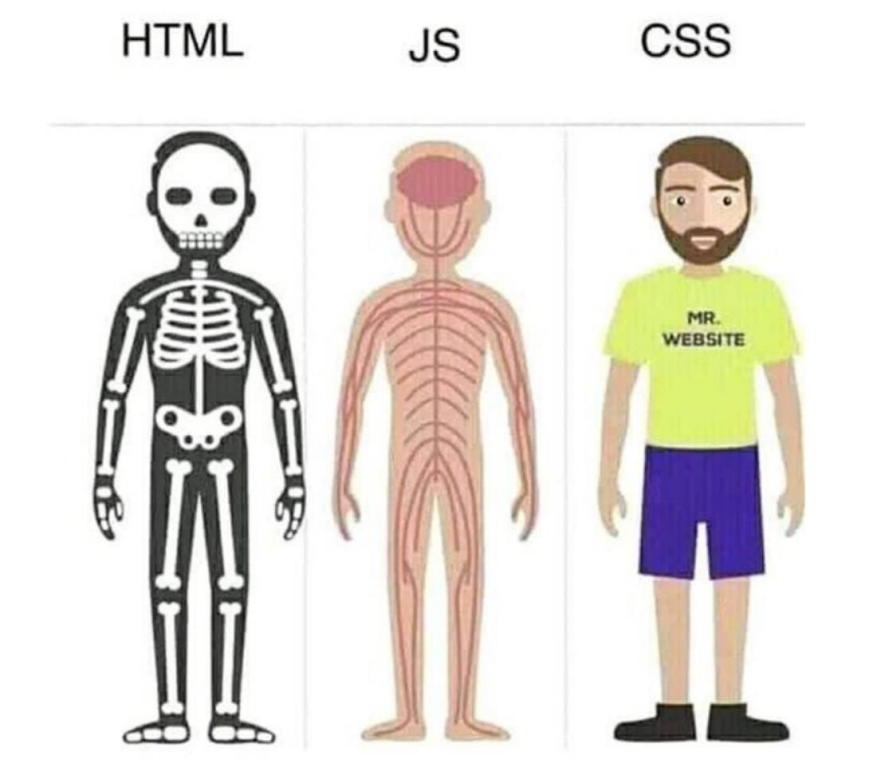
Chakra (JScript9): **IE, Microsoft Edge**

sao JavaScript?

HTML mô trúc và dung page

CSS làm cho page

JS giúp cho page có tác



webpage

JS có làm gì?

Làm cho giao web linh

Các

Thêm/ /thay dung trên trang web

Form validation: tra trên form

tra : không , ,
tra , vv

lý các dùng

thi các khi dùng rê , nút, gõ phím,

/ghi truy trang dùng (cookie)

Giao server / (Ajax)

Làm sao JS?

Mã JS có trong file HTML (inline script) hay nhúng tin ngoài (external file script)

Cách 1: Inline script, code trong

- Các tính và khai báo các trình , tuy nhiên tính có qua vì các trình không quan tâm tính này.

Làm sao JS?

Cách 2: External file script, code trong tin riêng

Ví 1.1: Nhúng mã js trong file trên cùng host

Ví 1.2: Nhúng mã js trong file khác host host

Lưu ý: Không chèn code js vào trong thẻ <script></script> ***khi nhúng code từ tập tin***

Code JS ?

Mã JS trong trang HTML, sau

Trong , mã JS trong <head> trang

mã JS có tác các trên trang, nó sau , thông không gây load trang

các file JS là quan , trong file1 mà file2 thì file1 file2.

Code JS ?

Ví 2. Cách mã JS trong trang HTML

thi mã JS

Mã JS thi theo trên

inline script, mã lênh thi ngay khi web browser load trí script trên page

external file inline script, dung download và thi ngay khi browser load trí nhúng file

tính (áp cho external script) trong tình script thi **sau khi browser load xong dung page** (khi trình)

tính có tác trên các trình phiên IE, Firefox, Safari và Chrome, các trình khác qua tính này

thi mã JS

Ví 3: mã JS thi sau khi load xong page

thi mã JS

Trong external script quá thì:

gian download script

load trang do browser

xong file script và thi xong load dung

tình này, tính async

quá trình trì hoãn load dung trang

tính qui :

Browser load dung page, không external script download xong

Script thi ngay sau khi nó download xong

→ thi không theo chèn script

Ví 4: defer vs async

Cho thi các script?

