香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

THE DOM - BASIC CONCEPTS

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

- You'll appreciate the concept of a DOM
- You'll understand the role of text nodes
- You'll appreciate the concept of whitespace and how it is stored

THE DOM

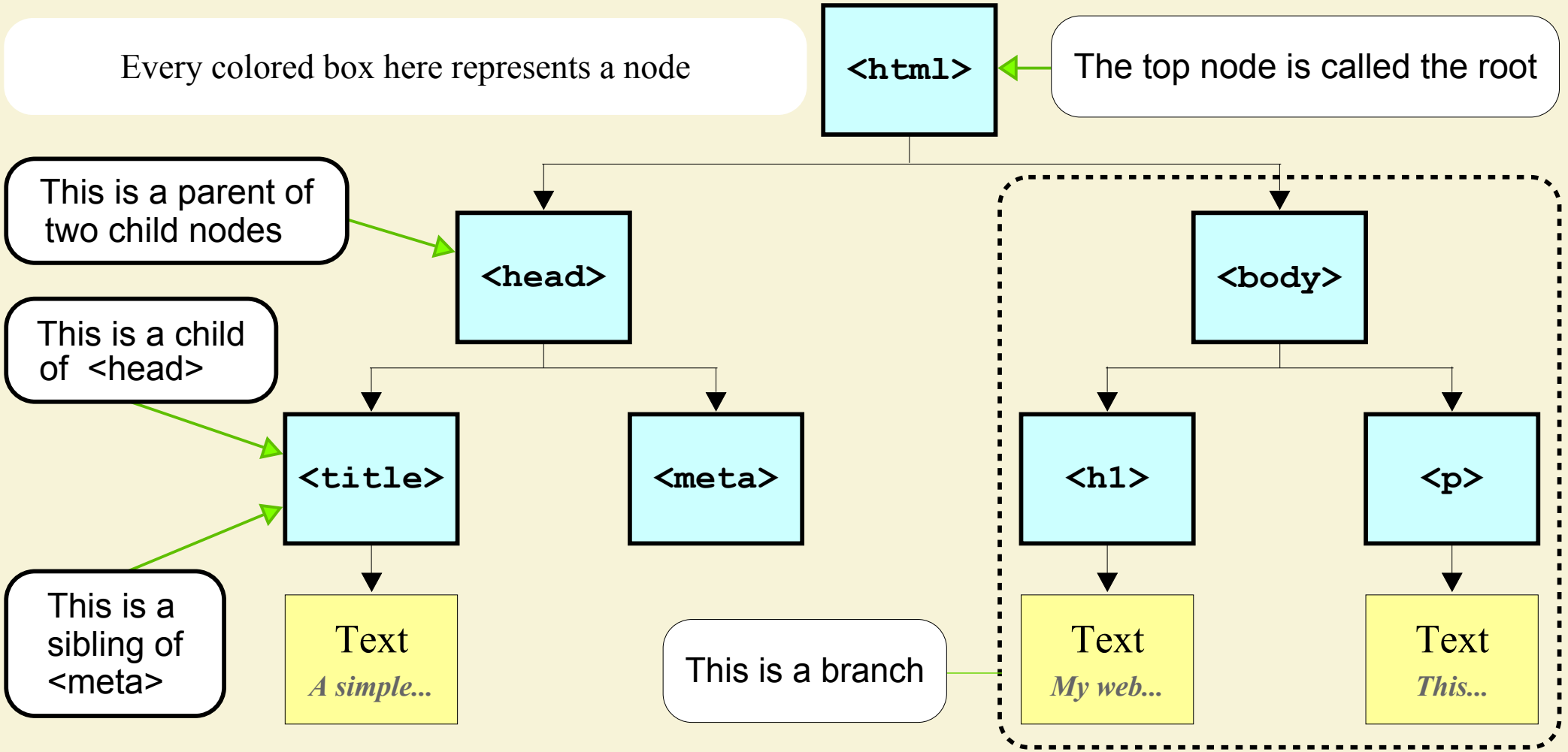
DOM means *Document Object Model*

When you load something into a browser,
it is converted into a DOM structure

CODE WE HAVE SEEN

```
<!DOCTYPE html>
<html>
<head>
    <title>A Simple Web Page</title>
    <meta name="author" content="David Rossiter">
</head>
<body>
    <h1>My Web Page</h1>
    <p>This web page is so awesome!</p>
</body>
</html>
```

WHAT THE PAGE LOOKS LIKE

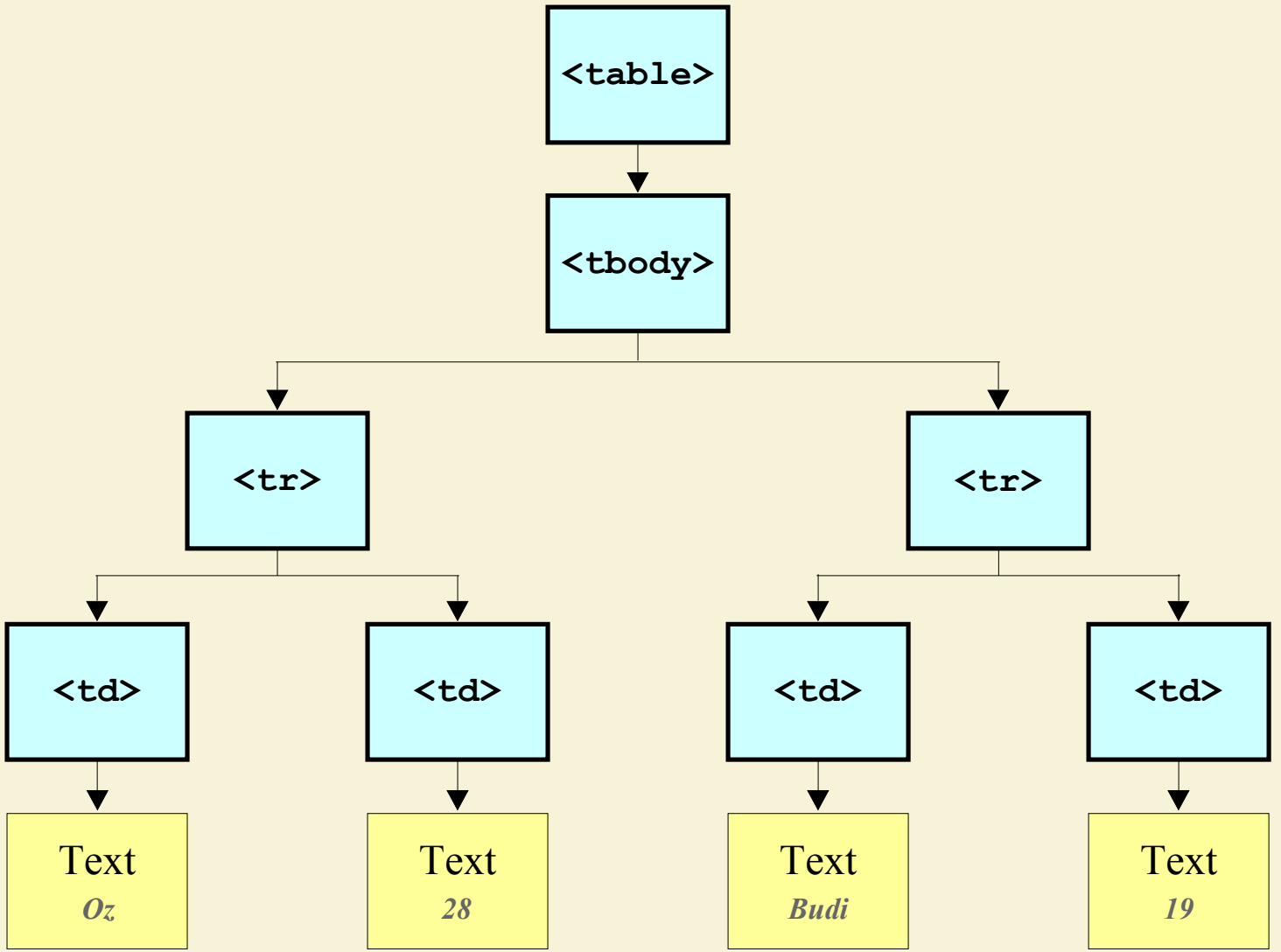

Every colored box here represents a node
<html>
The top node is called the root
This is a parent of two child nodes
<head>
<body>
This is a child of <head>
<title>
<meta>
<h1>
<p>
This is a sibling of <meta>
Text
A simple...
This is a branch
Text
My web...
Text
This...

```
<!DOCTYPE html>
<html>
<body>
<table>
<tbody>
<tr> <td>Oz</td> <td>28</td> </tr>
<tr> <td>Budi</td> <td>19</td> </tr>
</tbody>
</table>
</body>
</html>
```

WHAT THE PAGE LOOKS LIKE

Oz 28

Budi 19


<table>
<tbody>
<tr>
<tr>
<td>
<td>
<td>
<td>
Text
Oz
Text
28
Text
Budi
Text
19

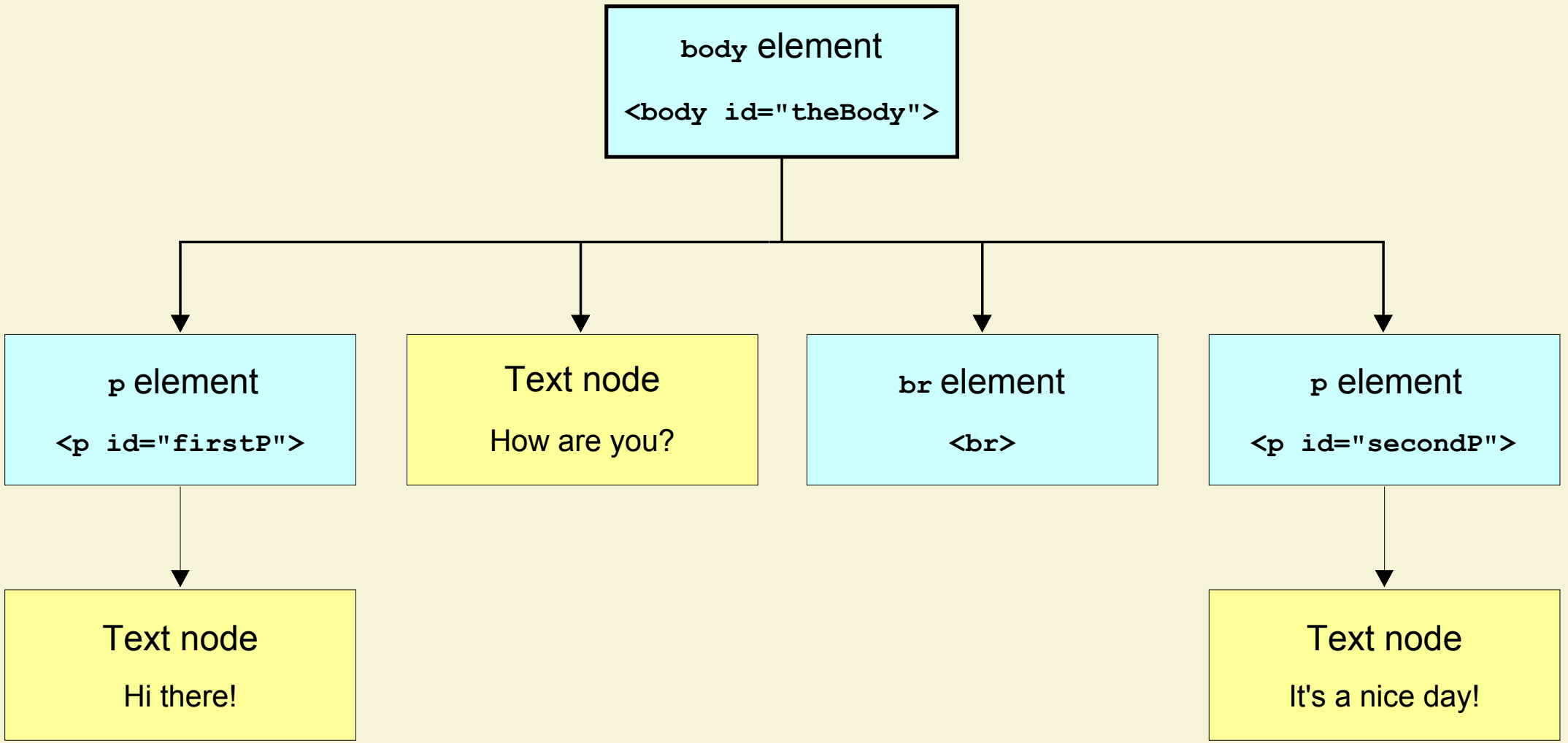
```
<!DOCTYPE html>
<html>
<body id="theBody"><p id="firstP">
Hi there!
</p>
How are you?
<br>
<p id="secondP">
It's a nice day!
</p>
</body>
</html>
```

