HTML 5 Drag & Drop

Bok, Jong Soon
jongsoon.bok@gmail.com
www.javaexpert.co.kr

HTML5 Drag & Drop

- Is a part of the HTML5 standard.
- Microsoft IE added drag and drop back in 1999 in IE5.
- Safari had implemented IE’s API.
- All Web Browsers support this API.
- Is a very common feature.
- It is when you "grab" an object and drag it to a different location.
- http://www.w3.org/html/wg/drafts/html/master/editing.html#dnd

HTML5 Drag & Drop Process

1. Make an element **draggable**.
    - To make an element **draggable**, set the **draggable** attribute to **true**.

```
<img draggable="true">
```

HTML5 Drag & Drop Process (Cont.)

2. What to drag
   - Specify what should happen when the element is dragged.
   - The `ondragstart` attribute calls a function.
   - Specifies what data to be dragged.
   - The `dataTransfer.setData()` method sets the data type and the value of the dragged data.

```
function drag(evt) {
   evt.dataTransfer.setData("Text",
                                  ev.target.id);
}
```

HTML5 Drag & Drop Process (Cont.)

3. What to drop
   - The `ondragover` event specifies where the dragged data can be dropped.
   - By default, data/elements cannot be dropped in other elements.
   - To allow a drop, we must prevent the default handling of the element.

```
event.preventDefault();
```

HTML5 Drag & Drop Process (Cont.)

4. Do the Drop
   - When the dragged data is dropped, a drop event occurs.
   - The **ondrop** attribute calls a function.

```
function drop(evt) {
   evt.preventDefault();
   var data=ev.dataTransfer.getData("Text");
   evt.target.appendChild(
                  document.getElementById(data))
}
```

The `DataTransfer` Interface’s attributes

- **`dropEffect`**
  - Returns the kind of operation that is currently selected.
  - **`none`** | **`copy`** | **`link`** | **`move`**
- **`effectAllowed`**
  - Returns the kinds of operations that are to be allowed.
  - Can be set to change the allowed operations.
  - **`none`** | **`copy`** | **`copyLink`** | **`copyMove`** | **`link`** | **`linkMove`** | **`move`** | **`all`** | **`uninitialized`**

The `DataTransfer` Interface’s attributes (Cont.)

- `items`
  - Returns a `DataTransferItemList` object, with the drag data.
- `types`
  - Returns an array listing the formats that were set in the dragstart event.
  - In addition, if any files are being dragged, then one of the types will be the string "Files".
- `files`
  - Returns a FileList of the files being dragged, if any.

The `DataTransfer` Interface’s Methods

- `setDragImage(element, x, y)`
  - Uses the given element to update the drag feedback, replacing any previously specified feedback.
- `getData(format)`
  - Returns the specified data.
  - If there is no such data, returns the empty string.
- `setData(format, data)`
  - Adds the specified data.

The `DataTransfer` Interface's Methods (Cont.)

- `setData(format, data)`
  - Adds the specified data.
- `clearData([format])`
  - Removes the data of the specified formats.
  - Removes all data if the argument is omitted.

Events

Event Name	Target	Action
`dragstart`	Source node	Initiate the drag & drop operation
`drag`	Source node	Continue the drag & drop operation
`dragenter`	Immediate user selection or the body element	Reject immediate user selection as potential target element
`dragexit`	Previous target element	None
`dragleave`	Previous target element	None
`dragover`	Current target element	Reset the current drag operation to "none"
`drop`	Current target element	Varies
`dragend`	Source node	Varies