Syntax

Cú pháp C, Java

Case-sensitive: JS phân ký HOA

Identifiers:

danh 1 ký , (_) $, ví

Các ký theo có là , , hay _

danh có ký Unicode không cáo, ví

Các khóa không tên danh (, hàm,..), ví

Syntax

Comments: các chú thích

C

Statements: các trên 1 dòng,

thúc (;)

không có, tuy nhiên

không cáo

Strict Mode

strict mode trong EMAScript v5

khóa thông báo cho JS engine thi code trong

strict mode thi mã JS cách an toàn, ví thông báo khi tên object khai báo, không cho phép tham hàm trùng tên,

khóa tin JS trong block code thì có tác trong block

Strict Mode

Ví dụ 4: strict mode

Variable

? trong JS khai báo không ,
xác khi gán

? khai báo thông qua khóa

Trong , khai báo
khi

Trong , không khai
báo , **có vi global**

khai báo tên cùng lúc, các cách nhau

Ví var n = 2, i = 4, k=6;

có vi trong hàm (function scope) và có
vi trong (block scope)

dùng khai báo ,

Variable

Ví dụ 5: khác nhau và

Giá trị undefined có nghĩa rằng biến đã được khai báo nhưng chưa gán trị

Scope

Trong JS có 4 vi là và

: khi khai báo bên ngoài function, có vi toàn

Scope

Ví dụ 6: Biến cục bộ

Data types

Trong ES có 5

- undefined
- null
- boolean
- number (float, int, decimal)
- string

Ngoài ra còn có thêm 2 khác là object và function

Toán

toán tử typeof có thể sử dụng như hàm
typeof(var)
typeof(null) là object!

undefined

Khi khai báo gán giá có là

Khi truy khi nó khai báo

Truy tính không

Ví 7

Number type

JS phân 2 là và

có bát phân

(octal) và phân (hexa)

octal:

hexa:

JS xem có

Number type - casting

JS 3 hàm ép và

Number() các quy ép :

- Ép : → 1; → 0
- Ép 0
- Ép , (Not-A-Number)
- Ép :
  - → →
  - →
  - →
  - →
  - →

Number type - casting

dùng ép sang các quy sau:

/

ký là ký →

ký là ký

sang

ký theo cho ký cùng

ký không là nó

ví : parseInt%123 ; // 123

phân

octal và hexa giá

tham 2 hàm là tham tùy ,
sang nào (phân, bát phân, phân)

Number type - casting

String

trong

ký theo sau \ không xem là ký

Toán dùng ghép 2

Ví 9.1: Khai báo

```
<script>
	var firstName = "Dang";
	var lastName = 'Vo';
	var fullName = "Dang H Vo'; // syntax error
	document.write("He said, \"hey.\""); // He said, "hey."
	document.write("Java" + "Script"); //JavaScript
</script>
```

String

Các (method) và tính (property)
lý
: dài
: trí ,
-1 không tìm
:
trí cùng
: trích con
, và là âm thì trí
sang trái