WHAT THE PAGE LOOKS LIKE

Hi there!

How are you?

It's a nice day!


body element
<body id="theBody">
p element
<p id="firstP">
Text node
How are you?
br element

p element
<p id="secondP">
Text node
Hi there!
Text node
It's a nice day!

WHITESPACE NODES

- Whitespace is anything you can't see i.e. spacing
- There may be a text node which contains only whitespace
- These is called a 'whitespace node'
- These are sometimes troublesome

A COMPARISON

```
<body><p>Hello.</p>
```

does not have a whitespace node between <body> and <p>

```
<body>
<p>Hello.</p>
```

does have a whitespace node between <body> and <p>

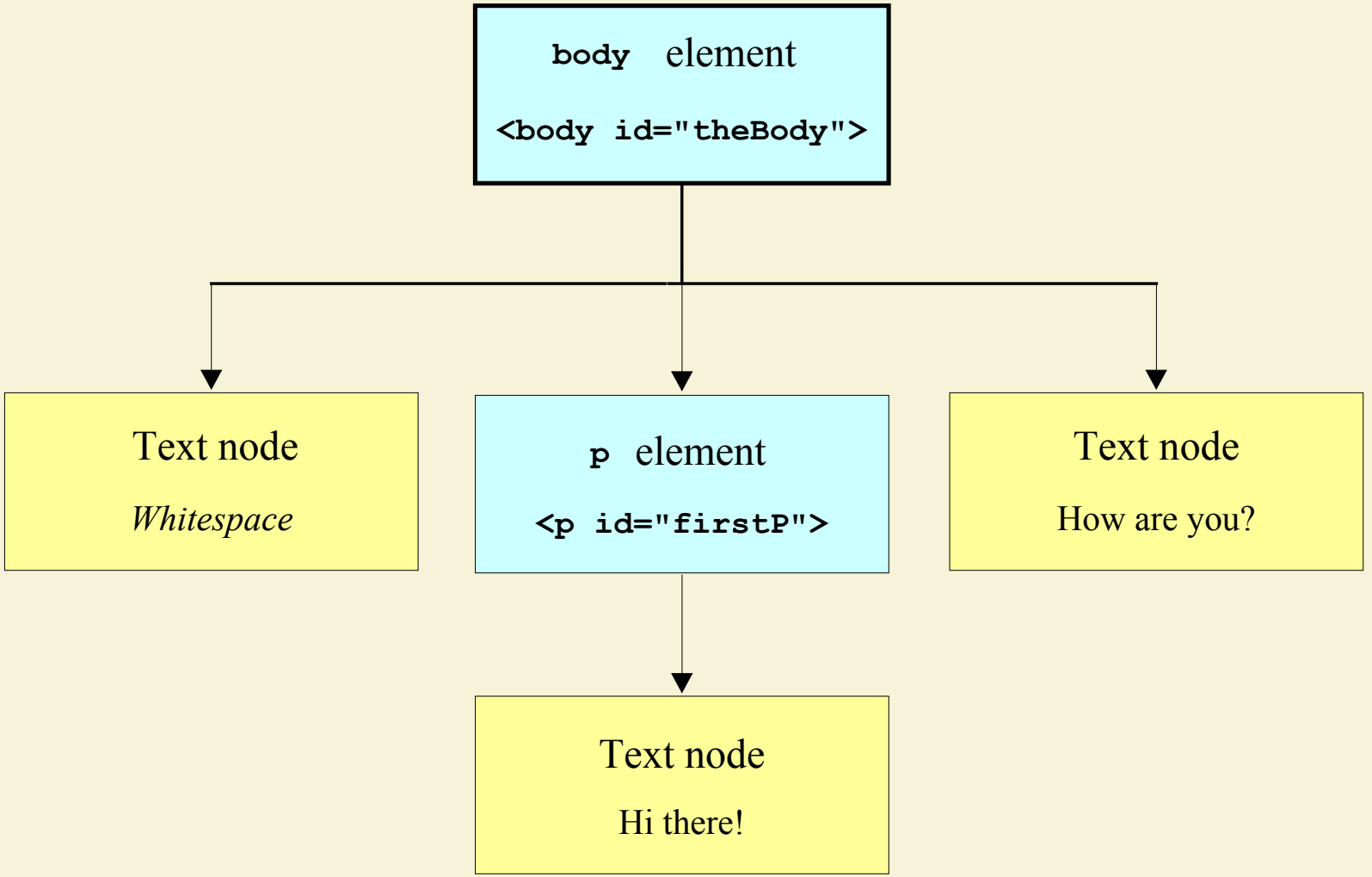
EXAMPLE OF A WHITESPACE NODE

```
<!DOCTYPE html>
<html>
<body id="theBody">
<p id="firstP">
Hi there!
</p>
How are you?
</body></html>
```

WHAT THE PAGE LOOKS LIKE

Hi there!

How are you?


body element
<body id="theBody">
Text node
Whitespace
p element
<p id="firstP">
Text node
How are you?
Text node
Hi there!

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

NODE RELATIONSHIPS

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

- You'll understand the relationship between nodes
- You'll be able to visualize the path for a node
- You'll appreciate the use of event handlers

WE WILL LOOK AT

Handling the parent	`parentNode`
Handling child nodes	`childNodes[]`, `firstChild`, `lastChild`
Handling siblings	`previousSibling`, `nextSibling`

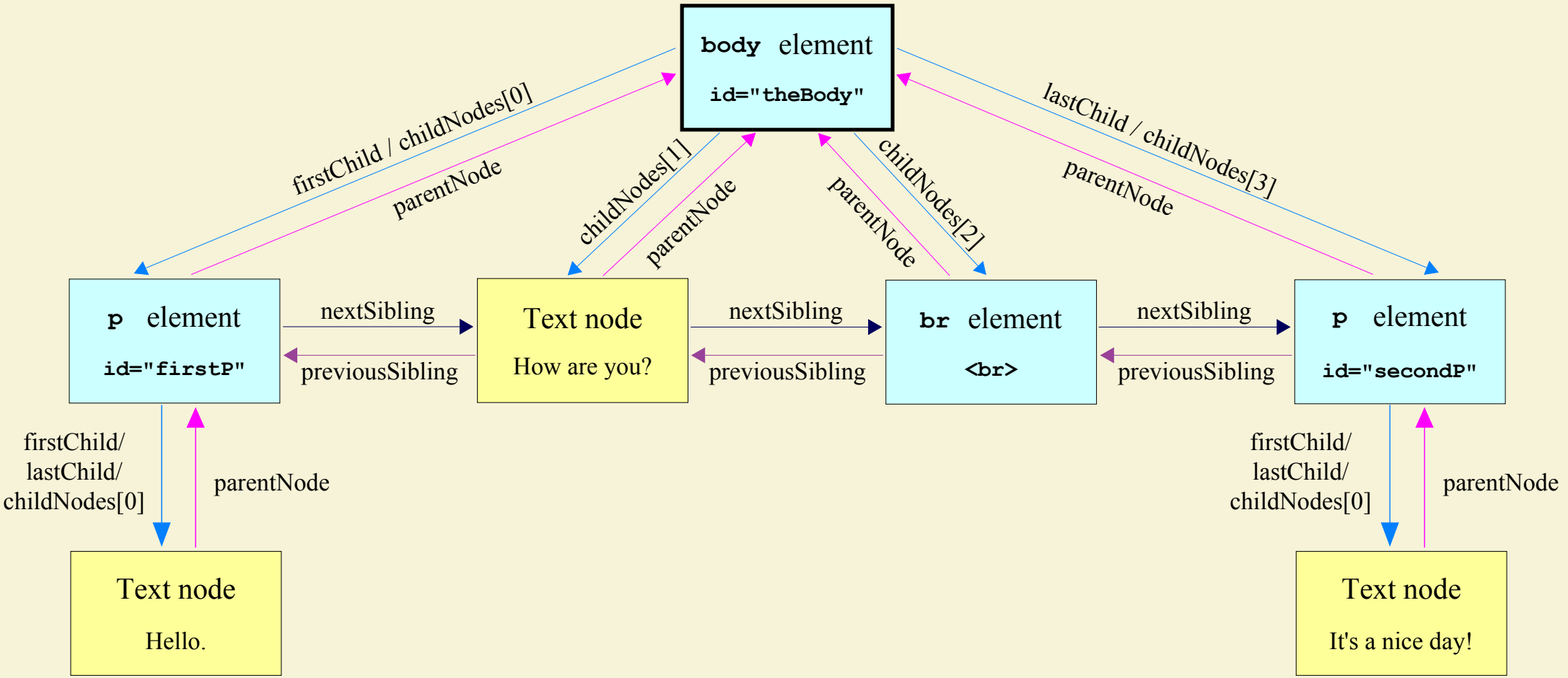
NODE RELATIONSHIPS

How is one node related to another?

Specific code lets you use such relationship

NODE RELATIONSHIPS

- Your code can access all of these:
  - `parentNode`
  - `childNodes[]`, `firstChild`, `lastChild`
  - `previousSibling`, `nextSibling`


body element
id="theBody"
firstChild / childNodes[0]
parentNode
childNodes[1]
parentNode
parentNode
childNodes[2]
lastChild / childNodes[3]
parentNode
p element
id="firstP"
nextSibling
previousSibling
Text node
How are you?
nextSibling
previousSibling
br element

nextSibling
previousSibling
p element
id="secondP"
firstChild/
lastChild/
childNodes[0]
parentNode
firstChild/
lastChild/
childNodes[0]
parentNode
Text node
Hello.
Text node
It's a nice day!