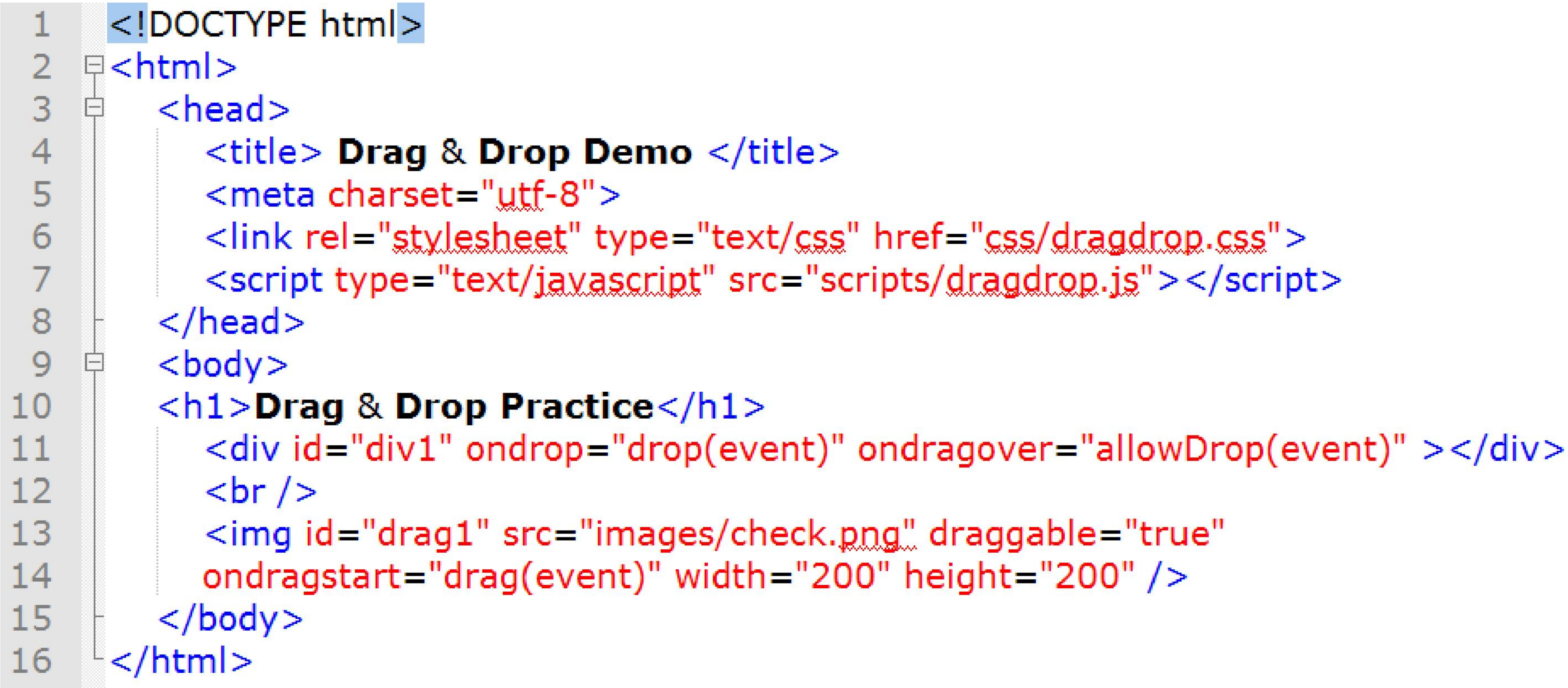
The `draggable` Attribute

- Returns `true` if the element is draggable; otherwise, returns `false`.
- Can be set, to override the default and set the draggable content attribute.
- The `true` state means the element is draggable; the `false` state means that it is not.
- The `auto` state uses the default behavior of the user agent.

Lab1 : Drag & Drop

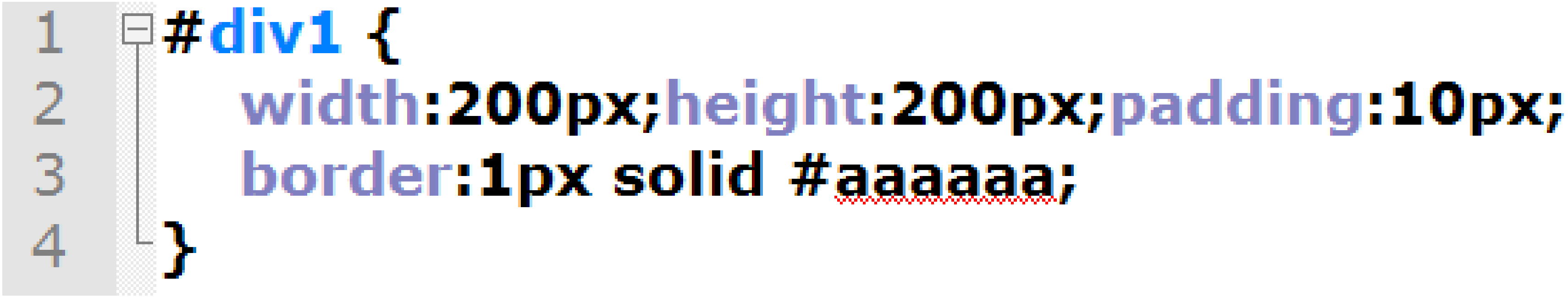
- Web Browsers
  - IE10, Firefox, Google Chrome, Opera, Safari
- Text Editors
  - Notepad++ or Editplus
- Files
  - dragdrop.html
  - images/check.png
  - scripts/dragdrop.js
  - css/dragdrop.css

Lab1 : dragdrop.html



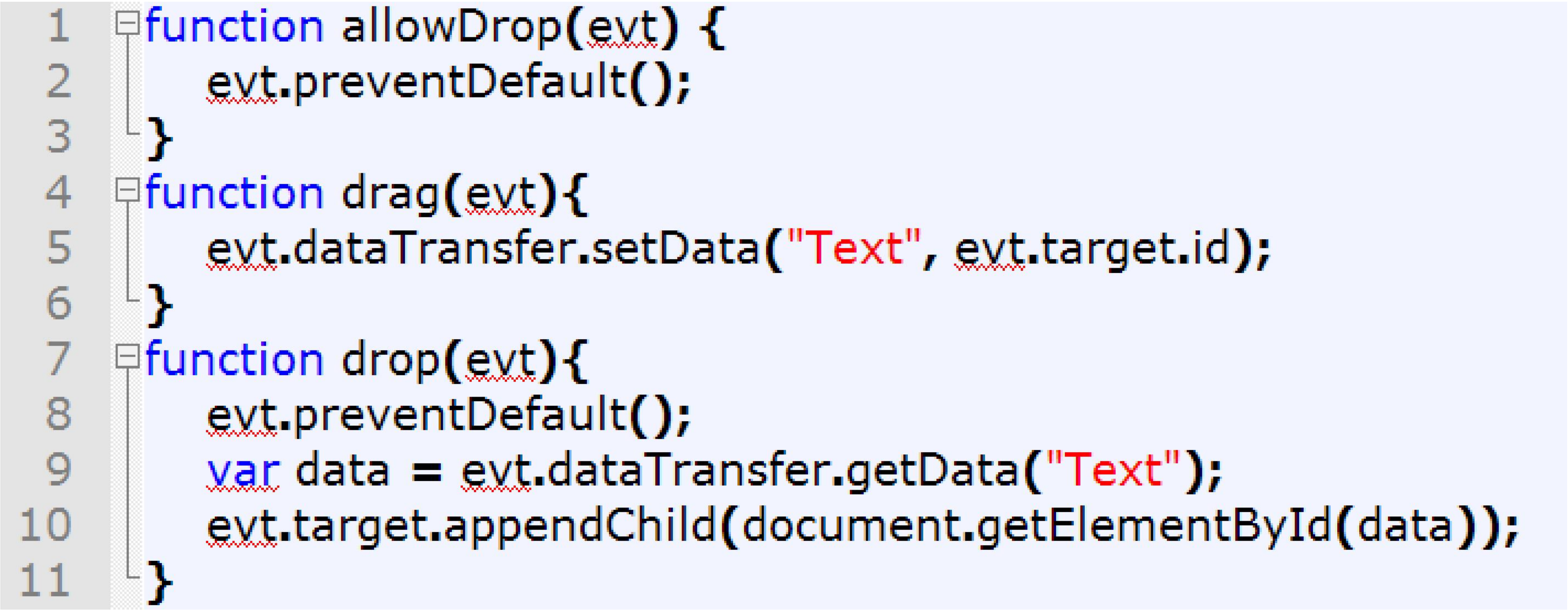
```html
<!DOCTYPE html>
<html>
    <head>
        <title> Drag & Drop Demo </title>
        <meta charset="utf-8">
        <link rel="stylesheet" type="text/css" href="css/dragdrop.css">
        <script type="text/javascript" src="scripts/dragdrop.js"></script>
    </head>
    <body>
    <h1>Drag & Drop Practice</h1>
        <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)" ></div>
        <br />
        <img id="drag1" src="images/check.png" draggable="true"
        ondragstart="drag(event)" width="200" height="200" />
    </body>
</html>
```

