Chapter 13

How to work with numbers, strings, and dates

Objectives

- How to work with numbers
- The PIG application
- How to work with strings
- How to work with dates and times
- The Count Down application

How to work with numbers

How to use the properties and methods of the Number object

- Properties of the Number object

Property	Shortcut	Description
`Number.MAX_VALUE`		The largest positive value that can be represented.
`Number.MIN_VALUE`		The smallest positive value that can be represented.
`Number.POSITIVE_INFINITY`	Infinity	Represents positive infinity
`Number.NEGATIVE_INFINITY`	- Infinity	Represents negative infinity
`Number.NaN`	NaN	Represents a value that isn’t a number

- Methods of the Number object

Method	Description
`toFixed(digits)`	Returns a string with the number rounded to the specified decimal digits.
`toString(base)`	Returns a string with the number in the give base. If basic is omitted, 10 is used.

How to use the properties and methods of the Number object (cont.)

- Example 1: Testing for Infinity, - Infinity and NaN

```
If(result == Infinity){
   alert(“The result exceeds ” + Number.MAX_VALUE);
}else if (result == -Infinity){
   alert(“The result is below ” + Number.MIN_VALUE);
}else if(isNaN(result)){
   alert(“The result is not a number ”);
}else {alert(“The result is” + result);}
```

- Example 2: Division by zero

```
alert(0/0);    //Display NaN
alert(10/0);      //Display Infinity
```

How to use the properties and methods of the Number object (cont.)

- Example 3: Using the toFixed() method

```
var subtotal = 19.99, rate = 0.075;
var tax = subtotal * rate; //tax is 1.49925
tax = parseFloat(tax.toFixed(2));   //tax is 1.5
alert(tax.toFixed(2));     //display 1.50
```

- Example 4: Implicit use of the toString() method for base 10 conversions

```
var age = parseInt(prompt(“Please enter your age.”));
alert(“Your age is ” + age);
```

How to use the properties and methods of the Math object

- One property of the Math object

Property	Description
`Math.PI`	Returns 3.141592.., which is the radio of the circumference of a circle to its diameter.

- Example 1: The PI property

```
var area = Math.PI * 3 *3;
```

How to use the properties and methods of the Math object(cont.)

- Common methods of the Math object

Method	**Description**
`Math.abs(x)`	Returns the absolute value of x.
`Math.round(s)`	Returns the value of x rounded to the closest integer value.
`Math.ceil(x)`	Returns the value of x rounded to the next higher integer value.
`Math.floor(x)`	Returns the value of x rounded to the next lower integer value.
`Math.pow(x,power)`	Returns the value of x raised to the power specified.
`Math.sqrt(x)`	Return the square root of x.
`Math.max(x1,x2,…)`	Returns the largest value from its parameters.
`Math.min(x1,x2,…)`	Returns the smallest value from its parameters.

How to use the properties and methods of the Math object(cont.)

- Example: The abs() method

```
var result_2a = Math.abs(-3.4);
```

- Example: The round() method

```
var result_3a = Math.round(12.5);
var result_3b = Math.round(-3.4);
```

- Example: The floor() and ceil() methods

```
var result_4a = Math.floor(12.5);   //Return 12
var result_4b = Math.ceil(12.5);    //Return 13
var result_4c = Math.floor(-3.4);   //Return -3
var result_4d = Math.ceil(-3.4); //Return -4
```

How to use the properties and methods of the Math object(cont.)

- Example: The pow() and sqrt() methods

```
var result_5a = Math.pow(2,3);
var result_5b = Math.pow(125,1/3);
var result_5c = Math.sqrt(16);
```

- Example: The min() and max() methods

```
var x=12.5, y = -3.4;
var max = Math.max(x,y);
var min = Math.min(x,y);
```

How to generate a random number

- The random() method of the Math object

Method	Description
`Math.random()`	Returns a random decimal number >=0.0 but <1.0