HOW TO FIND THE PATH?

In the next example we show the path to a node

1. The function starts with a node
2. The type of the node is added to a string
3. The code moves to the parent of the node
4. If the node has a parent, repeat (2) and (3)

```
function handleClick(event) {
  event.stopPropagation();

  var node = event.target
  var thisPath = node.nodeName;

  while (node.parentNode) {
    node = node.parentNode;
    thisPath = node.nodeName + " > " + thisPath;
  }

  alert(thisPath);
}
```

控制此代码只执行一次，防止其在这里执行完后仍被其他事件捕捉

HOW TO TRIGGER THE CODE?

To trigger the code, the user clicks on a node

To enable this, event handlers are added to the nodes

Two examples follow: HTML and SVG

They use the same code

Event handlers are added to every element

将此函数注册到所有的node

```
// Register the click event handler for all nodes
function attachHandler(node) {
  if(node == null) return;
  node.onclick = handleClick;

  for (var i = 0; i < node.childNodes.length; ++i)
    attachHandler(node.childNodes[i]);
}
```

Click on the link or form elements to see the DOM path to that node

(The html that you see below is just some 'random' html which helps to demonstrate the technique).

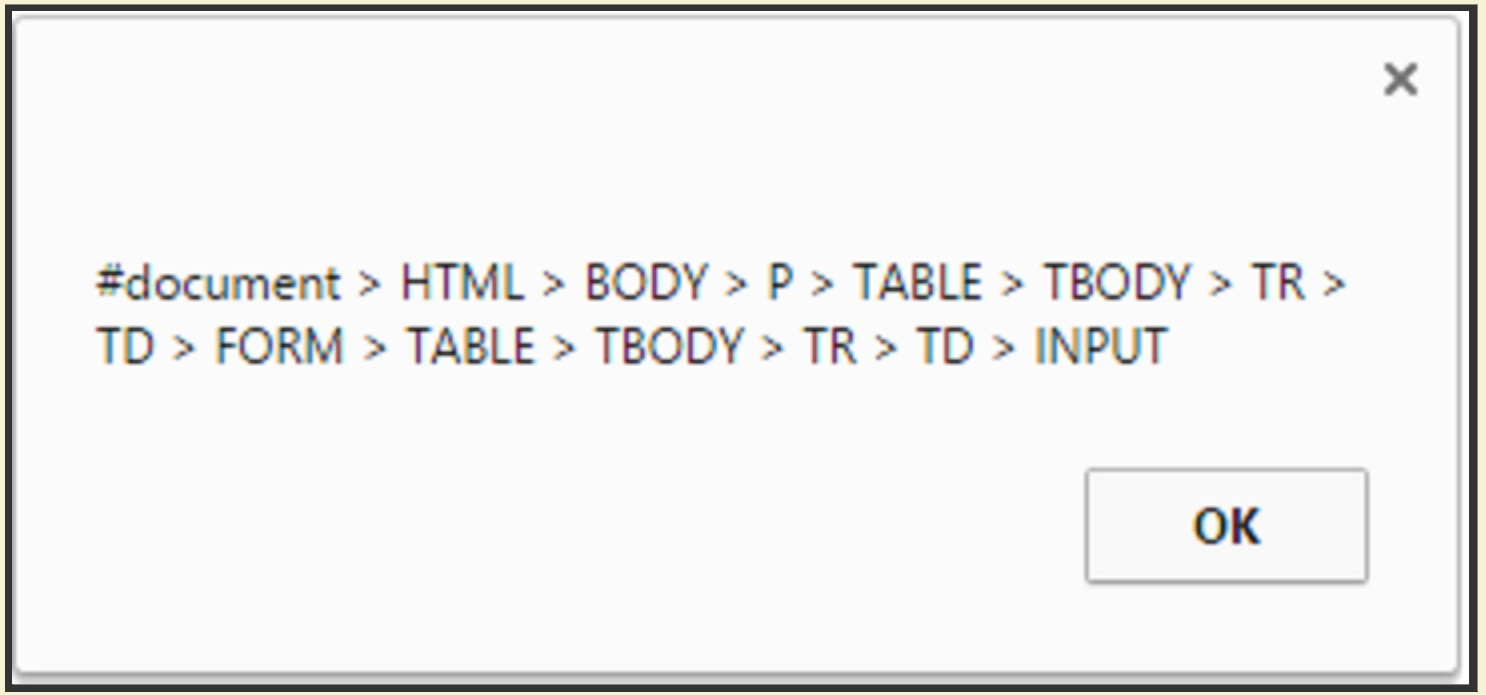
LISTS

- List 1
- List 2
- List 3
- List 4
    1. Order List 1
    2. Order List 2
    3. Order List 3
    4. Order List 4

TABLES

Cell 1	Cell 2	Cell 3	Cell 4
		Click!	

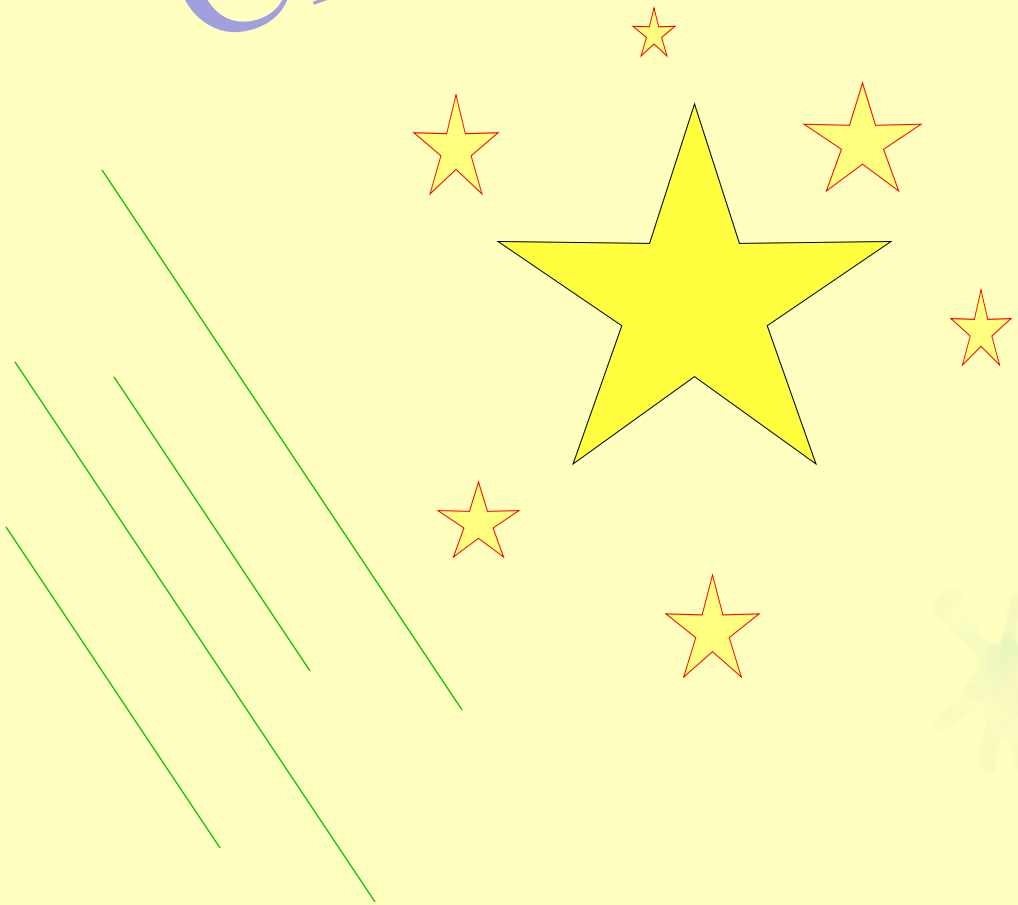
HTML RESULT

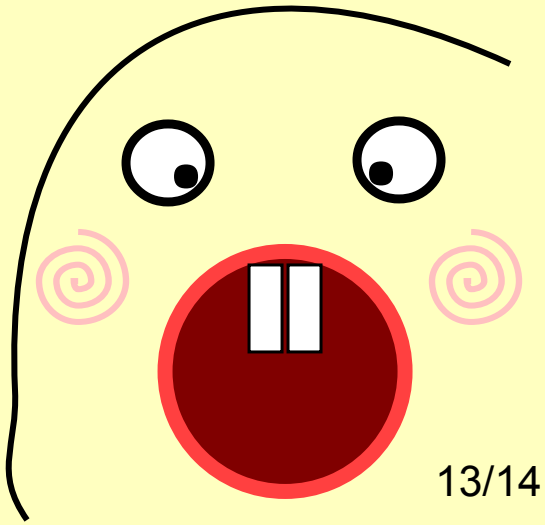

#document > HTML > BODY > P > TABLE > TBODY > TR >
TD > FORM > TABLE > TBODY > TR > TD > INPUT
OK

Click on any object to see the DOM path to that node

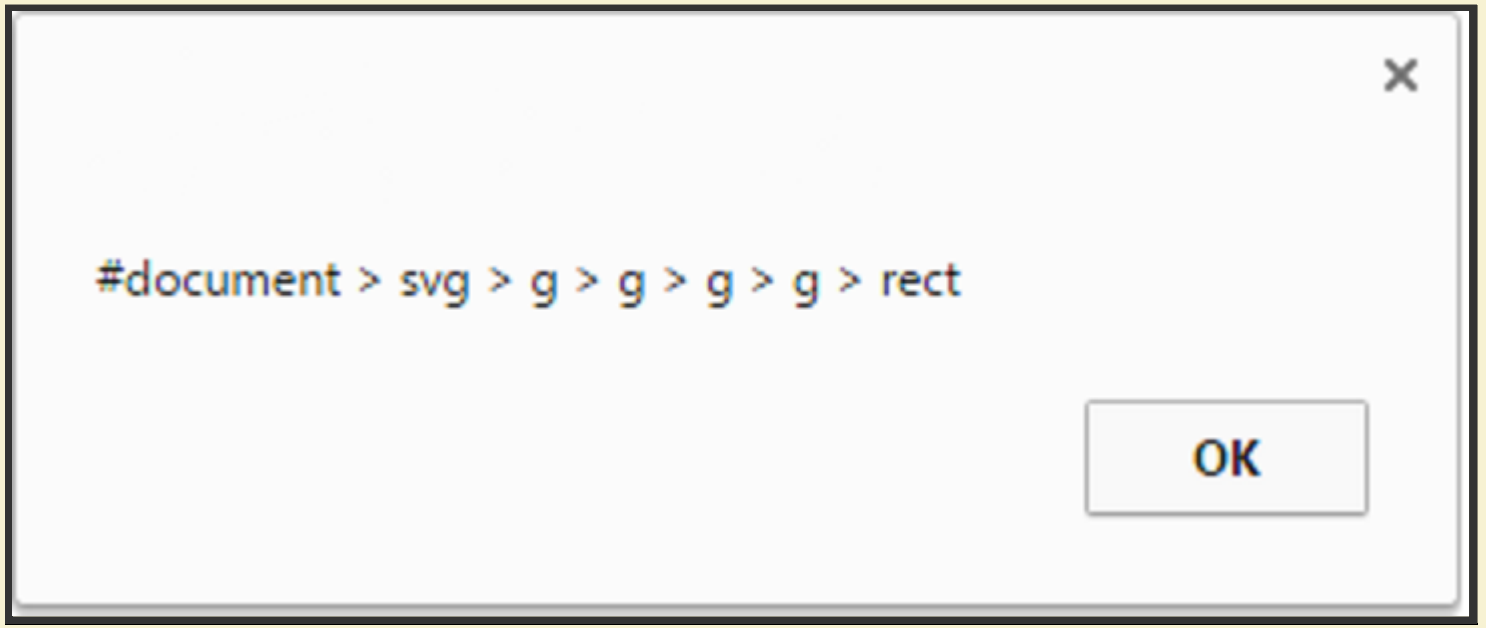
Click Me!