Lab1 : dragdrop.css



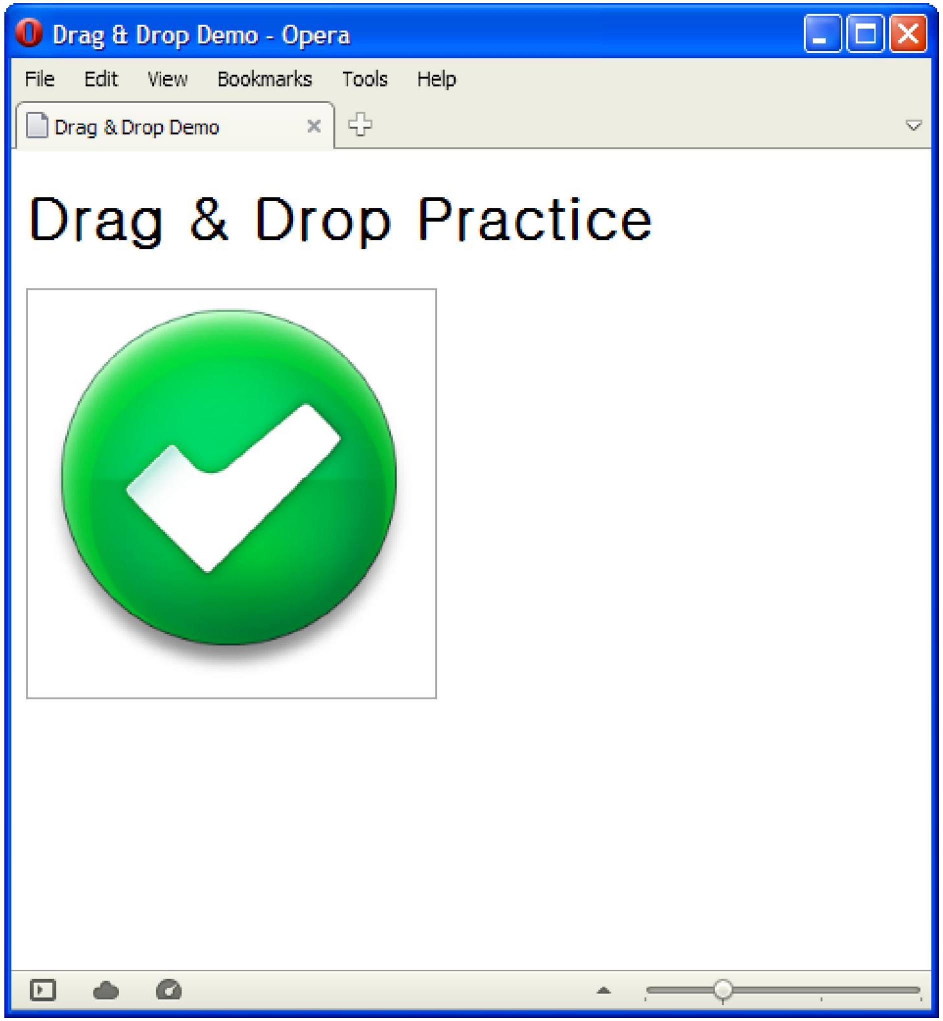
```css
#div1 {
    width:200px;height:200px;padding:10px;
    border:1px solid #aaaaaa;
}
```

Lab1 : dragdrop.js



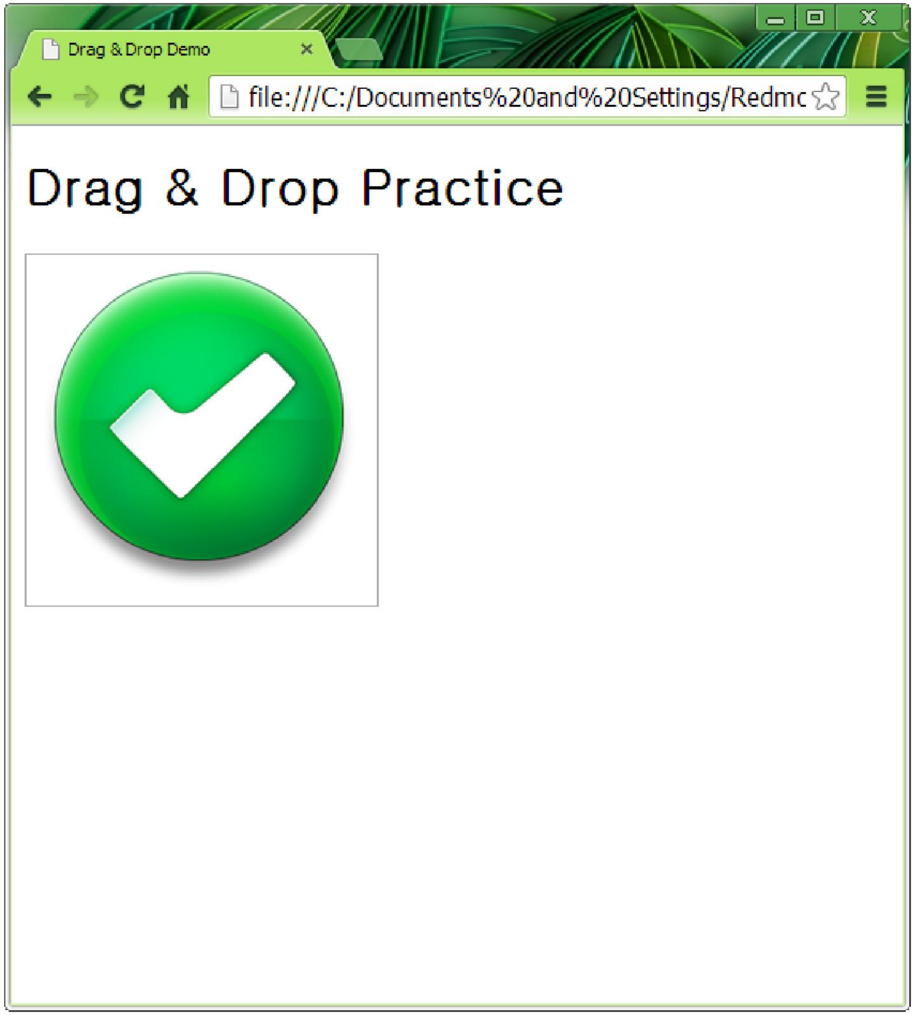
```
function allowDrop(evt) {
    evt.preventDefault();
}
function drag(evt){
    evt.dataTransfer.setData("Text", evt.target.id);
}
function drop(evt){
    evt.preventDefault();
    var data = evt.dataTransfer.getData("Text");
    evt.target.appendChild(document.getElementById(data));
}
```