- Example 1: Generating a random number

```
 var result = Math.random();
```

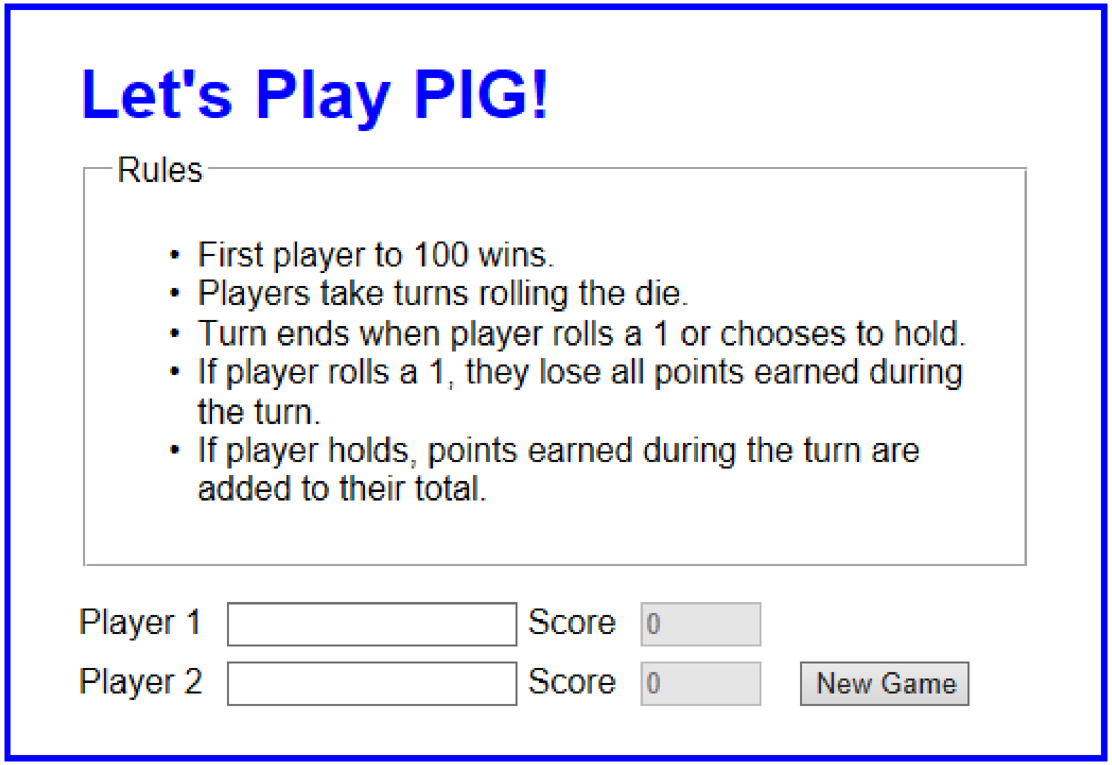
- Example 2: A function that generates a random number

```
var getRandomNumber = function(max){
    var random;
    if(!isNaN(max)){
        random = Math.random();
        random = Math.floor(random * max);
        random = random + 1;
    }
    return random;
}
var randomNumber = getRandomNumber(100);
```