Please click ME!





SVG RESULT



香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

LOCATING NODES

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

- You'll appreciate how to find a node in the DOM
- You'll appreciate how to set the attribute of a node

WE WILL LOOK AT

`getElementsByTagName()`

`getElementById()`

`setAttribute()`

THE DOM

Everything is in the DOM

We can add/delete/copy/change anything

To do this, we need to understand how to access things

USING THE EXACT PATH

- Method 1: Use the exact DOM path
  - Sometimes hard to work out the exact path
  - Easy to make mistakes
  - In another browser the DOM may be slightly different - code fails!

USING THE TYPE

- Method 2: Use `getElementsByTagName()`
  - This requires you to know the exact tag name
    e.g. is it `h2` or `h3`?
  - There might be several nodes of that type,
    so you have to know exactly which one it is

need to know which one you refer to, eg, fisrt one, second one…

USING THE NAME

- Method 3: Use `getElementById()`
  - If you give a node a unique name e.g.

    `<element_name id="thing"> . . . </element_name>`

    then this method is the easiest way

```
<body>
<h2 style="color:black" id="cute_text">
Click on a button to change the colour
</h2>

<form>
<input onclick="change_color1()" type="button"
    value="Change using method 1">
<input onclick="change_color2()" type="button"
    value="Change using method 2">
<input onclick="change_color3()" type="button"
    value="Change using method 3">
</form>
</body>

</html>
```

EXAMPLE - CODE

```
<!DOCTYPE html>
<html> <head>
<script>
function change_color1(){
  document.childNodes[0].childNodes[2].childNodes[1]
    .style.color="red";
}
function change_color2(){
  document.getElementsByTagName("h2")[0].style.color
    ="yellow";
}
function change_color3(){
  document.getElementById("cute_text").style.color="blue";
}
</script>
</head>
```

method 2: 选择第1个h2

Click on a button to change the colour of this text

Change using method 1 Change using method 2

Change using method 3

Click here to open the example

SETATTRIBUTE()

This is a common way to change something

For example:

```
the_node=getElementById("thisNode");
the_node.setAttribute("style", "color:red");
```

```
<!DOCTYPE html>
<html> <head>
<script>
function change_color1(){
  document.childNodes[0].childNodes[2].childNodes[1]
    .setAttribute("style", "color:red");
}
function change_color2(){
  document.getElementsByTagName("h2")[0]
    .setAttribute("style", "color:yellow");
}
function change_color3(){
  document.getElementById("cute_text")
    .setAttribute("style", "color:blue");
}
</script>
</head>
```

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

CREATING AND ADDING NODES

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

- You'll be able to create different types of nodes
- You'll be able to add nodes to a DOM

WE WILL LOOK AT

Creating a node	`createElement()`, `createTextNode()`
Adding a node	`insertBefore()`, `appendChild()`

ADDING TO THE WEB PAGE

First, create whatever you want to add

- whatever you create is not yet in the DOM

Second, add it at the desired place

CREATING NODES

Use `createElement()` e.g.

```
result = document.createElement("div");
```

For text nodes, use `createTextNode()` e.g.

```
result = document.createTextNode("Hello");
```

ADDING NODES - INSERTBEFORE()

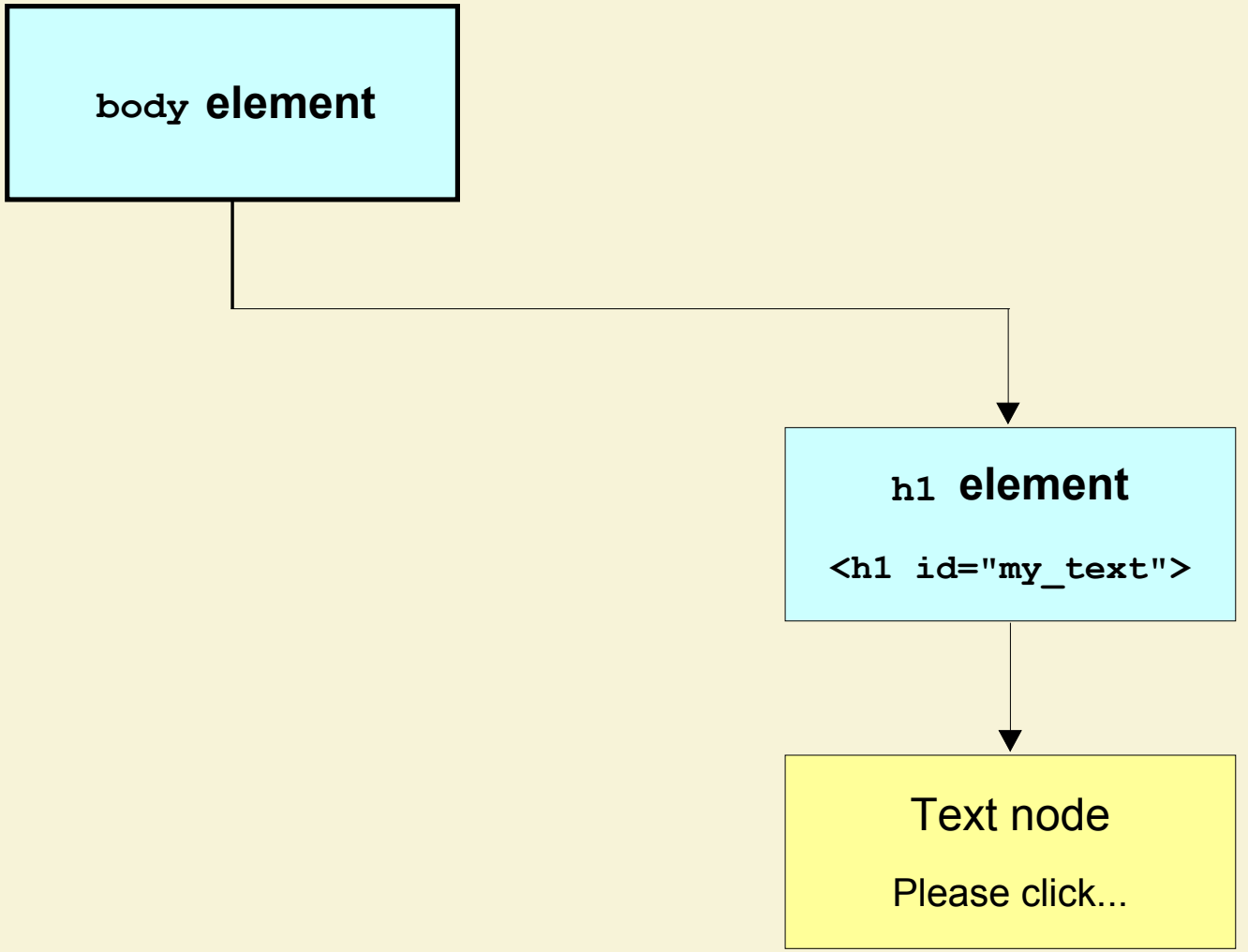
For example:

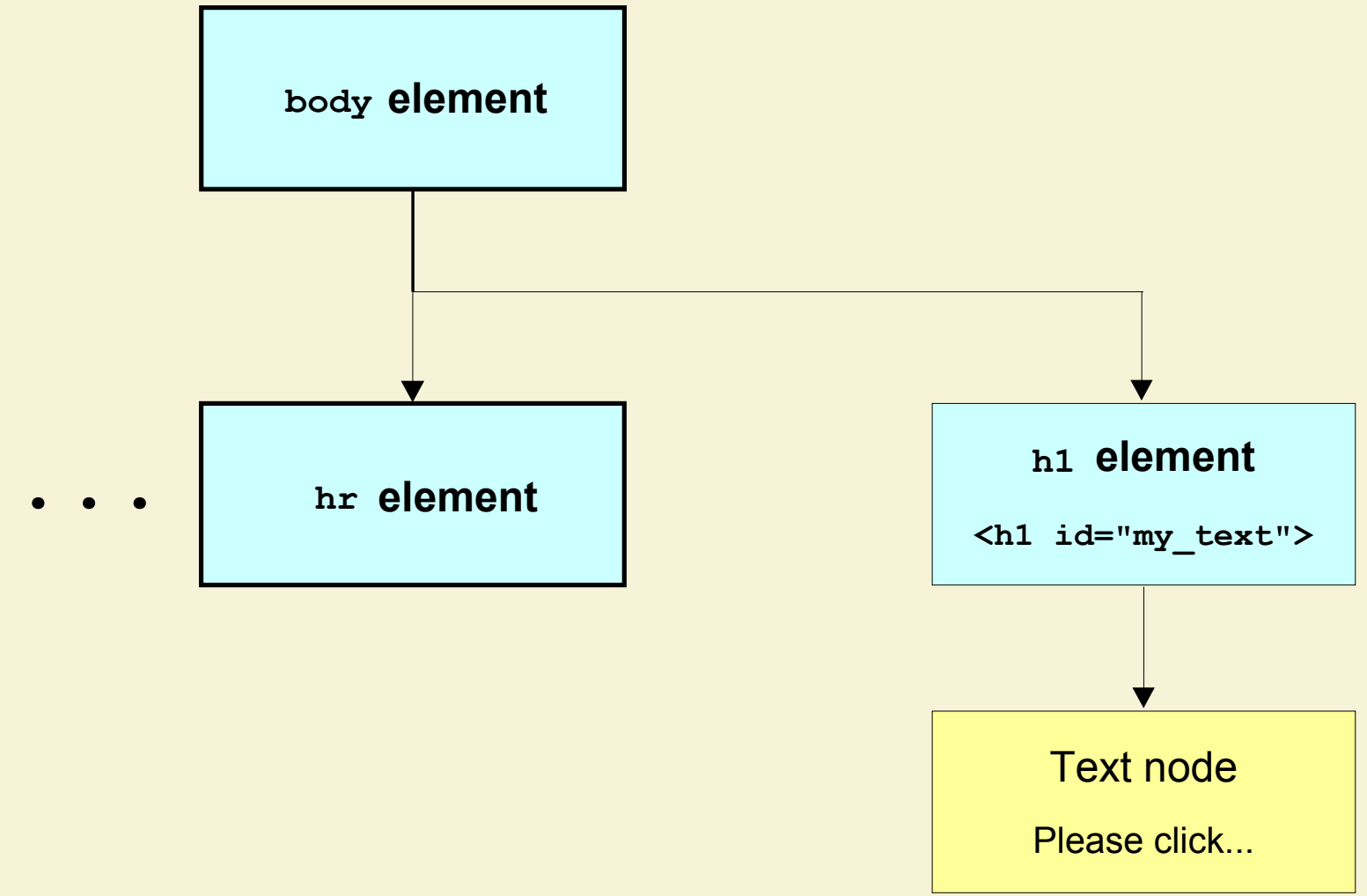
```
newItem = document.createElement("hr");

destParent = document
             .getElementsByTagName("body")[0];
destParent.insertBefore(
             newItem, destParent.firstChild);
```