Lab1 : Result



Drag & Drop Demo - Opera

Drag & Drop Practice

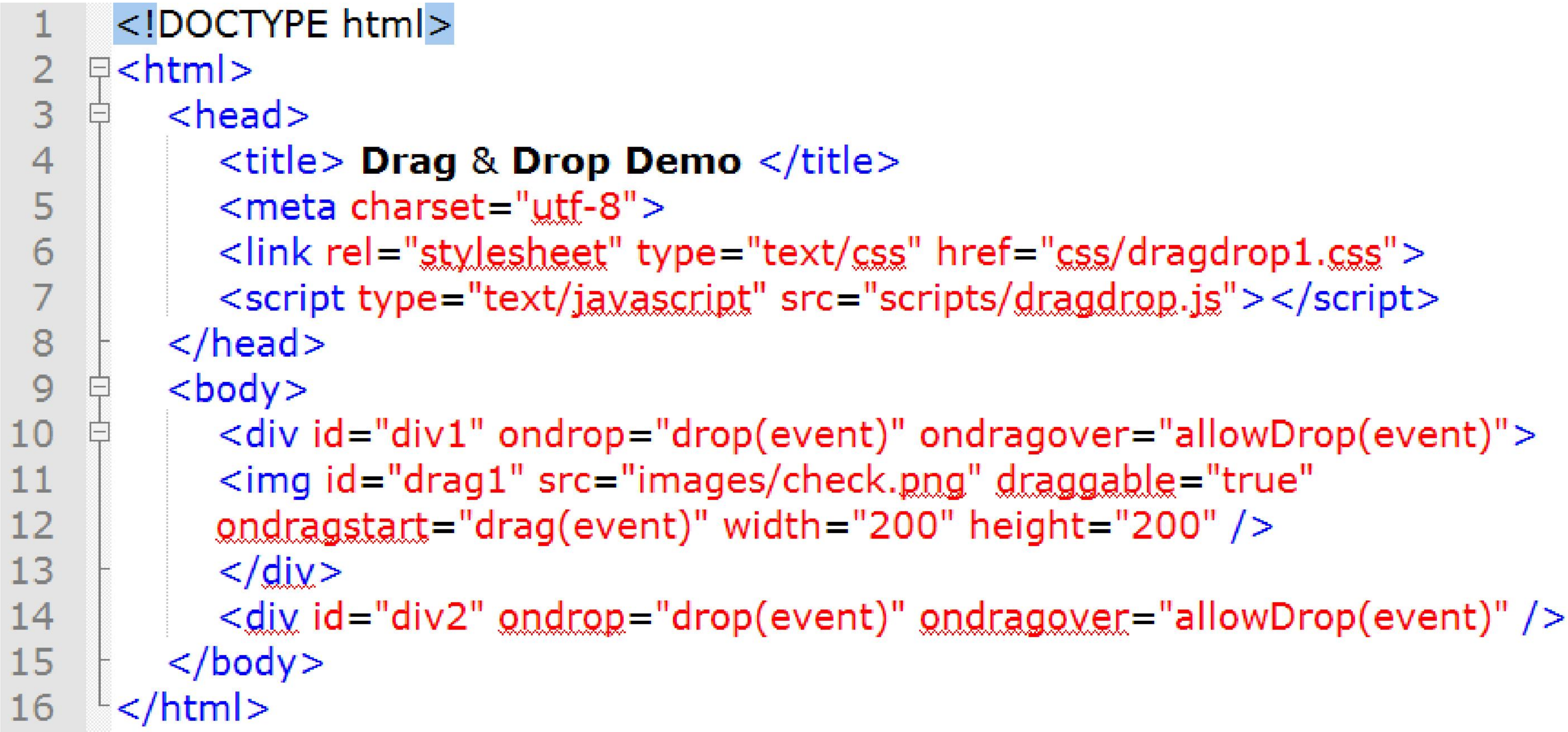


Drag & Drop Practice

Lab2 : Drag & Drop

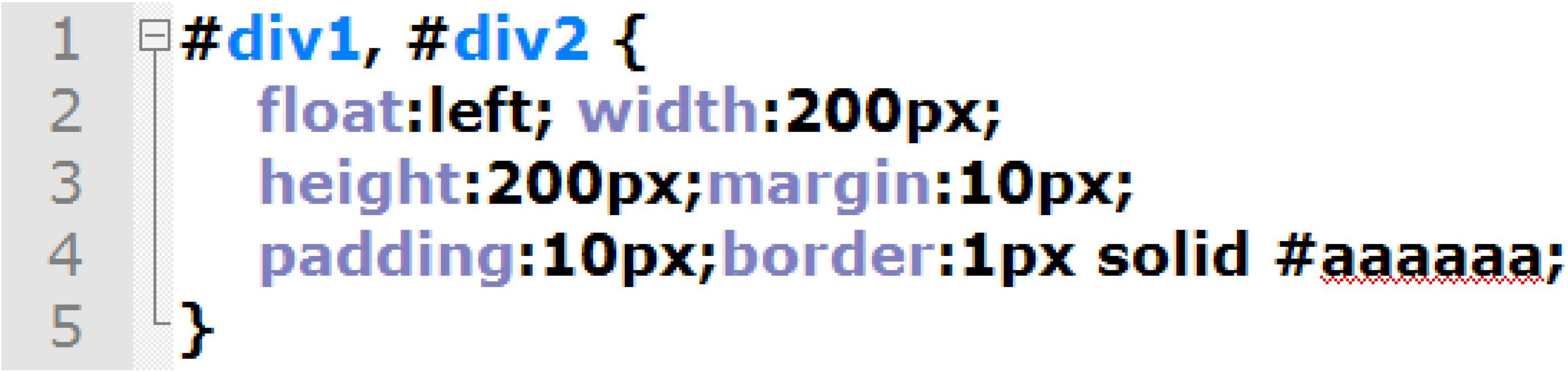
- Web Browsers
  - IE10, Firefox, Google Chrome, Opera, Safari
- Text Editors
  - Notepad++ or Editplus
- Files
  - dragdrop1.html
  - images/check.png
  - scripts/dragdrop.js
  - css/dragdrop1.css