không tham có giá âm
:
tham 2 là dài con

String

Các (method) và tính (property)
lý

: sang HOA

: sang

: whitespace và sau

: ký index

: mã Unicode ký

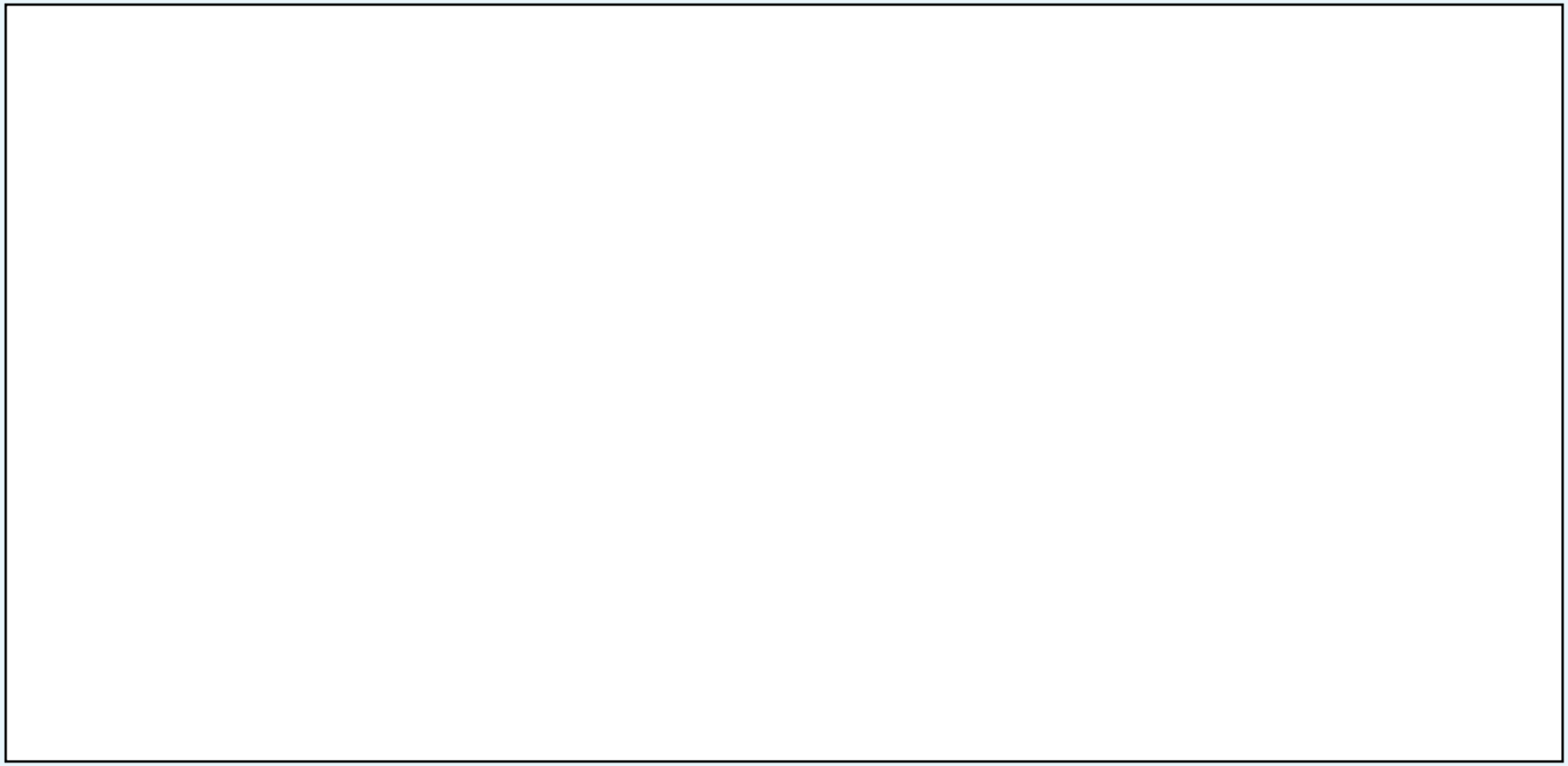
: tách thành ký , phân
cách char

: thay str1 str2

Ví 9.2: các tính và

if statement

if (condition) statment1 else statement 2



do-while statement

khi không còn

```
var i = 0;
do {
        i += 2;
} while (i < 10);
```

while statement

tra khi thi statement, khi không còn

for statement

forEach statement

qua các trong

tham là 1 function (callback function)