Please click on the page

Click here to open the example


body element
h1 element
<h1 id="my_text">
Text node
Please click...


body element
hr element
h1 element
<h1 id="my_text">
Text node
Please click...

```
<!DOCTYPE html>
<html>
<head>
<script>
function insert_new_text() {
  var newItem=document.createElement("hr");

  var destParent=document.getElementsByTagName("body")[0];
  destParent.insertBefore(newItem, destParent.firstChild);
}
</script>
</head>
<body onclick="insert_new_text()">
<h1 id="my_text">Please click on the page</h1>
</body>
</html>
```

ADDING NODES - APPENDCHILD()

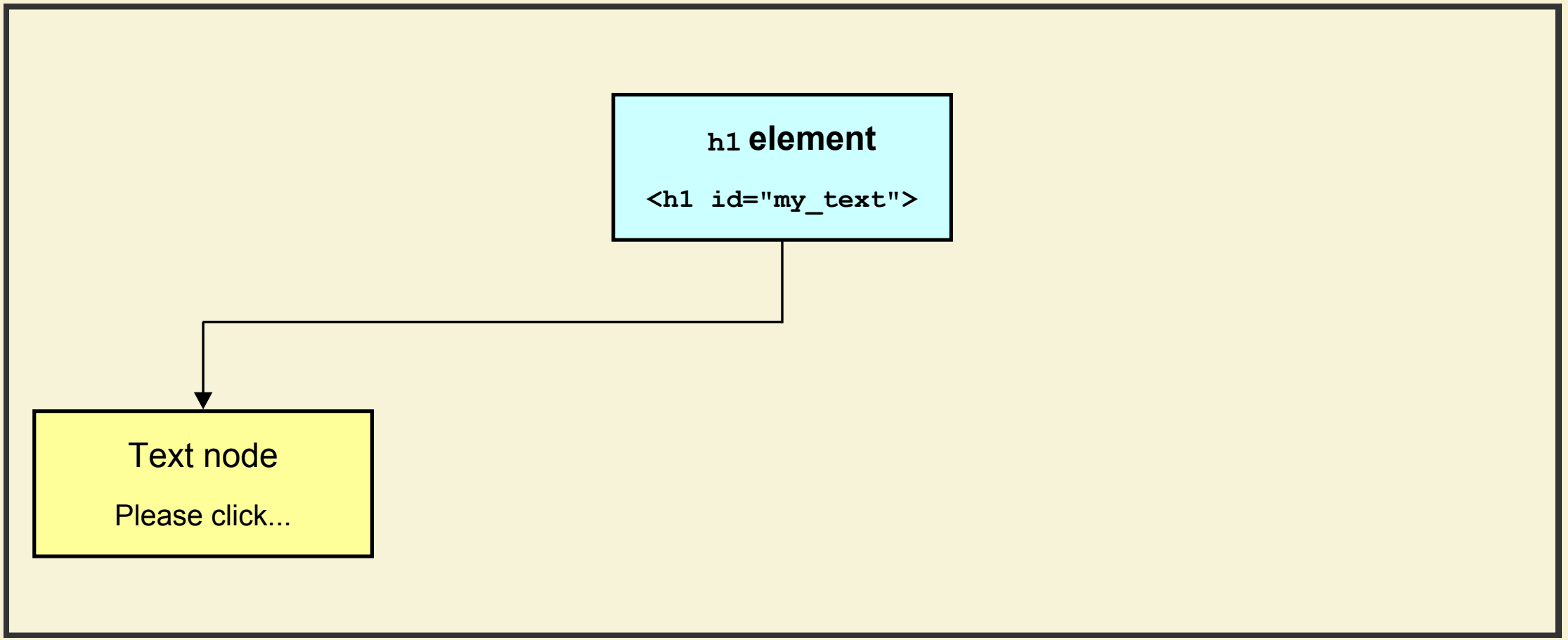
For example:

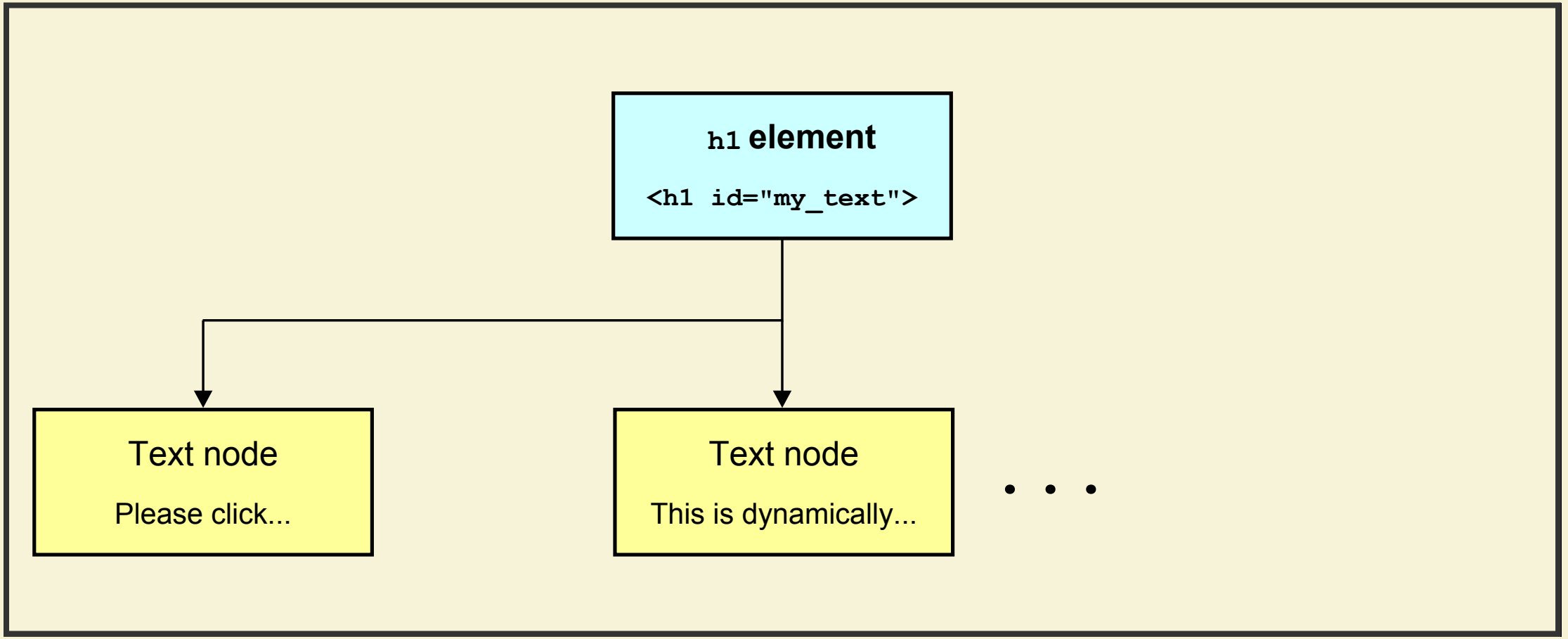
```
result=document.createTextNode(
            "This is dynamically added Text!");

document.getElementById("my_text")
        .appendChild(result);
```

Please click on the page

Click here to open the example


h1 element
<h1 id="my_text">
Text node
Please click...


h1 element
<h1 id="my_text">
Text node
Please click...
Text node
This is dynamically...
• • •

```
<!DOCTYPE html>
<html>
<head>
<script>
function insert_new_text()
{
    var newText = document.createTextNode(
        "This is dynamically added text!");
    var textpart = document.getElementById("my_text");
    textpart.appendChild(newText);
}
</script>
</head>
<body onclick="insert_new_text()">
<h1 id="my_text">Please click on the page</h1>
</body> </html>
```

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

DELETING NODES

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

- You'll be able to delete node from a DOM

WE WILL LOOK AT

Removing a node `removeChild()`

DELETING A NODE

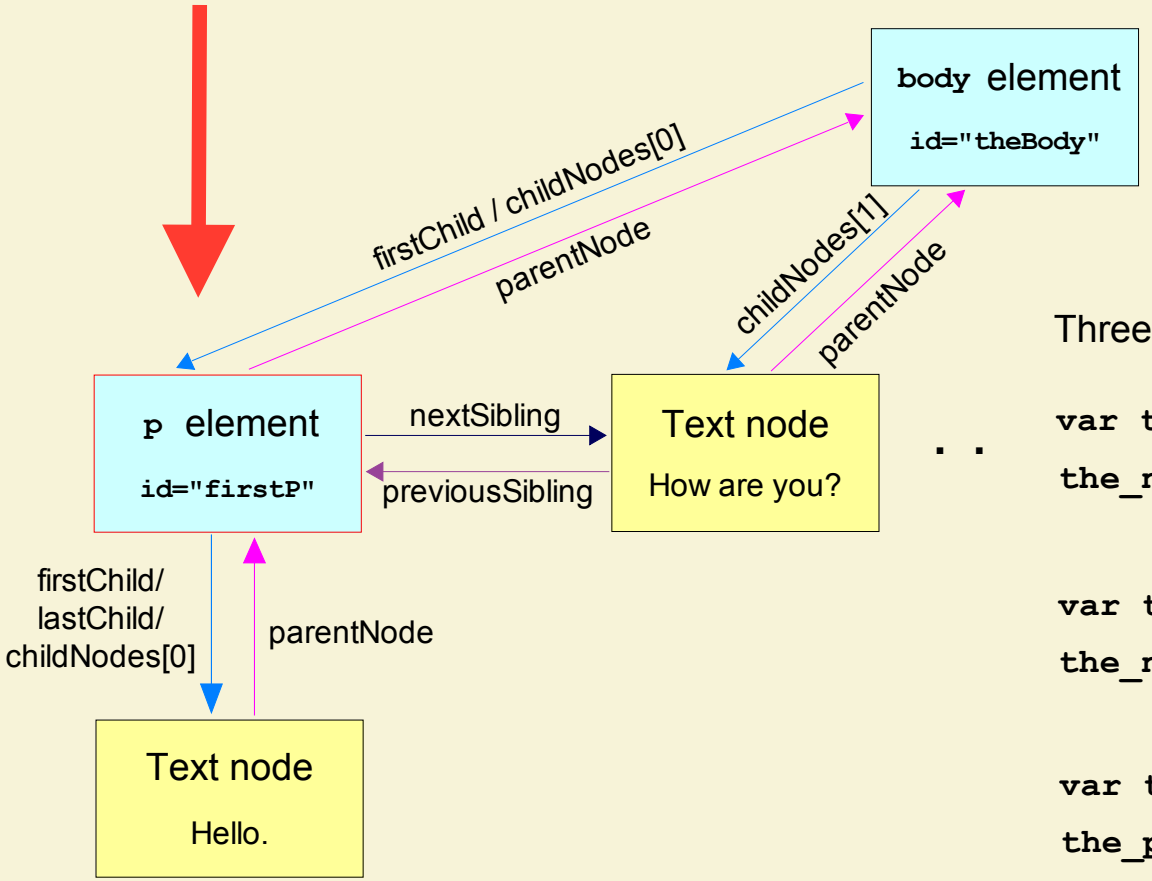
Tell the parent of the node to delete it e.g.

```
this_node=getElementById("myPara");
this_node.parentNode.removeChild(this_node);
```