Lab2 : dragdrop1.html



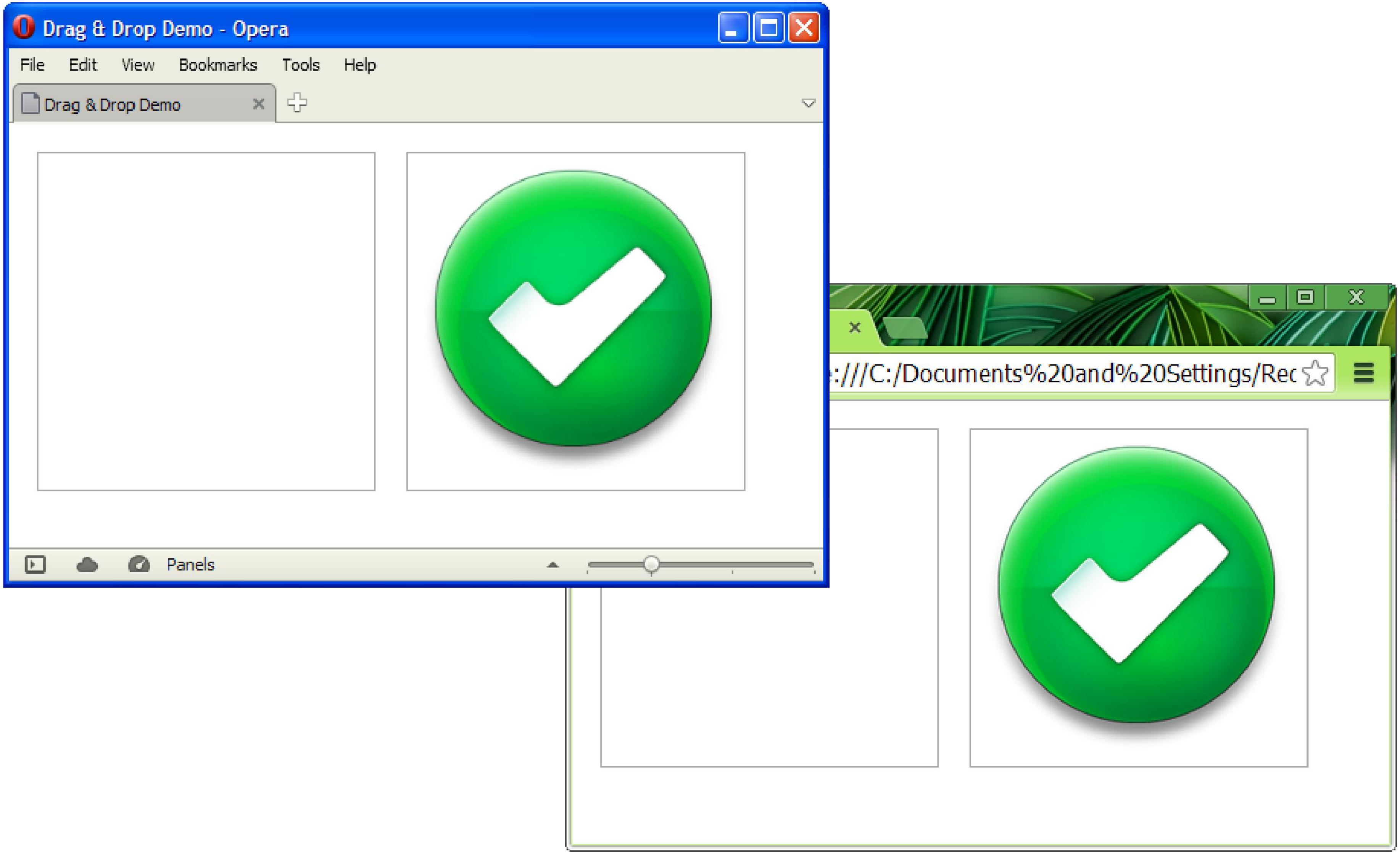
```
<!DOCTYPE html>
<html>
    <head>
        <title> Drag & Drop Demo </title>
        <meta charset="utf-8">
        <link rel="stylesheet" type="text/css" href="css/dragdrop1.css">
        <script type="text/javascript" src="scripts/dragdrop.js"></script>
    </head>
    <body>
        <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">
        <img id="drag1" src="images/check.png" draggable="true"
        ondragstart="drag(event)" width="200" height="200" />
        </div>
        <div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)" />
    </body>
</html>
```

Lab2 : dragdrop1.css



```css
#div1, #div2 {
    float:left; width:200px;
    height:200px;margin:10px;
    padding:10px;border:1px solid #aaaaaa;
}
```