item: là

index: trong collection

switch statement

function

Cú pháp chung

Hàm có khóa , không nào

Các dòng sau khóa không thi

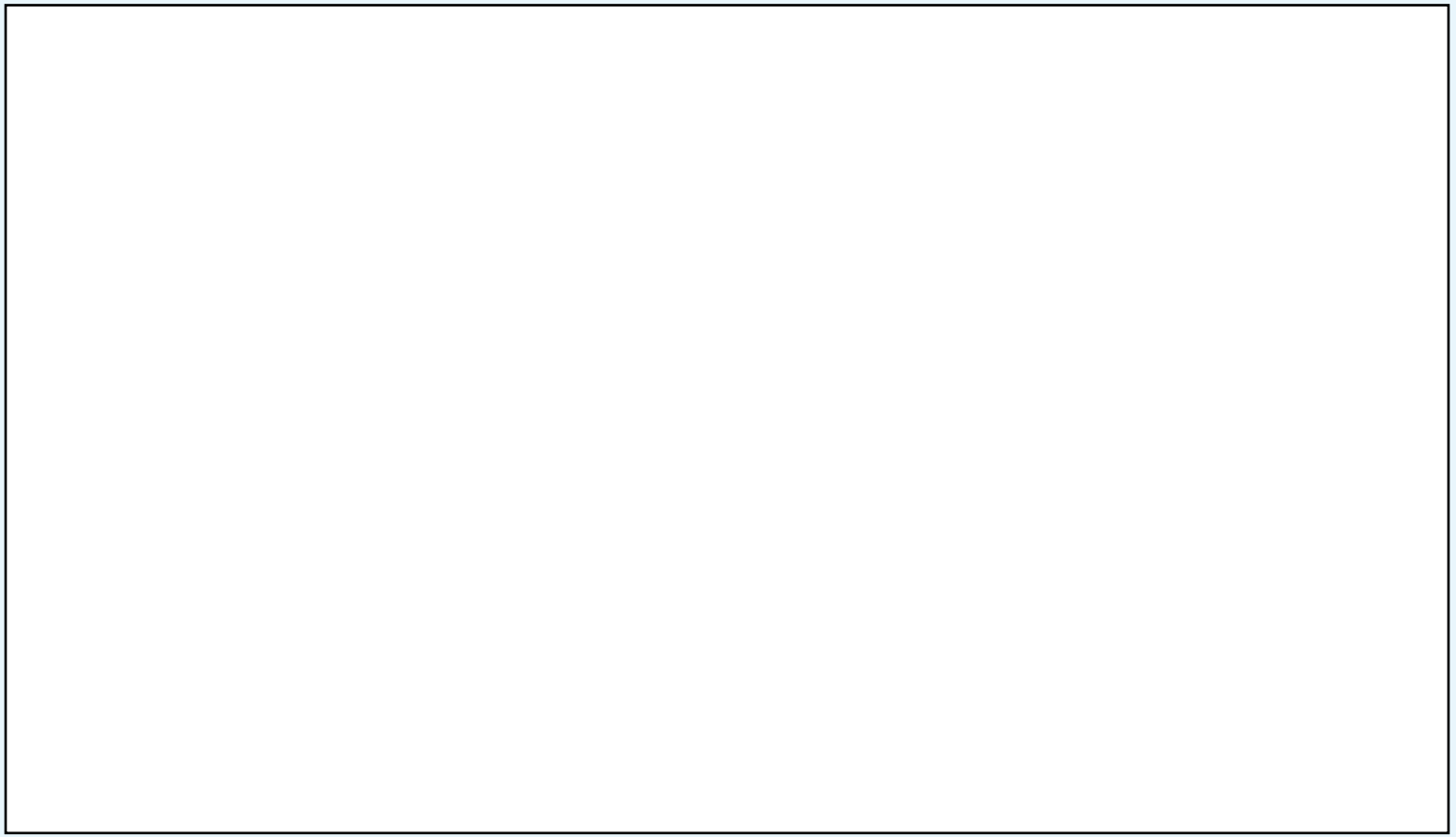
Có có khóa trong hàm, có 1 thi

function

function - arguments

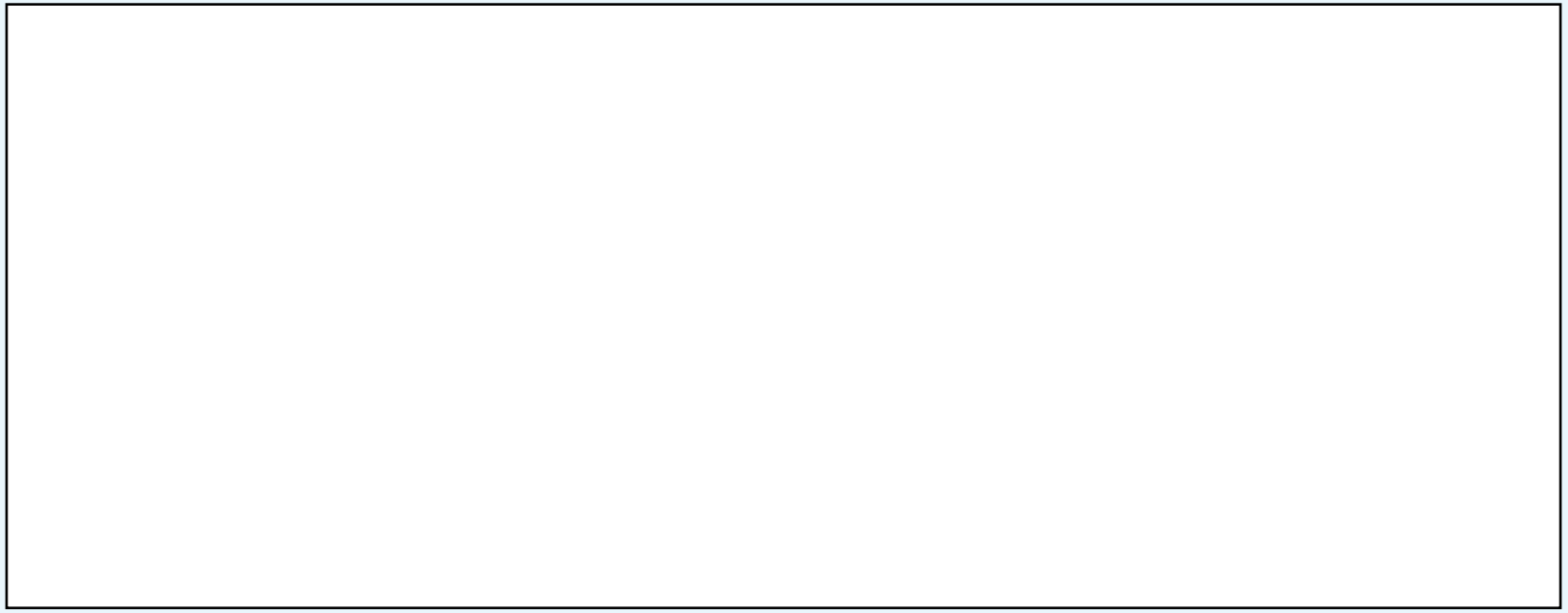
các tham hàm không xác

→ không overload hàm C hay Java



function - arguments

có truy tham hàm thông qua tên object



function - arguments

Tham

tham có giá

tham có giá

function - arguments

Hàm không tham

ví 10: truy tham qua object arguments

tham truy thông qua object

function - arguments

Hàm không tham (ES6)