(1). find the node wanted to delete

(2). find its parents and delete it

要删除的node



Three example code to delete the first paragraph in this DOM:

```
var the_node=document.getElementById("firstP");
the_node.parentNode.removeChild(the_node);
```

or

```
var the_node=document.getElementsByTagName("p")[0];
the_node.parentNode.removeChild(the_node);
```

or

```
var the_parent=document.getElementById("theBody");
the_parent.removeChild(the_parent.firstChild);
```

THREE EXAMPLE CODE

```
var the_node=document.getElementById("firstP");
the_node.parentNode.removeChild(the_node);
```

```
var the_node=document.getElementsByTagName("p")[0];
the_node.parentNode.removeChild(the_node);
```

```
var the_parent=document.getElementById("theBody");
the_parent.removeChild(the_parent.firstChild);
```

Hello.

How are you?

It's a nice day!

Example 1 Example 2 Example 3

Reload the page to reset the DOM.

Click here to open the example

```
<body id="theBody"><p id="firstP">
Hello.
</p>
How are you?
<br>
<p id="secondP">
It's a nice day!
</p>
<button type="button" onclick="delete1()">Example 1</button>
<button type="button" onclick="delete2()">Example 2</button>
<button type="button" onclick="delete3()">Example 3</button>
<p>
Reload the page to reset the DOM.
</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head> <script>
    function delete1() {
        var the_node=document.getElementById("firstP");
        the_node.parentNode.removeChild(the_node);
    }
    function delete2() {
        var the_node=document.getElementsByTagName("p")[0];
        the_node.parentNode.removeChild(the_node);
    }
    function delete3() {
        var the_parent=document.getElementById("theBody");
        the_parent.removeChild(the_parent.firstChild);
    }
</script> </head>
```

DELETING ALL CHILDREN

Sometimes you want to delete everything under a node

For example, deleting all web page content

One way to do that is to delete every child

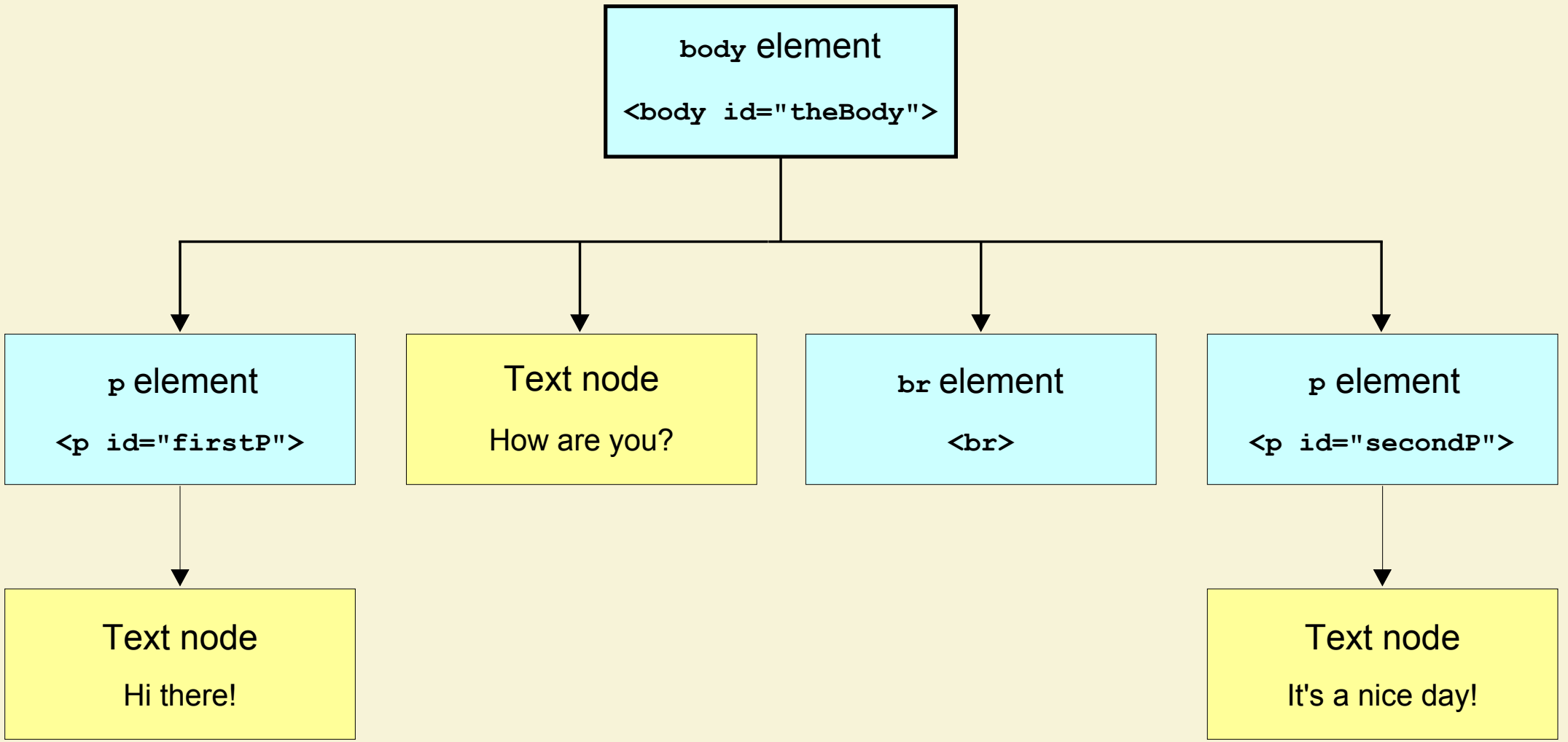
Hello.

How are you?

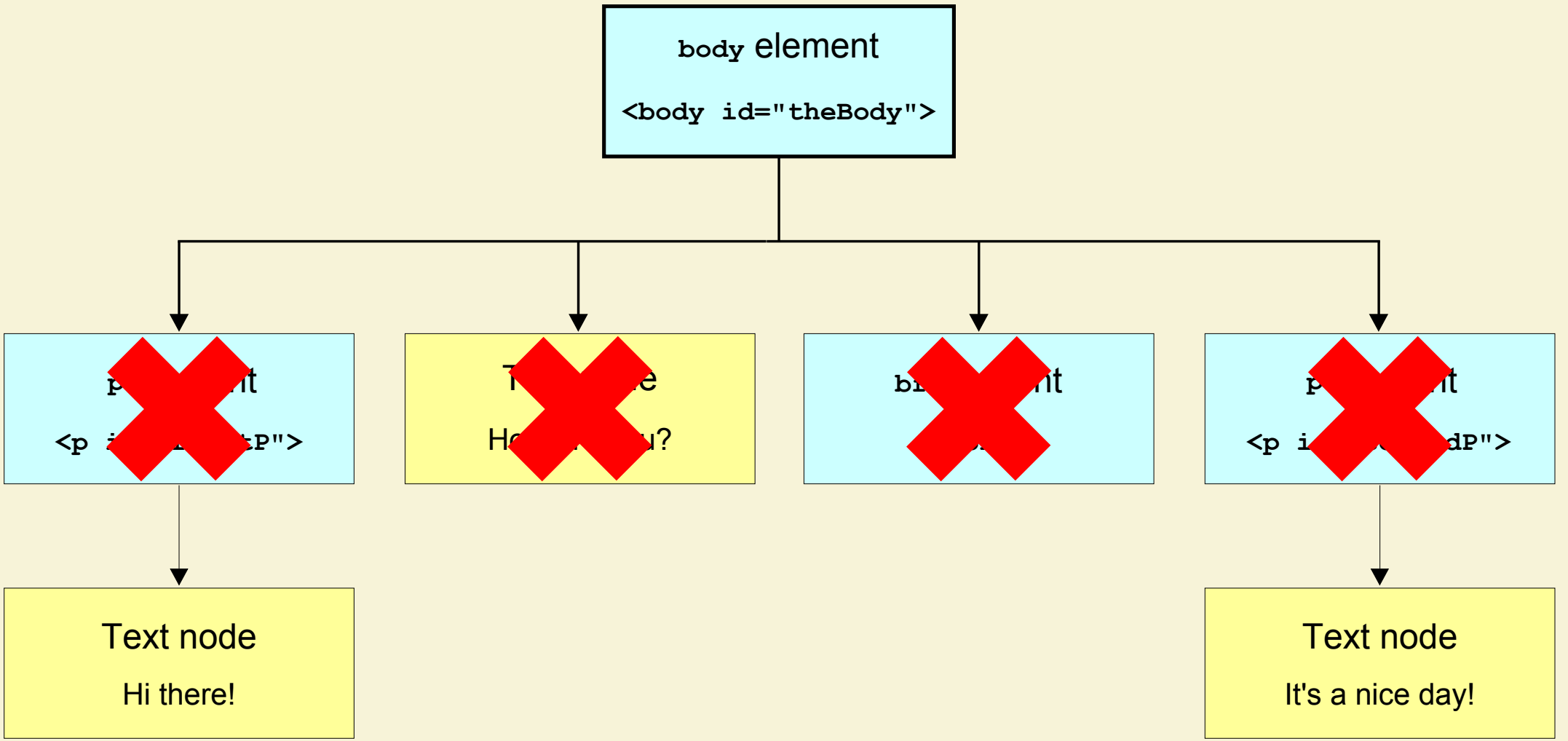
It's a nice day!

Delete children

Click here to open the example


body element
<body id="theBody">
p element
<p id="firstP">
Text node
How are you?
br element

p element
<p id="secondP">
Text node
Hi there!
Text node
It's a nice day!


body element
<body id="theBody">
Text node
Hi there!
Text node
It's a nice day!

