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`

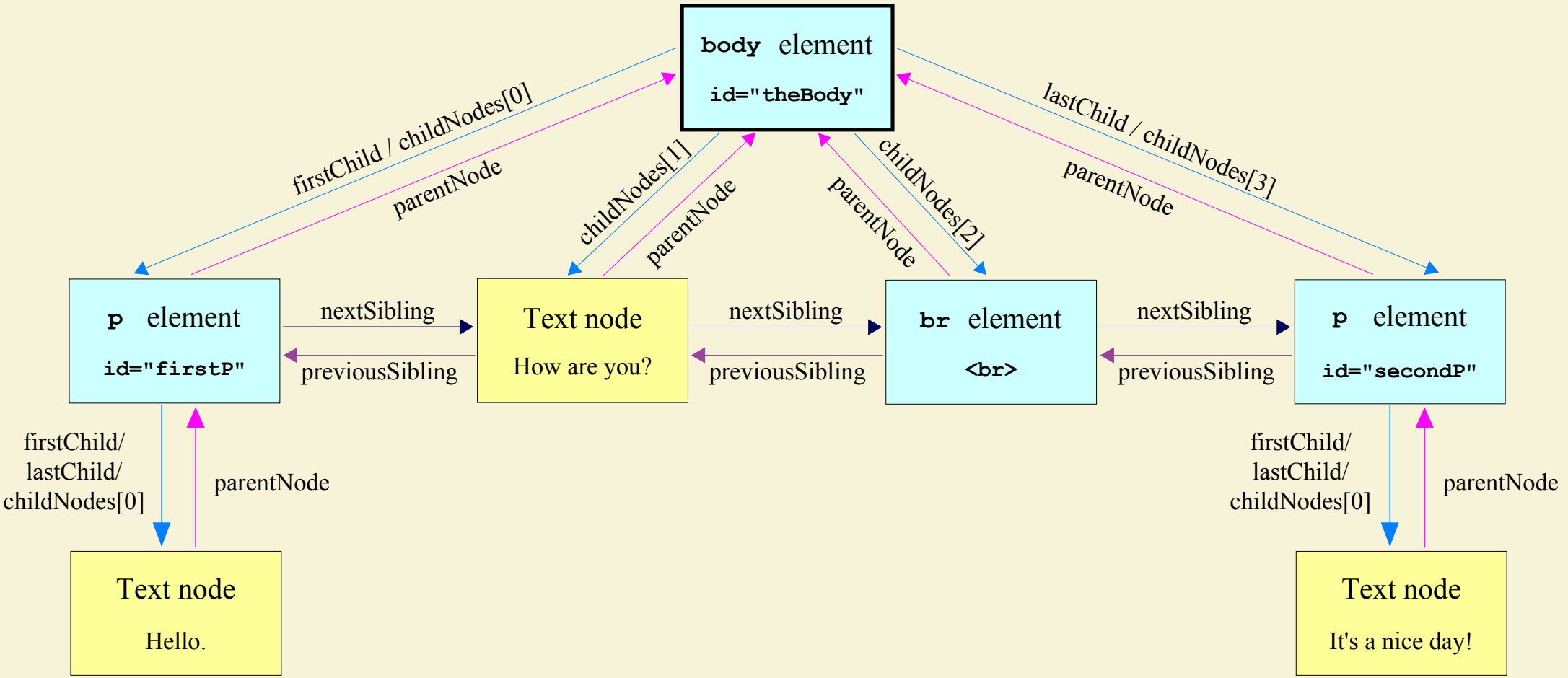
NODE RELATIONSHIPS

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
Text node
Hello.
firstChild/
lastChild/
childNodes[0]
parentNode
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);
}
```

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

```
// 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).