Ví 11: truy danh sách các tham

```
function sum(...args){
  let total = 0;
      for(let i in args) {
            total += args[i];
      }
  return total;
}
var t;
t = sum(2, 2, 3); // 7
t = sum(2, 2, 3, 2, 2, 4); //15
```

các tham truy thông qua tên tham

thi hàm trong JS

JS thi các hàm không theo

Xét ví bên

Hàm fn2() thi hàm fn1() dù nó sau do fn1 delay 500ms

➔ fn2() cho ra

Làm nào fn2() thi sau fn1()?

Callback function

Callback function là hàm tham là hàm khác

Các hàm theo

Ví 12: Callback function

```
function myFunc(str) {
        console.log("hi " + str);
}

function test(msg, myFunc) {
    console.log(msg);
    myFunc(msg);
}

test("Dang", myFunc);
// output
// Dang
// hi Dang   <-- called by myFunc
```

Callback function

Anonymous function là hàm không có tên

Ví 13: Anonymous Callback function

```
function test(msg, myFunc) {
        console.log(msg);
        myFunc(msg);
}

test("Dang", function(str){
        console.log("Hello " + str);
});

// output
// Dang
// Hello Dang <-- called by anonymous function
```


JavaScript DOM
JavaScript

Document Object Model (DOM)

Trong trang HTML, element xem là object

Element các element → nên trúc phân cha con

trúc phân này là cây, có nút là window

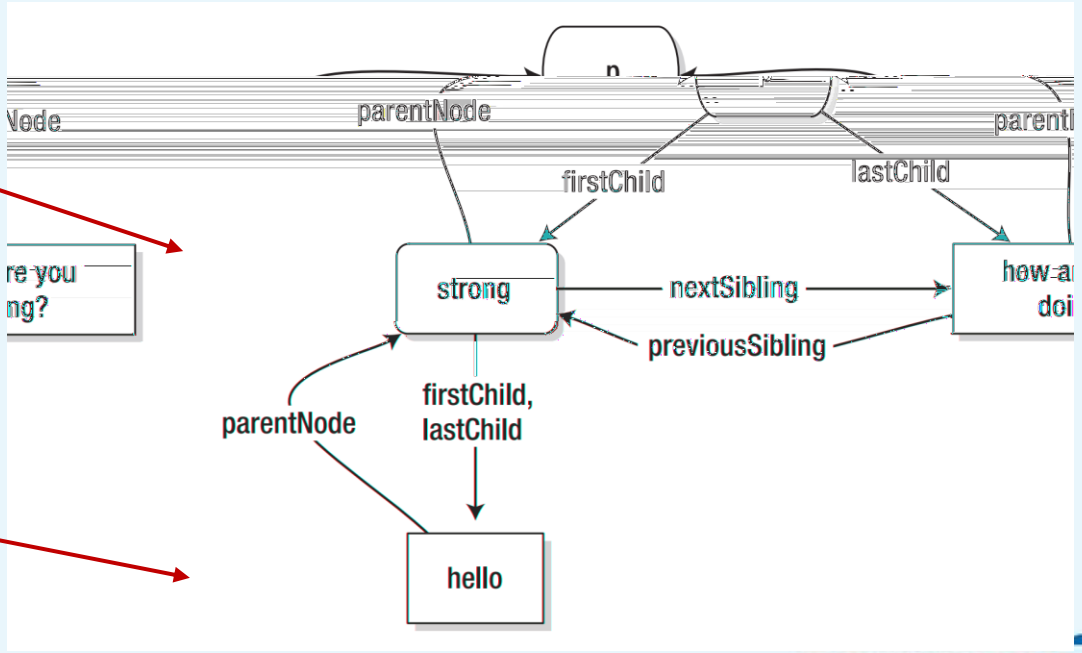
Các object có là element, text, document, comment, tag là các nút (node)