The PIG application

The PIG application

- The User Interface



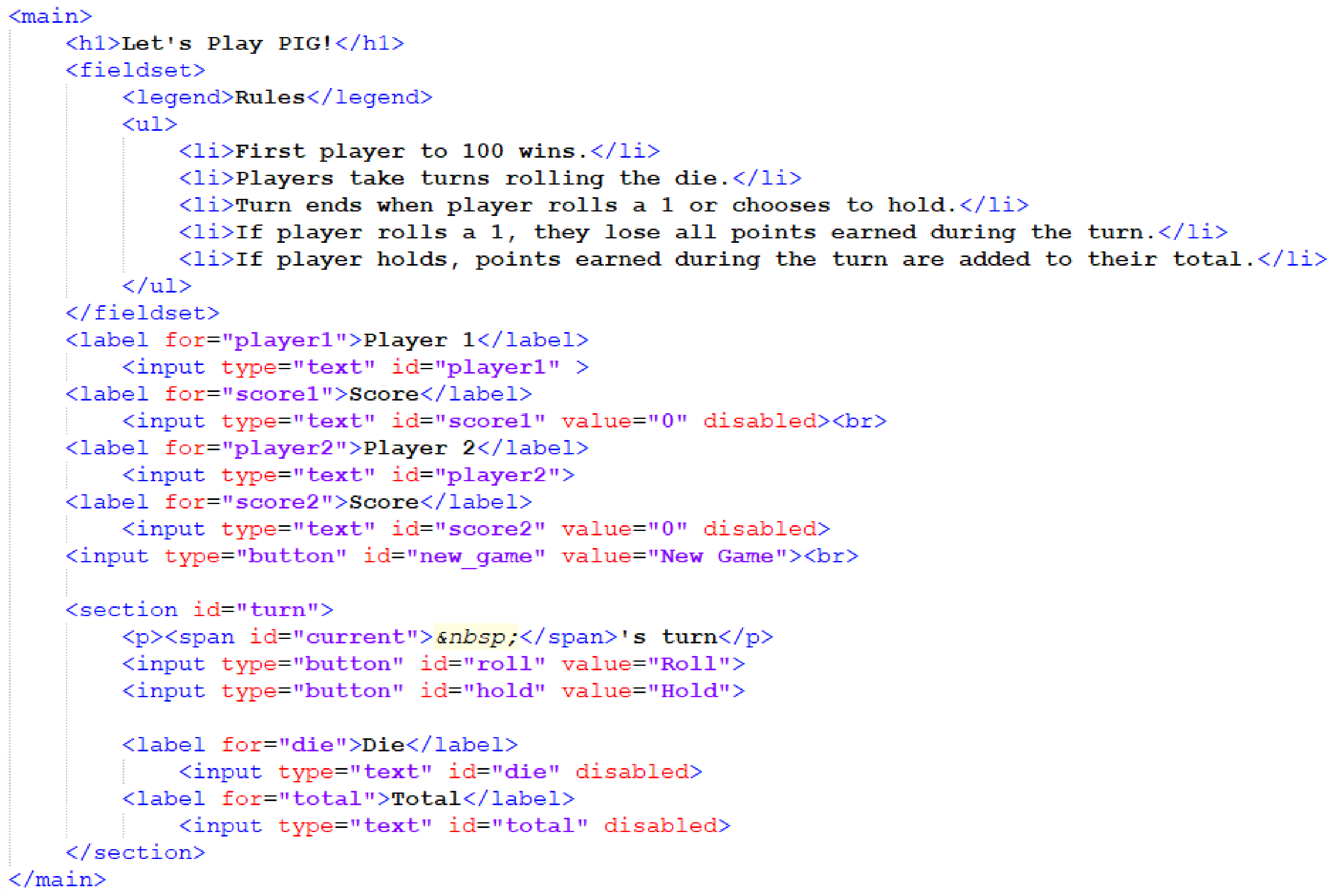
Let's Play PIG!

Rules

- First player to 100 wins.
- Players take turns rolling the die.
- Turn ends when player rolls a 1 or chooses to hold.
- If player rolls a 1, they lose all points earned during the turn.
- If player holds, points earned during the turn are added to their total.