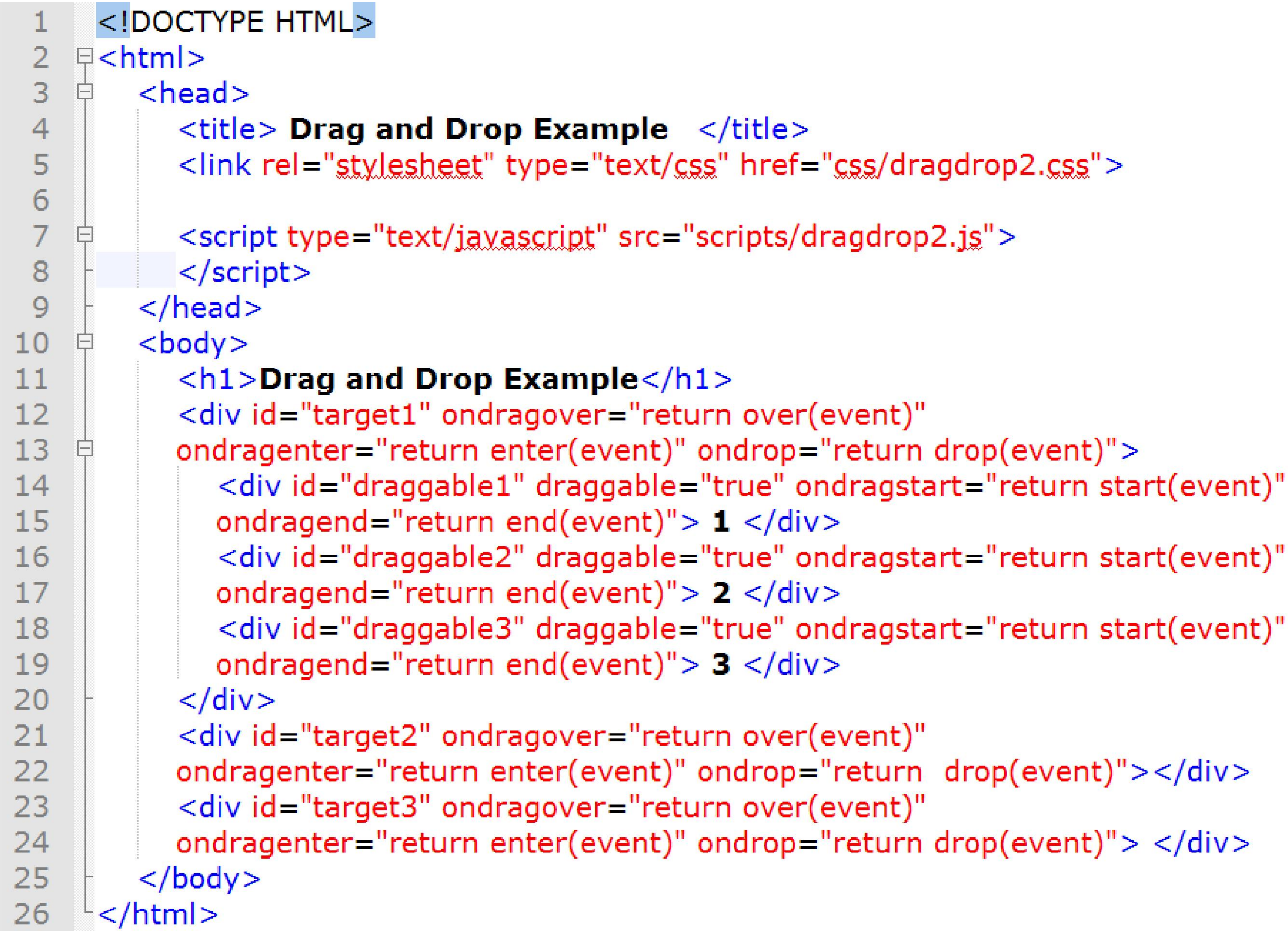
Lab2 : Result


Drag & Drop Demo - Opera
File
Edit
View
Bookmarks
Tools
Help
Drag & Drop Demo
Panels
:///C:/Documents%20and%20Settings/Rec

Lab3 : Drag & Drop

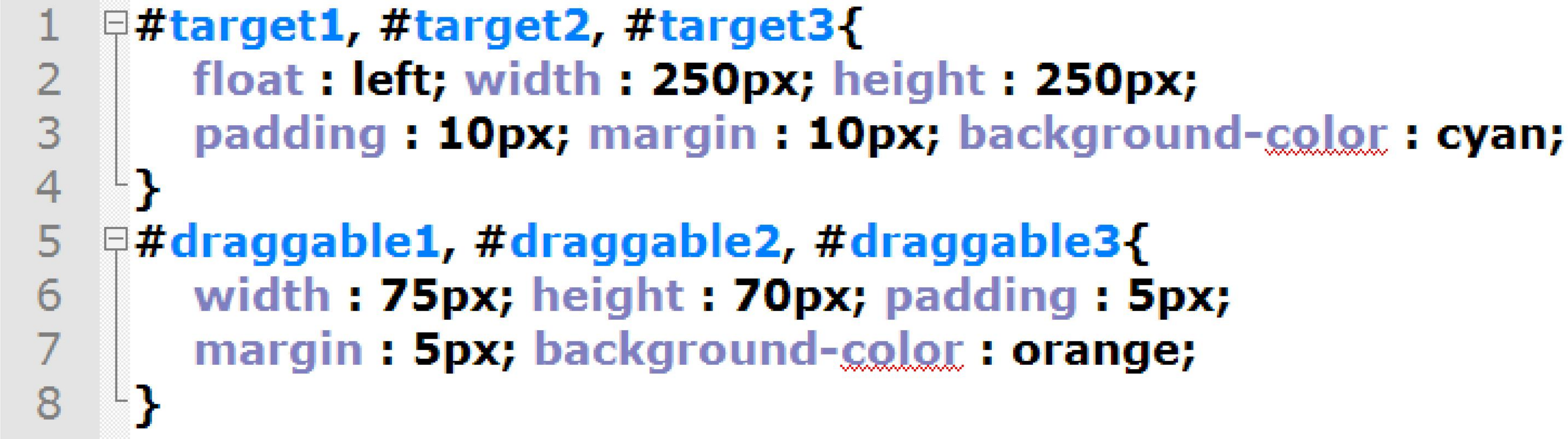
- Web Browsers
  - IE10, Firefox, Google Chrome, Opera, Safari
- Text Editors
  - Notepad++ or Editplus
- Files
  - dragdrop2.html
  - scripts/dragdrop2.js
  - css/dragdrop2.css