`body` element

```
<body id="theBody">
```

DELETING ALL CHILDREN

```
var theNode = document.getElementById("theBody");

while (theNode.firstChild) // While there is a child
    theNode.removeChild(theNode.firstChild);
```

```
<!DOCTYPE html>
<html> <head> <script>
function delete_all_children() {
    var theNode = document.getElementById("theBody");
    while (theNode.firstChild)
        theNode.removeChild(theNode.firstChild);
}
</script> </head>
<body id="theBody">
<p id="firstP">Hello.</p>
How are you?
<br>
<p id="secondP">It's a nice day!</p>
<button type="button"
    onclick="delete_all_children()">Delete children</button>
</body> </html>
```

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

CLONING NODES

PROF. DAVID ROSSITER

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

AFTER THIS PRESENTATION

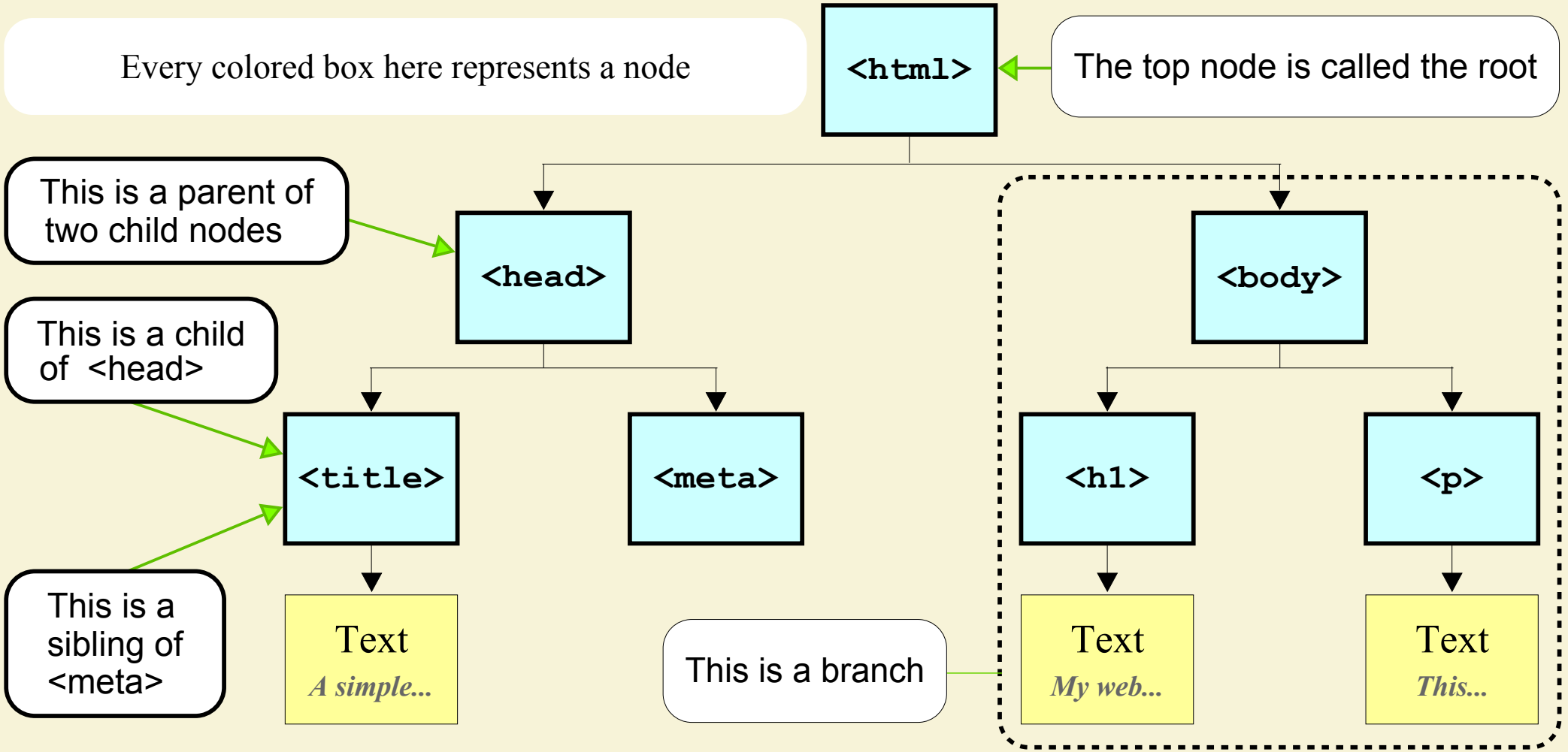
- You'll be able to copy (clone) a node
- You'll be able to copy a branch of nodes

WE WILL LOOK AT

Copying a node	`the_node.cloneNode()`
Copying a branch	`the_node.cloneNode(true)`
Adding node(s)	`dest.appendChild(the_node)`

THE BASIC IDEA

1. Copy node(s) from the DOM
2. Paste the copied node(s) in the DOM


Every colored box here represents a node
<html>
The top node is called the root
This is a parent of two child nodes
<head>
<body>
This is a child of <head>
<title>
<meta>
<h1>
<p>
This is a sibling of <meta>
Text
A simple...
This is a branch
Text
My web...
Text
This...

CLONING A NODE

Use *node*`.cloneNode()`

It's the same as *node*`.cloneNode(false)`

just copy the node without its children

EXAMPLE

1. A list item node `<li>` is copied
2. The copy is then added to the end of the list

- Good morning
- Hello

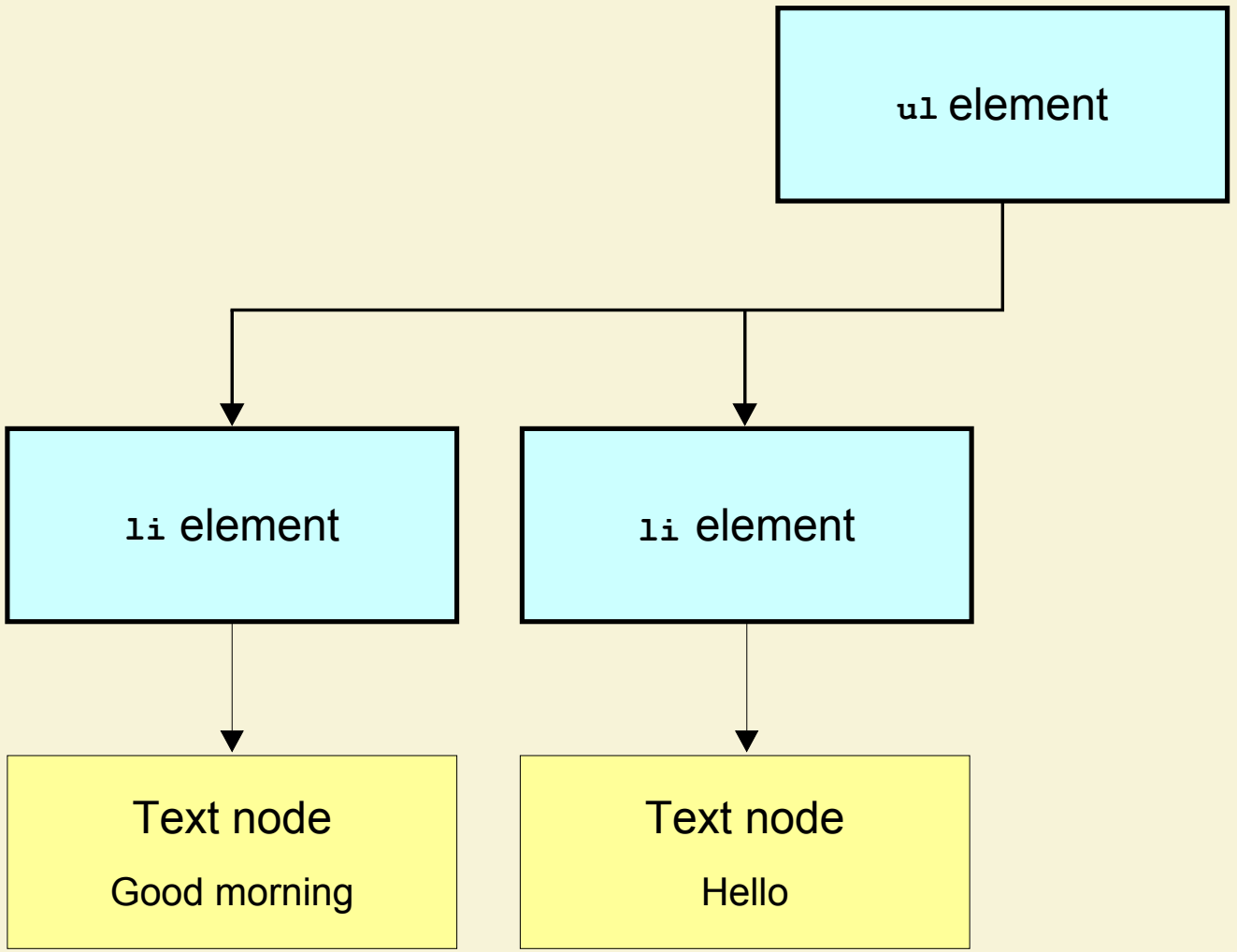
Click on the button to cloneNode()

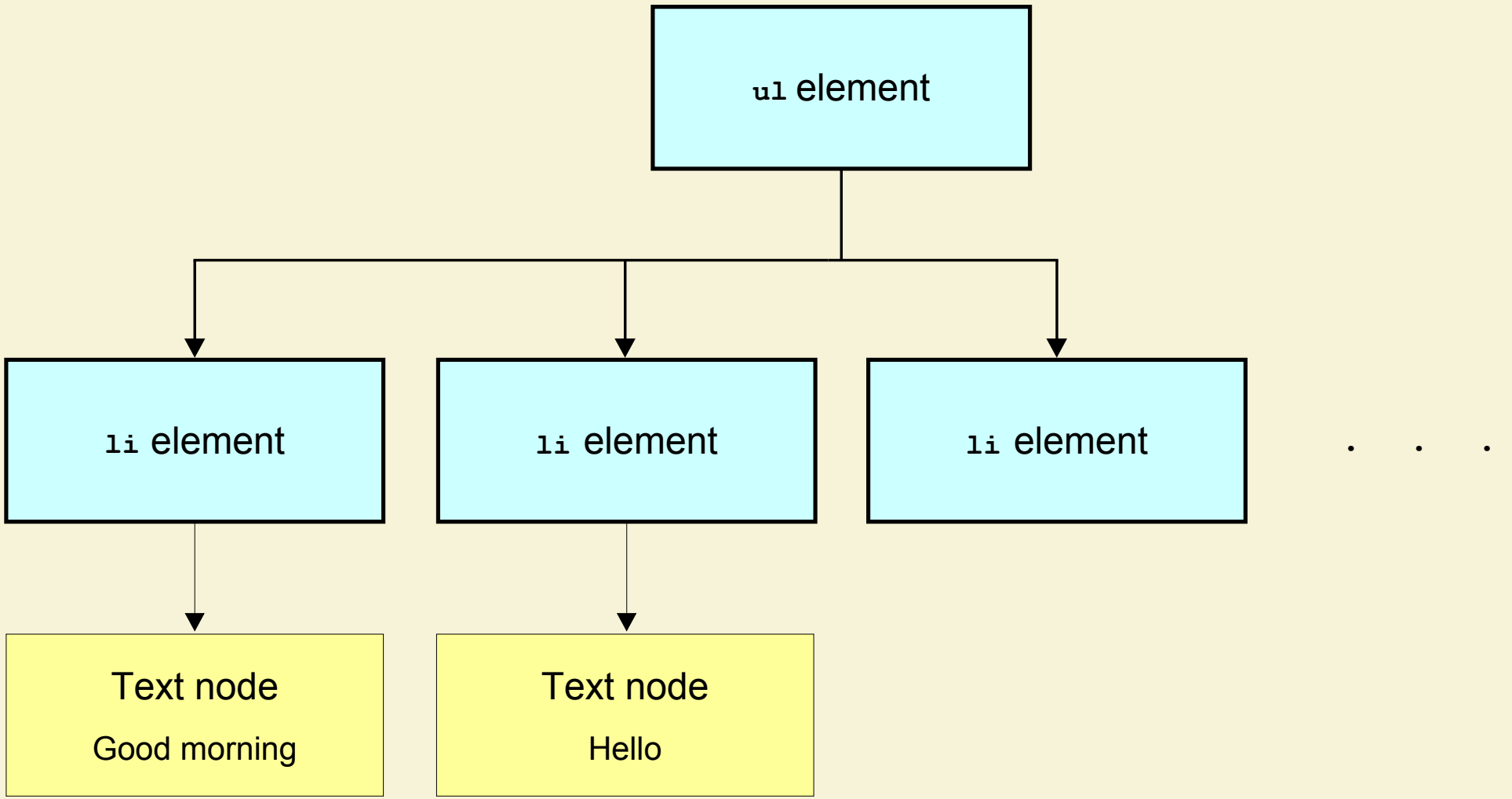
Copy it!