Ví dụ:

```
<p><strong>Hello</strong> how are you doing?</p>
```

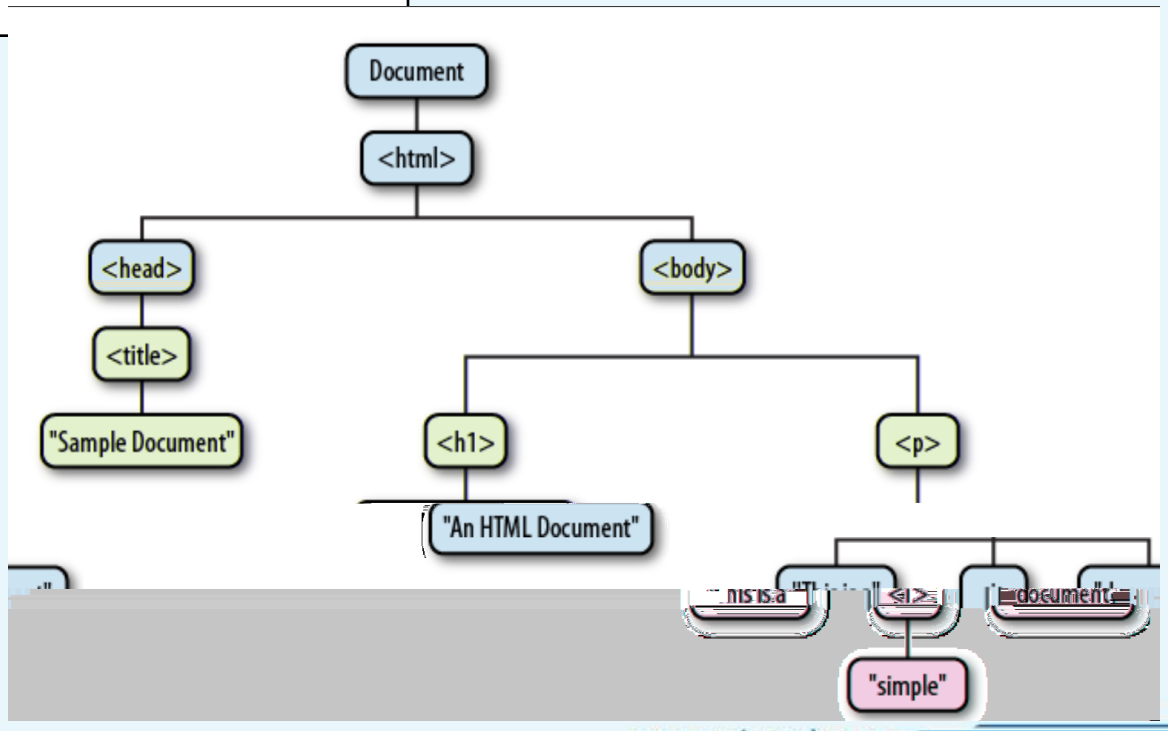
element node

text node



Document Object Model (DOM)

Ví dụ 14: Cấu trúc DOM của trang HTML



Document Object Model (DOM)

JS cung các cho phép thao tác trên DOM bao

DOM

Thay tính, giá , các element trên trang

Chèn thêm vào node trên trang HTML

trên trang HTML

Tìm các khác nhau

Các tính truy DOM

Property	Description
document.baseURI	URL trang html
document.body	<body>
document.cookie	cookie
document.doctype	doctype
document.documentElement	<html> trang
document.domain	tên , ví yoursite.com
document.forms	các form

Các tính truy DOM

Property	Description
document.head	<head>
document.images	<img>
document.inputEncoding	mã hóa trang (character set)
document.links	<area> và <a> có tính href
document.readyState	thái (loading) trang
document.scripts	<script>
document.title	<title>

DOM

:

có tính

:

có tính

:

theo , ví ,

:

có tính

:

các CSS

Lưu ý: (2),(3),(4), (5) trả về kết quả là 1 danh sách các node thỏa mãn điều kiện

DOM

Ví dụ 15: Tạo và thay đổi thuộc tính

;

DOM

Ví dụ 16:

DOM

: danh sách các node con, bao text node và element node

: danh sách **element node** con

: node con tiên node, node không có node con null

: element node con

: node con cùng node

: element node con cùng

DOM

: node cha

: element node cha

: tìm node tiên ancestor node

: node anh em

: element node

: node anh em trái

: element node anh em trái

Ví dụ 17: element nodes con

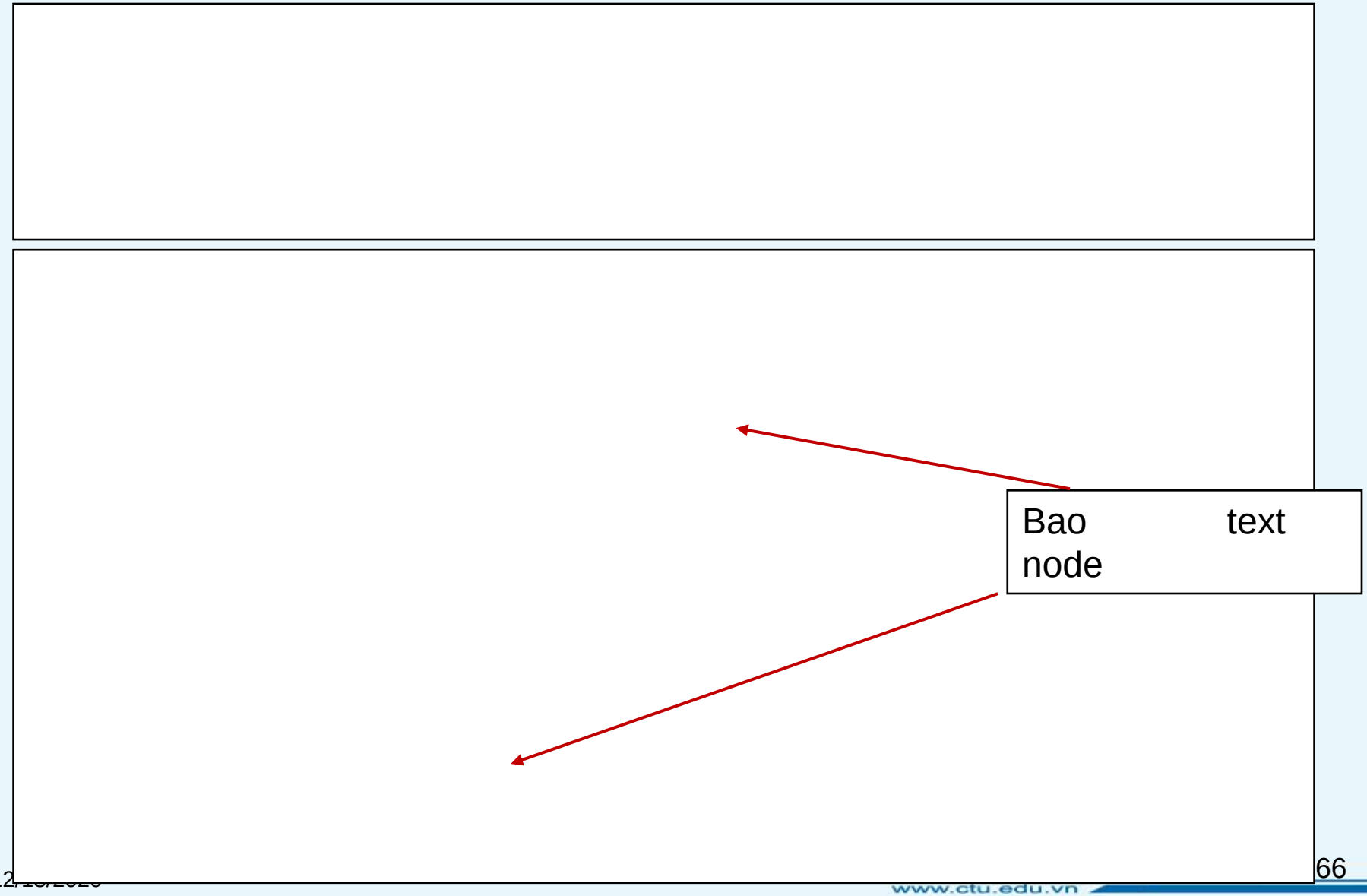
```
<div id="main">
	<p>This is an element node</p>
	This is a text node
	<h3>This is heading element</h3>
</div>
```

```
<script>
	var root = document.getElementById("main");

	var elements = root.children;
	for(let i = 0; i < elements.length; i++) {
		console.log(elements[i].textContent);
	}

// output
// This is an element node
// This is heading element
</script>
```

Ví 18: nodes con



Các và tính

ghi giá trên trang HTML

/ gán giá cho tính node

giá tính

gán val cho tính

/ thay giá

Các và tính

: dung , không
bao mã HTML, và dung có
tính CSS (ví display:none, hidden)