Lab3 : dragdrop2.html



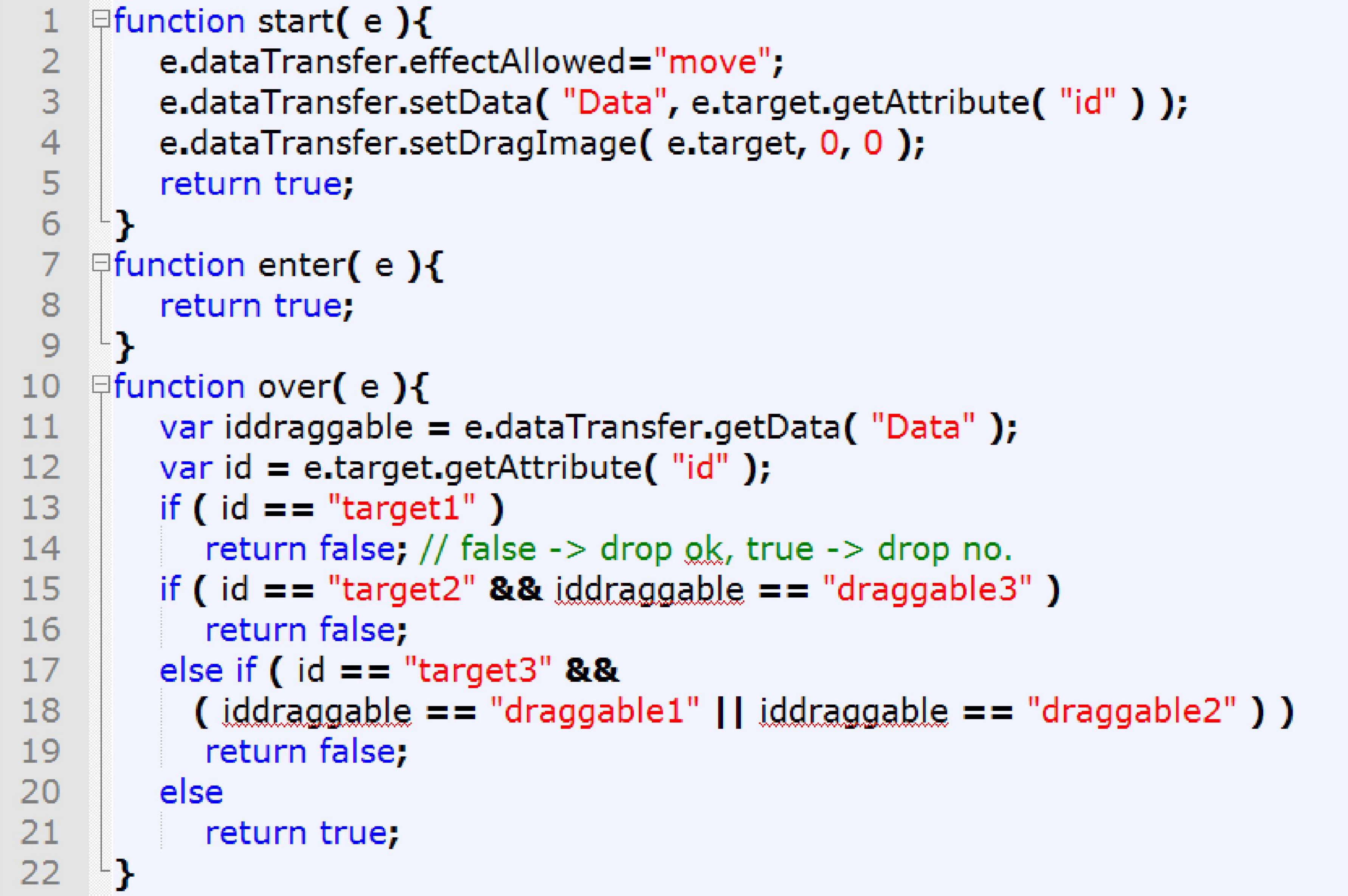
```html
<!DOCTYPE HTML>
<html>
    <head>
        <title> Drag and Drop Example   </title>
        <link rel="stylesheet" type="text/css" href="css/dragdrop2.css">

        <script type="text/javascript" src="scripts/dragdrop2.js">
        </script>
    </head>
    <body>
        <h1>Drag and Drop Example</h1>
        <div id="target1" ondragover="return over(event)"
        ondragenter="return enter(event)" ondrop="return drop(event)">
            <div id="draggable1" draggable="true" ondragstart="return start(event)"
            ondragend="return end(event)"> 1 </div>
            <div id="draggable2" draggable="true" ondragstart="return start(event)"
            ondragend="return end(event)"> 2 </div>
            <div id="draggable3" draggable="true" ondragstart="return start(event)"
            ondragend="return end(event)"> 3 </div>
        </div>
        <div id="target2" ondragover="return over(event)"
        ondragenter="return enter(event)" ondrop="return  drop(event)"></div>
        <div id="target3" ondragover="return over(event)"
        ondragenter="return enter(event)" ondrop="return drop(event)"> </div>
    </body>
</html>
```

Lab3 : dragdrop2.css



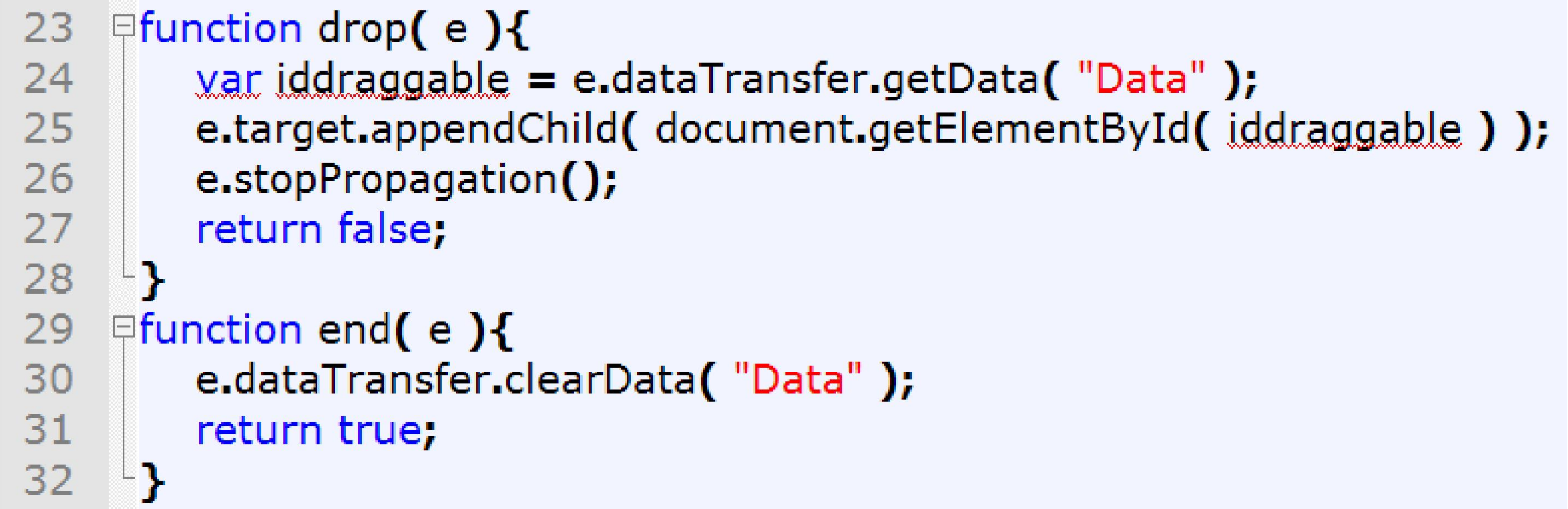
```css
#target1, #target2, #target3{
    float : left; width : 250px; height : 250px;
    padding : 10px; margin : 10px; background-color : cyan;
}
#draggable1, #draggable2, #draggable3{
    width : 75px; height : 70px; padding : 5px;
    margin : 5px; background-color : orange;
}
```

