中山大學
SUN YAT-SEN UNIVERSITY

Lecture 18
Advanced Events and Client Side Validations

SE-805 Web 2.0 Programming (supported by Google)

http://my.ss.sysu.edu.cn/courses/web2.0/

School of Software, Sun Yat-sen University

Outline

- **Advanced events**
- Client side validations

Page/Window Events

Name	Description
load	the browser loads the page
unload	the browser exits the page
resize	the browser window is resized
contextmenu	the user right-clicks to pop up a context menu
error	an error occurs when loading a document or an image

- The above events can be handled on the global `window` object. Also:

```
// best way to attach event handlers on page load
window.onload = function() { ... };
document.observe("dom:loaded", function() {
  $("orderform").observe("submit", verify);
});
```

Form Events

Event Name	Description
submit	form is being submitted
reset	form is being reset
change	the text or state of a form control has changed

```
window.observe("load", function() {
  $("orderform").observe("submit", verify);
});

function verify(event) {
  if ($("zipcode").value.length < 5) {
    event.stop();          // cancel form submission unless
  }                        // zip code is 5 chars long
}
```

Prototype and Forms

```
$("id")["name"]
```

- Gets parameter with given **name** from form with given **id**

```
$F("id")
```

- $F returns the value of a form control with the given **id**

```
var name = $F("username");
if (name.length < 4) {
  $("username").clear();
  $("login").disable();
}
```

- Other form control methods:

activate	clear	disable	enable
focus	getValue	present	select

Keyboard/Text Events

Name	Description
keydown	user presses a key while this element has keyboard focus
keyup	user releases a key while this element has keyboard focus
keypress	user presses and releases a key while this element has keyboard focus
focus	this element gains keyboard focus
blur	this element loses keyboard focus
select	this element's text is selected or deselected)

- **focus**: the attention of the user's keyboard (given to one element at a time)

Key Event Objects

Property Name	Description
`keyCode`	ASCII integer value of key that was pressed (convert to char with String.fromCharCode)
`altKey, ctrlKey, shiftKey`	true if Alt/Ctrl/Shift key is being held

Event.KEY_BACKSPACE	Event.KEY_DELETE	Event.KEY_DOWN	Event.KEY_END
Event.KEY_ESC	Event.KEY_HOME	Event.KEY_LEFT	Event.KEY_PAGEDOWN
Event.KEY_PAGEUP	Event.KEY_RETURN	Event.KEY_RIGHT	Event.KEY_TAB
Event.KEY_UP			

Prototype's key code constants

- **Issue**: if the event you attach your listener to doesn't have the focus, you won't hear the event
- **Possible solution**: attach key listener to entire page body, outer element, etc.

Outline

- Advanced events
- **Client side validations**

Client-Side Validation

```HTML
<form id="exampleform" action="http://foo.com/foo.php">
```

```JS
window.onload = function() {
  $("exampleform").onsubmit = checkData;
};

function checkData(event) {
  if ($("city").value == "" || $("state").value.length != 2) {
    Event.stop(event);
    alert("Error, invalid city/state.");  // show error message
  }
}
```

- Forms expose **`onsubmit`** and **`onreset`** events
- To abort a form submission, call Prototype's Event.stop on the event

Regular Expression in JavaScript

- `string.match(regex)`
  - If `string` fits the pattern, returns the matching text; else returns `null`
  - Can be used as a `Boolean` **truthy/falsey** test:

```
var name = $("name").value;
if (name.match(/[a-z]+/)) { ... }
```

- An `i` can be placed after the **regex** for a case-insensitive match
  - `name.match(/Eric/i)` will match "`eric`", "ERic", ...

Replacing Text with Regular Expressions

- `string.replace(regex, "text")`
  - Replaces the first occurrence of given pattern with the given text
  - `var str = "Wang Qing";`
    `str.replace(/[a-z]/, "x"); //returns "Wxng Qing"`
  - Returns the **modified** string as its result; must be stored
    `str = str.replace(/[a-z]/, "x")`
- A **g** can be placed after the regex for a global match (replace all occurrences)
  - `str.replace(/[a-z]/, "x"); //returns "Wxxx Qxxx"`
- Using a regex as a filter
  - `str = str.replace(/[^A-Z]+/g, ""); // turns str into "WQ"`

Summary

- Advanced events
  - Page / window events
  - Form events
  - Keyboard/text events
  - Key events
  - Prototype form functions
- Client side validations
  - Regular Expression in JavaScript

Exercises

- Write a simple to-do list application as a web page.
  - A element wraps all html elements for this to-do application
  - A form with a textarea for specifying a new to-do item and a "add" button for adding it to the list
  - A list of current to-do items
  - Each item has a checkbox for select
  - Buttons "select all", "deselect all", "remove" (which removes all selected to-do items from the list)
  - When the "add" button is clicked the new to-do item will be inserted to the bottom of the list
  - Use unobtrusive JavaScript
  - add hot-keys to the to-do list
    - "↑" and "↓" keys for moving up and down in the to-do list
    - "Enter" key for selecting/deselecting the current to-do item(s)

Further Readings

- W3C DOM Specification
  http://www.w3.org/TR/DOM-Level-2-Events/events.html
- W3School DOM Events reference
  http://www.w3school.com/dom/dom_events.asp/
- W3School DOM tutorial
  http://www.w3schools.com/htmldom/
- Quirksmode DOM tutorial
  http://www.quirksmode.org/dom/intro.html
- Prototype Learning Center
  http://www.prototypejs.org/learn
- Developer Notes for prototype.js
  http://www.sergiopereira.com/articles/prototype.js.html

Thank you!