

**Department of Computer Science**  
**City University of Hong Kong**  
**CS2204 Fundamentals of Internet Application Development**  
**Tutorial 10a**

**Learning Outcomes:**

- learn different ways to setup event handler
- study how the predefined object *this* is affected by the setup of event handler

Download the file from canvas tutorial10a.zip. Extract the zip file and check how the page works. The event handlers are set up with *onmouseover* attributes in HTML.

1. Rewrite the setup of event handlers by Javascript only. The function *playAudio(x)* takes a parameter but event handler set up by Javascript cannot use the function call operator ( ). Consider the use of an anonymous function and call the event handler inside with argument, e.g. `obj.onmouseover = function() {playAudio(0)}`
2. The codes of setting up 7 event handlers line by line look clumsy. Try the following and see if it works :

```
notes=document.querySelectorAll('.note');
for (i=0; i<notes.length; i++) {
    notes[i].onmouseover=function () {playAudio(i)}
}
```

What is the result?

3. Now use Javascript and a loop to do the event handler setup without the use of ( ) operator. Rewrite the function *playAudio( )* to use the *this* object instead of parameter. The *this* object is defined as the object which the function is attached when the function is called. Use of *this* object automatically gives the object that was mouse-over.

```
function playAudio() {
    audio[(parseInt(this.id.charAt(2)))].play();
}
```

4. Now change back the event handler set up to using HTML, i.e.

```
<div id="id0" onmouseover="playAudio();" class="note">
</div>
```

What is the result? In the function *playAudio()*, add an `alert(this.location)` before the `audio[(parseInt(this.id.charAt(2)))].play()` and see what will be output. Explain. Now change the HTML and function *playAudio()* to :

```
<div id="id0" onmouseover="playAudio(this);" class="note">
</div>
```

```
function playAudio(t) {
    audio[(parseInt(t.id.charAt(2)))].play();
}
```

What is the result?