Lab3 : dragdrop2.js(1/2)



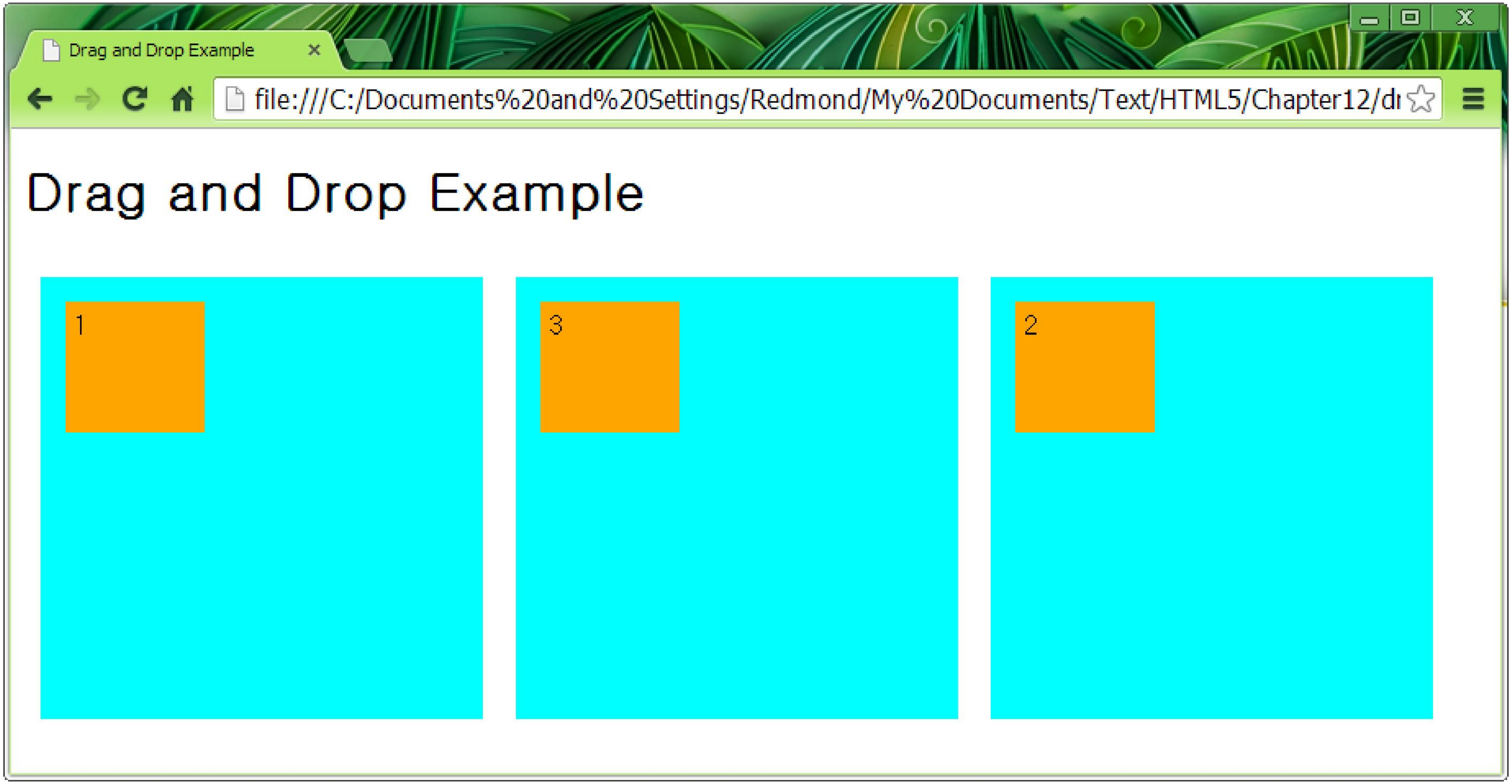
```
function start( e ){
    e.dataTransfer.effectAllowed="move";
    e.dataTransfer.setData( "Data", e.target.getAttribute( "id" ) );
    e.dataTransfer.setDragImage( e.target, 0, 0 );
    return true;
}
function enter( e ){
    return true;
}
function over( e ){
    var iddraggable = e.dataTransfer.getData( "Data" );
    var id = e.target.getAttribute( "id" );
    if ( id == "target1" )
        return false; // false -> drop ok, true -> drop no.
    if ( id == "target2" && iddraggable == "draggable3" )
        return false;
    else if ( id == "target3" &&
       ( iddraggable == "draggable1" || iddraggable == "draggable2" ) )
        return false;
    else
        return true;
}
```

Lab3 : dragdrop2.js(2/2)



```
function drop( e ){
    var iddraggable = e.dataTransfer.getData( "Data" );
    e.target.appendChild( document.getElementById( iddraggable ) );
    e.stopPropagation();
    return false;
}
function end( e ){
    e.dataTransfer.clearData( "Data" );
    return true;
}
```

Lab3 : Result



file:///C:/Documents%20and%20Settings/Redmond/My%20Documents/Text/HTML5/Chapter12/d

Drag and Drop Example