

# More Events and Validation

# Page/window events

2

<b>name</b>	<b>description</b>
<u>load</u>	the browser loads the page
<u>unload</u>	the browser exits the page
<u>resize</u>	the browser window is resized
contextmenu	the user right-clicks to pop up a context menu
<u>error</u>	an error occurs when loading a document or an image

# Page/window events

3

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

JS

# Form events

4

event name	description
<u>submit</u>	form is being submitted
<u>reset</u>	form is being reset
<u>change</u>	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
}
```

JS

# Prototype and forms

5

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

JS

- gets parameter with given name from form with given id

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

JS

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

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

JS

# Prototype and forms

6

- other form control methods:

activate

clear

disable

enable

focus

getValue

present

select

# Client-side validation code

7

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

HTML

```
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
    }
}
```

JS

- forms expose onsubmit and onreset events
- to abort a form submission, call Prototype's Event.stop on the event

# Regular expressions in JavaScript

8

- `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(/Xenia/i)` will match “xenia”, “XeNiA”,  
...

# Replacing text with regular expressions

9

- `string.replace(regex, "text")`
  - ▣ replaces the first occurrence of given pattern with the given text
  - ▣ `var str = "Xenia Mountrouidou";  
str.replace(/ [a-z] /, "x")` returns " Xxnia Mountrouidou"
  - ▣ 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] /g, "x")` returns "XXXXXX"

# Keyboard/text events

10

<b>name</b>	<b>description</b>
<u>keydown</u>	user presses a key while this element has keyboard focus
<u>keyup</u>	user releases a key while this element has keyboard focus
<u>keypress</u>	user presses and releases a key while this element has keyboard focus
<u>focus</u>	this element gains keyboard focus
<u>blur</u>	this element loses keyboard focus

# Key event objects

11

property name	description
keyCode <small>prototype's key code constants</small>	ASCII integer value of key that was pressed (convert to char with <a href="#"><u>String.fromCharCode</u></a> )
altKey, ctrlKey, shiftKey	true if Alt/Ctrl/Shift key is being held

# Key event objects

12

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			

- 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.