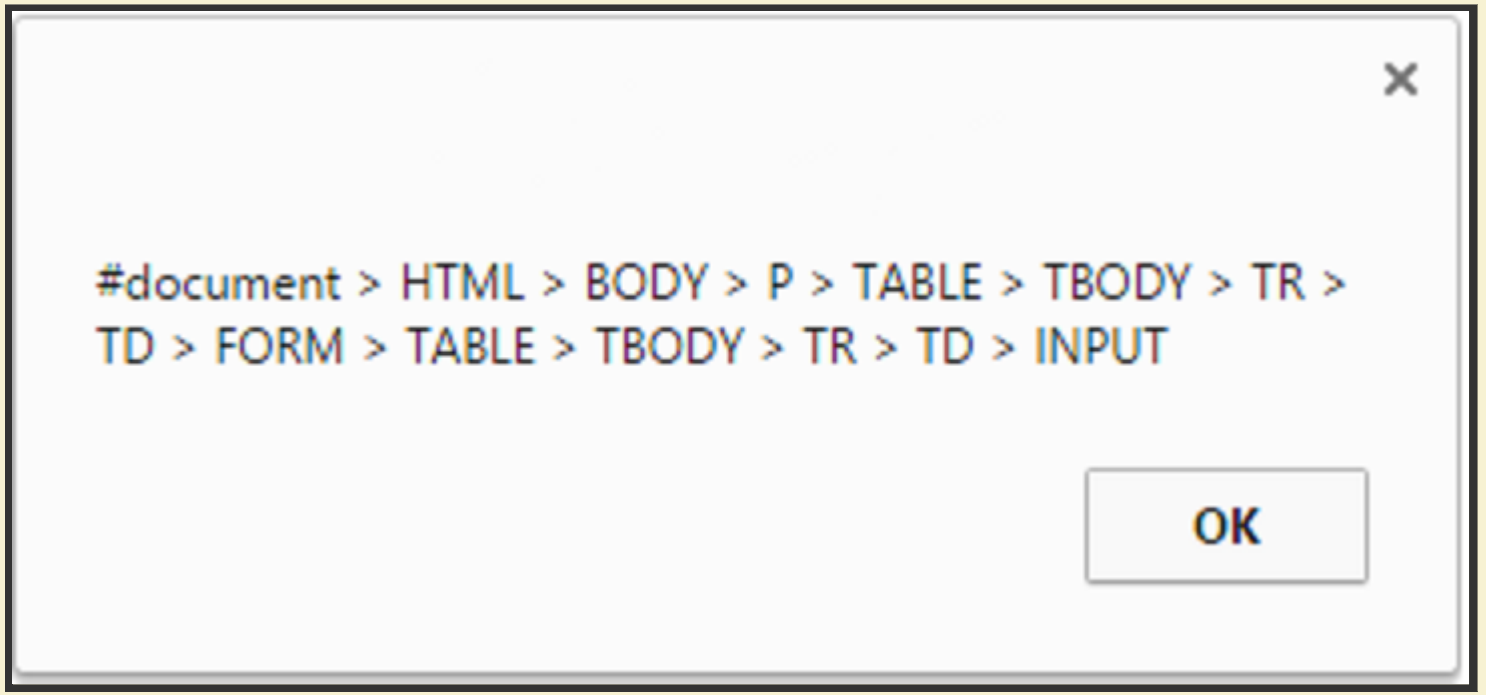
LISTS

- 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!		