The PIG application

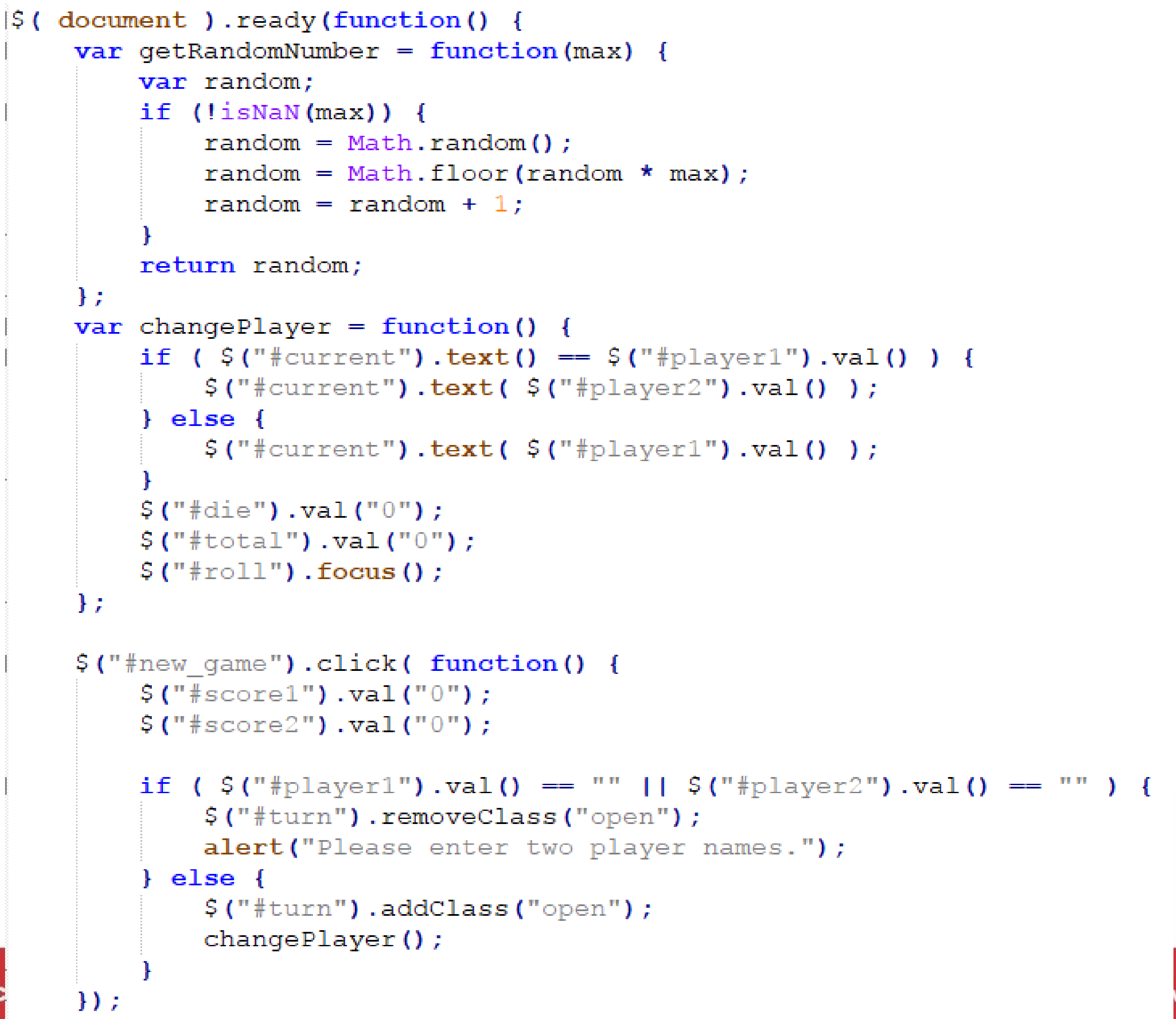
- The HTML code



```html
<main>
    <h1>Let's Play PIG!</h1>
    <fieldset>
        <legend>Rules</legend>
        <ul>
            <li>First player to 100 wins.</li>
            <li>Players take turns rolling the die.</li>
            <li>Turn ends when player rolls a 1 or chooses to hold.</li>
            <li>If player rolls a 1, they lose all points earned during the turn.</li>
            <li>If player holds, points earned during the turn are added to their total.</li>
        </ul>
    </fieldset>
    <label for="player1">Player 1</label>
        <input type="text" id="player1" >
    <label for="score1">Score</label>
        <input type="text" id="score1" value="0" disabled><br>
    <label for="player2">Player 2</label>
        <input type="text" id="player2">
    <label for="score2">Score</label>
        <input type="text" id="score2" value="0" disabled>
    <input type="button" id="new_game" value="New Game"><br>

    <section id="turn">
        <p><span id="current">&nbsp;</span>'s turn</p>
        <input type="button" id="roll" value="Roll">
        <input type="button" id="hold" value="Hold">

        <label for="die">Die</label>
            <input type="text" id="die" disabled>
        <label for="total">Total</label>
            <input type="text" id="total" disabled>
    </section>
</main>
```

The PIG application

- The JavaScript code



```
$( document ).ready(function() {
    var getRandomNumber = function(max) {
        var random;
        if (!isNaN(max)) {
            random = Math.random();
            random = Math.floor(random * max);
            random = random + 1;
        }
        return random;
    };
    var changePlayer = function() {
        if ( $("#current").text() == $("#player1").val() ) {
            $("#current").text( $("#player2").val() );
        } else {
            $("#current").text( $("#player1").val() );
        }
        $("#die").val("0");
        $("#total").val("0");
        $("#roll").focus();
    };

    $("#new_game").click( function() {
        $("#score1").val("0");
        $("#score2").val("0");

        if ( $("#player1").val() == "" || $("#player2").val() == "" ) {
            $("#turn").removeClass("open");
            alert("Please enter two player names.");
        } else {
            $("#turn").addClass("open");
            changePlayer();
        }
    });
```

The PIG application

- The JavaScript code

```javascript
    $("#roll").click( function() {
        var total = parseInt( $("#total").val() );
        var die = getRandomNumber(6);
        if (die == 1) {
            total = 0;
            changePlayer();
        } else { total = total + die; }

        $("#die").val(die);
        $("#total").val(total);
    });
    $("#hold").click( function() {
        var score;
        var total = parseInt( $("#total").val() );
        if ( $("#current").text() == $("#player1").val() ) {
            score = $("#score1");
        } else { score = $("#score2"); }

        score.val( parseInt( score.val() ) + total );
        if (score.val() >= 100) {
            alert( $("#current").text() + " WINS!" );
            newGame();
        } else { changePlayer(); }
    });
});
```

How to work with strings

How to use the properties and methods of the String object

- One property of a String object

Property	Description
`length`	The number of characters in the string

- Example 1: Displaying the length of a string

```
var message_1 = “JavaScript”;
var result_1 = message_1.length; //Result_1 is 10
```

How to use the properties and methods of the String object (cont.)