bao
, dung có tính CSS

toàn text, mã html và

Các và tính

Ví 19: khác nha các tính

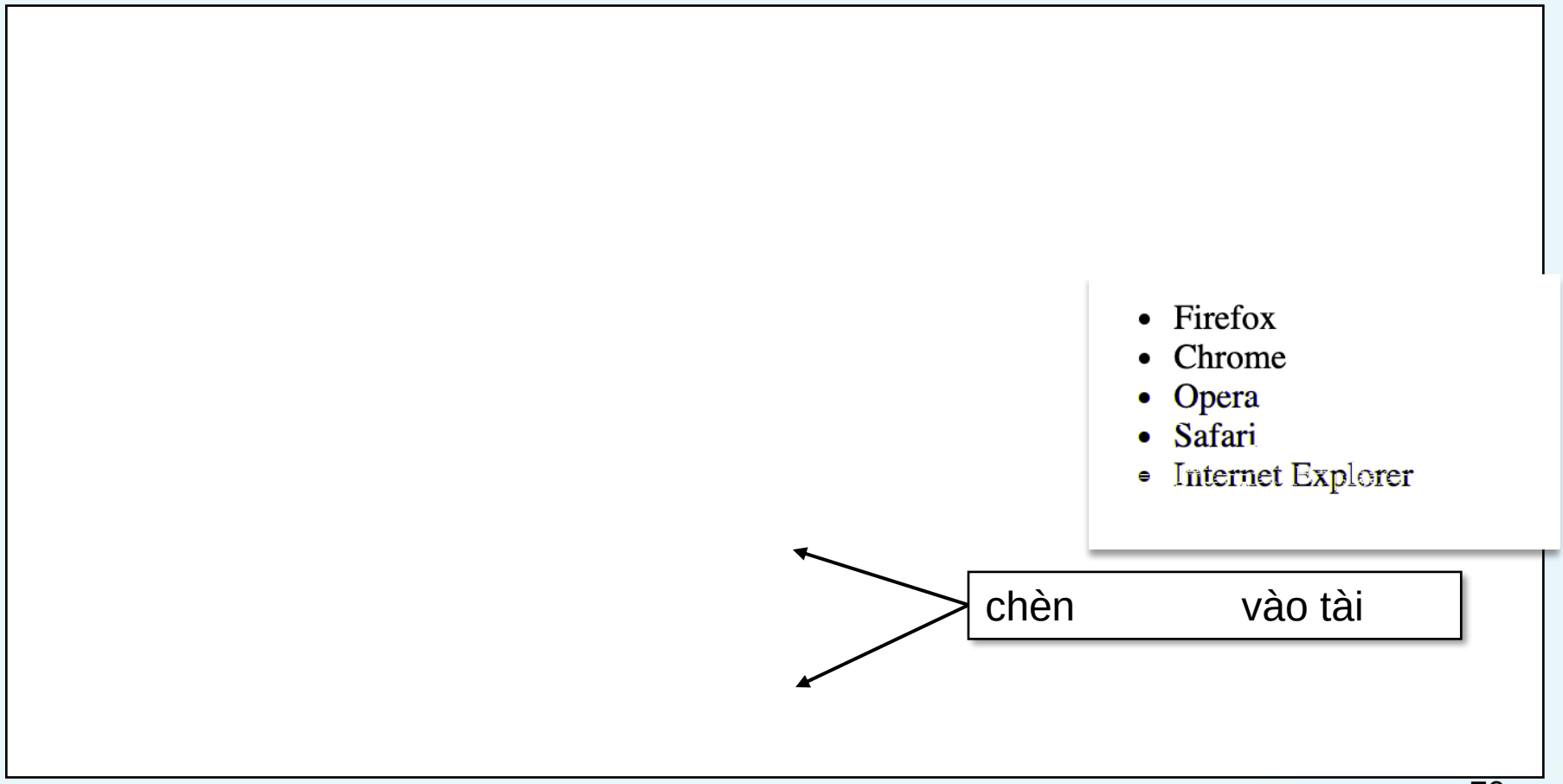
```
x.innerHTML = "New <b>content</b>";
x.style.color = "red";
```

Thêm, , xóa node

tài

, dùng làm container thêm các node con vào khi chèn vào trang HTML

Ví 20: chèn <li> và danh sách



Thêm, , xóa node

tên

(ví span, div, h1, table, td,)

cho element

gán cho

là ra

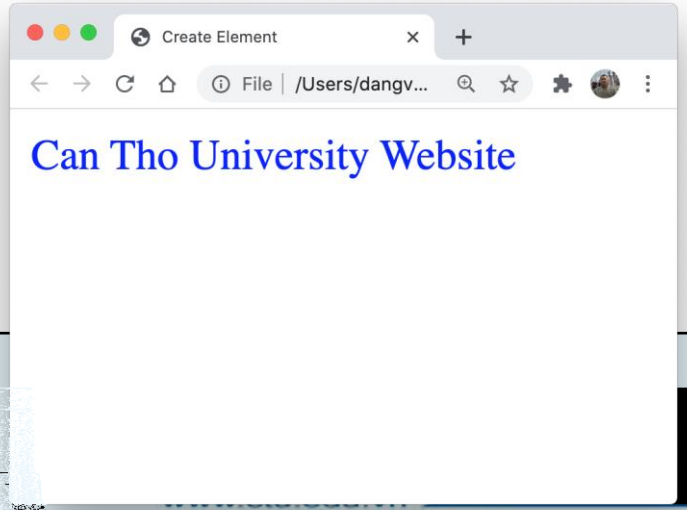
: nhân bao và

các con bên trong

: nhân và

Thêm, , xóa node

Ví 21: liên và chèn vào tài



Thêm, , xóa node

: thành con , n
trí cùng trong danh sách con
chèn là nút
con , ngay nút
xóa node document
xóa ra
, không
xóa tính
là , object xóa
xóa nút n là con ,
nút xóa xóa thành công, null nút con
không
thay nút con
, nút thay

Thêm, , xóa node

Ví 22: Chèn, xóa

- Java
- Python
- JavaScript