Click here to open the example

```
<!DOCTYPE html>
<html>
<body>
<script>
function myFunction() {
    var the_node=document.getElementById("myList").lastChild;
    var the_clone=the_node.cloneNode();
    document.getElementById("myList").appendChild(the_clone);
}
</script>
<ul id="myList"><li>Good morning</li><li>Hello</li></ul>

<p>Click on the button to cloneNode()</p>
<button onclick="myFunction()">Copy it!</button>
</body>
</html>
```


ul element
li element
li element
Text node
Good morning
Text node
Hello


ul element
li element
li element
li element
. . .
Text node
Good morning
Text node
Hello

CLONING A BRANCH

Use *node*`.cloneNode(true)`

copy the node together with its children

EXAMPLE

1. A list item branch `<li>` with text node child are copied
2. The copy is then added to the end of the list

- Good morning
- Hello

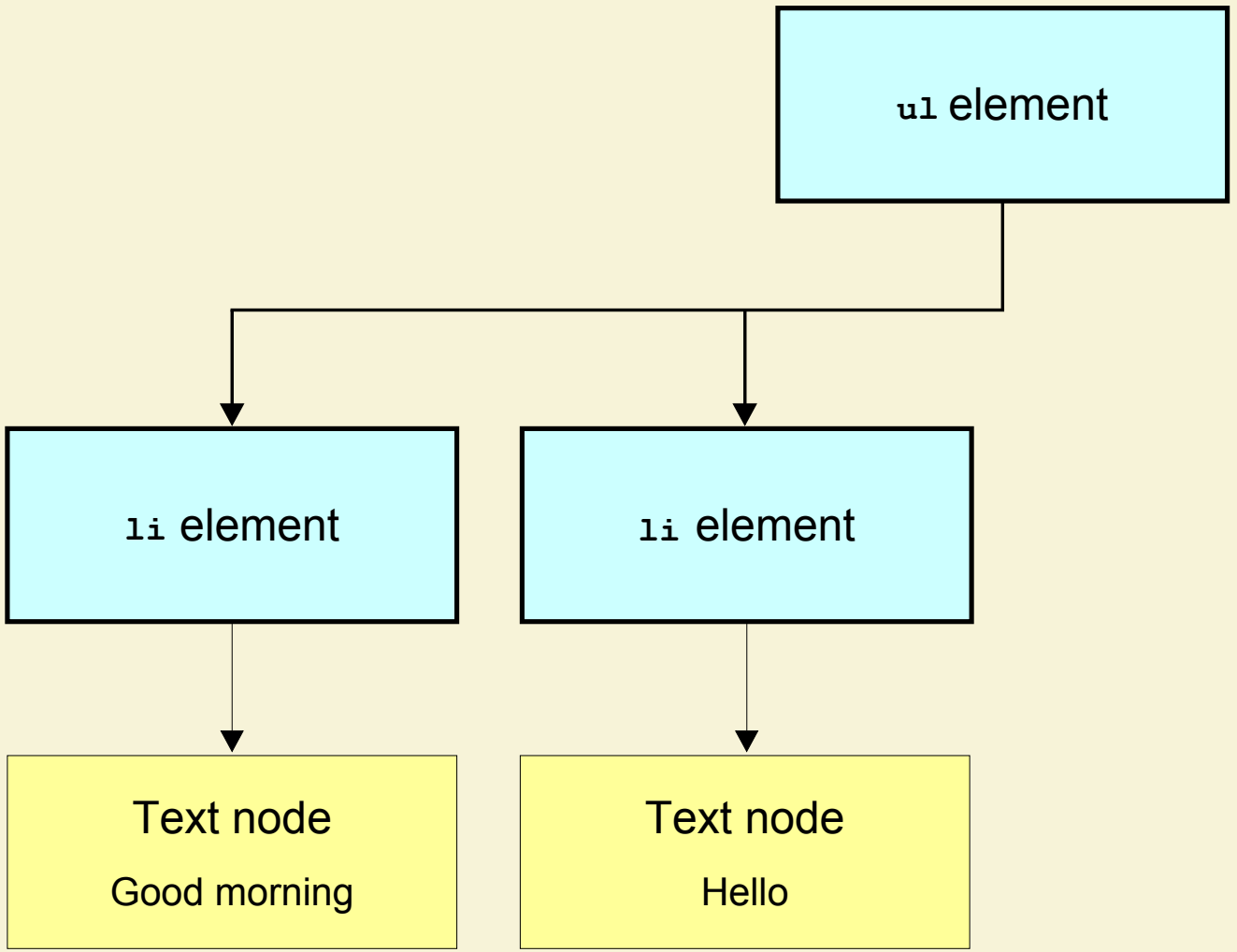
Click on the button to cloneNode(true)

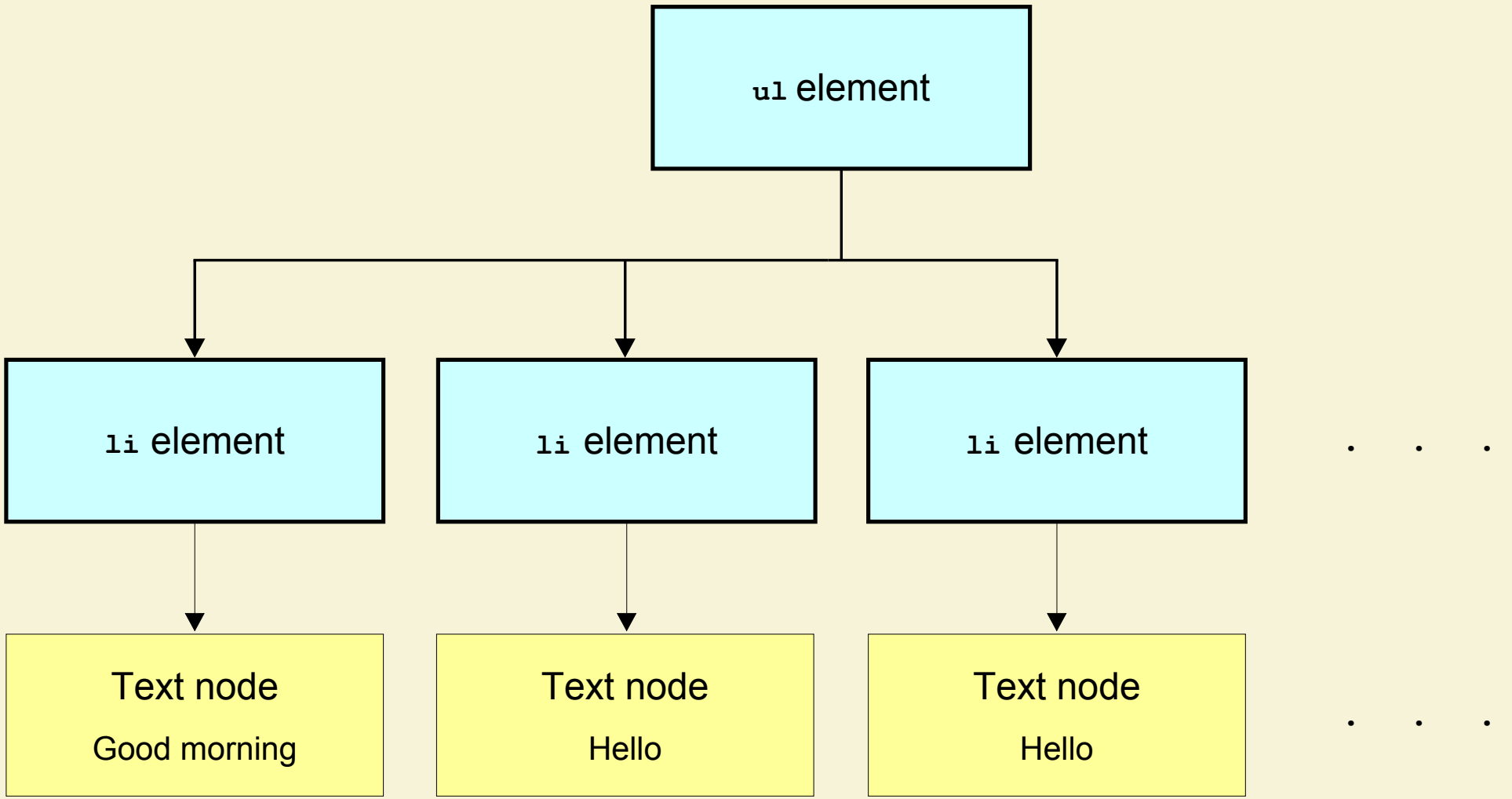
Copy it!

Click here to open the example

```
<!DOCTYPE html>
<html>
<body>
<script>
function myFunction() {
    var the_node=document.getElementById("myList").lastChild;
    var the_clone=the_node.cloneNode(true);
    document.getElementById("myList").appendChild(the_clone);
}
</script>
<ul id="myList"><li>Good morning</li><li>Hello</li></ul>

<p>Click on the button to cloneNode(true)</p>
<button onclick="myFunction()">Copy it!</button>
</body>
</html>
```


ul element
li element
li element
Text node
Good morning
Text node
Hello


ul element
li element
li element
li element
Text node
Good morning
Text node
Hello
Text node
Hello