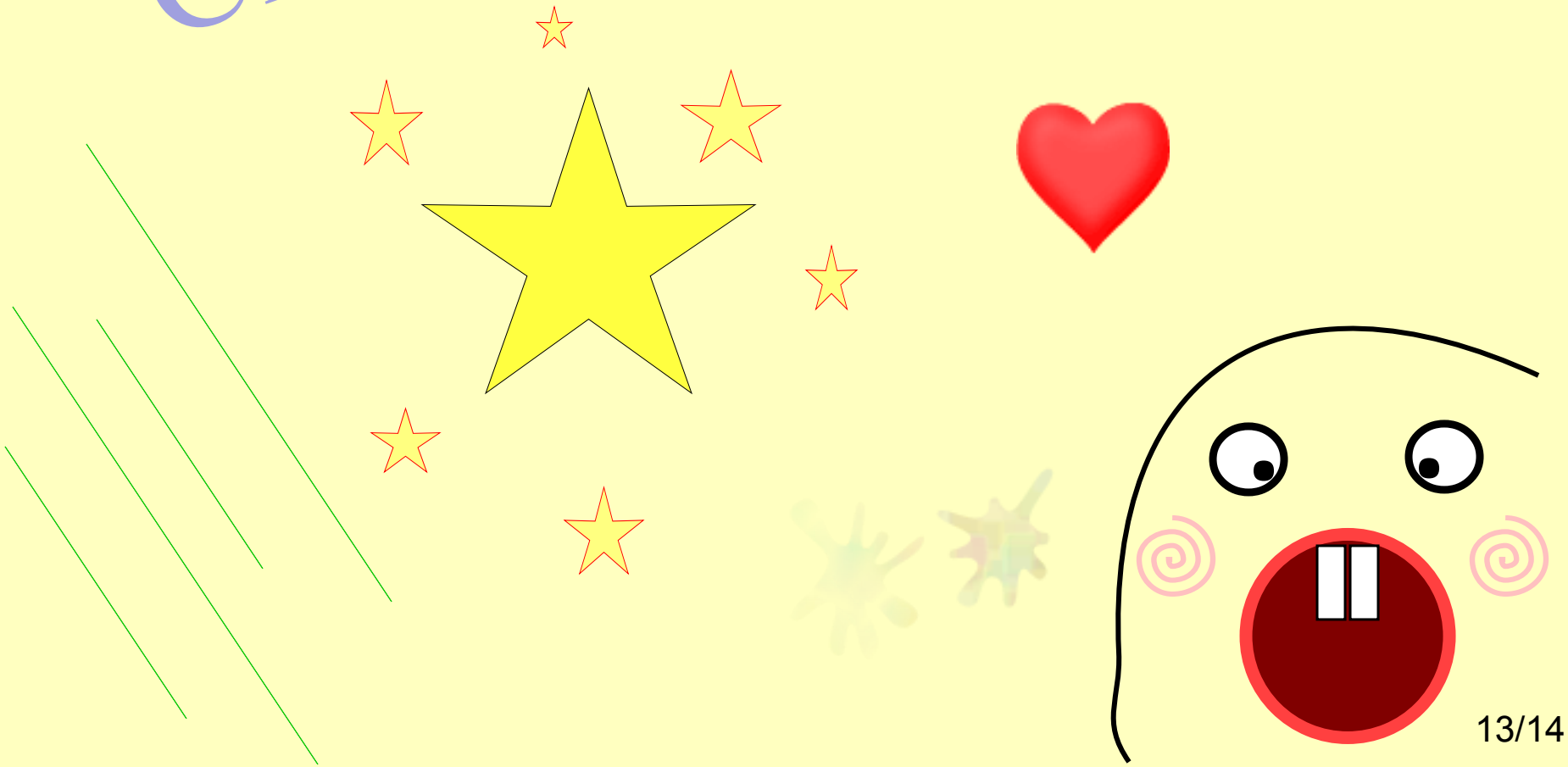
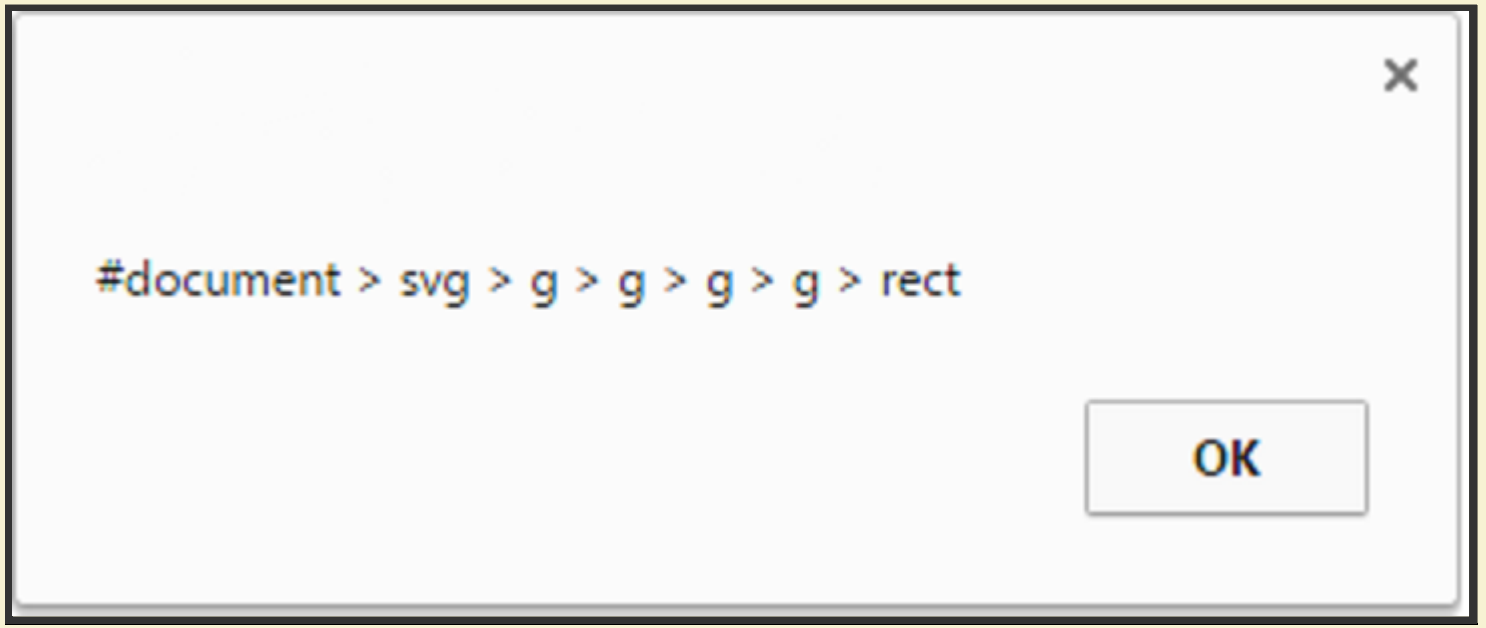
HTML RESULT


#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