- Methods of a String object

Method	Description
`charAt(position)`	Return the character at the specific position in the string.
`concat(string1, string2,…)`	Return a new string that concatenation of strings in parameter list.
`indexOf(search,start)`	Return the position of search string if it occurs. If no -1 is returned.
`substr(start,length)`	Return the substring with number character in length parameter and from start position.
`substring(start)`	Return the substring from the start position to end of the string.
`substring(start,end)`	Return the substring from the start position to but not including the end position.
`toLowerCase()`	Return the string with lowercase character.
`toUpperCase()`	Return the string with uppercase character.

How to use the properties and methods of the String object (cont.)

- Example 2: The charAt() method

```
 var message_2 = “JavaScript”;
 var letter = message_2.charAt(4); //letter is “S”
```

- Example 3: The concat() method

```
 var message_3 = “Java”;
 var result_3 = message_3.concat(“Script”);
       //result_3 is “JavaScript”
```

- Example 4: The indexOf() method

```
var result_4a = message_2.indexOf(“a”);   //result is 1
var result_4b = message_2.indexOf(“a”,2); //result is 3
var result_4c = message_2.indexOf(“s”);   //result is -1
```

How to use the properties and methods of the String object (cont.)

- Example 5: The substr() and substring() methods

```
var result_5a = message_2.substr(4,5);  //result is “Scrip”
var result_5b = message_2.subString(4); //result is “Script”
var result_5c = message_2.substring(0.4);//result is “Java”
```

- Example 6: The toLowerCase() and toUppercase() methods

```
var result_6a = message_2.toLowerCase();
         //result is “javascript”
var result_6a = message_2.toUpperCase();
         //result is “JAVASCRIPT”
```

How to work with dates and times

How to create Date objects

- How to create a Date object with current date and time

```
var now = new Date();
```

- How to create a Date object by specifying a date string

```
var electionDay = new Date(“11/6/2018”);
var grandOpening = new Date(“2/16/2017 8:00”);
var departureTime = new Date(“4/6/2017 18:30:00”);
```

- How to create a Date object by specifying date part

Syntax:

```
new Date(year,month,day,hours,minutes,seconds,milliseconds)
```

Example:

```
var electionDay = new Date(2018,10,6);
var grandOpening = new Date(2017,1,16,8);
var departureTime = new Date(2017,3,6,18,30);
```

The methods od the Date object

- The formatting methods of a Date object

Method	Description
`toString()`	Returns a string containing the date and time in local time in the local time using the client’s time zone.
`toDateString()`	Returns a string representing just the date in local time.
`toTimeString()`	Returns a string representing just the time in local time.

- Examples of the formatting methods

```
var birthday = new Date(2017,0,7,8,25); //Jan 7 2017 8:25am
alert(birthday.toString());
          //”Sat Jan 07 2017 08:25:00 GMT +0700”
alert(birthday.toDateString());  //”Sat Jan 07 2017”
alert(birthday.toTimeString());  //”08:25:00 GMT-+0700”
```

The methods od the Date object (cont.)

- The get methods of a Date object

Method	Description
`getTime()`	Returns the number of milliseconds since midnight, Jan 1, 1970.
`getFullYear()`	Returns the four-digit year in local time.
`getMonth()`	Returns month in local time, starting with 0 for January.
`getDate()`	Returns the day of the month in local time.
`getDay()`	Returns the day of the week (1=Sunday, 2=Monday…)
`getHours()`	Returns the hour in 24 hour format.
`getMinutes()`	Returns the minutes in local time
`getSeconds()`	Returns the seconds in local time
`getMilliseconds()`	Returns the milliseconds in local time

The methods od the Date object (cont.)

- The set methods of a Date object

Method	Description
`setFullYear(year)`	Sets the four-digit year in local time.
`setMonth(month)`	Sets the month in local time.
`setDate(day)`	Sets the date of the month in local time.
`setHours(hour)`	Sets the hour in 24-hour format in local time.
`setMinutes(minute)`	Sets the minutes in local time.
`setSeconds(second)`	Sets the seconds in local time.
`setMilliseconds(ms)`	Sets the milliseconds in local time.

Examples of working with dates

- Example 1: How to display the date in your own format

```
var departTime = new Date(2017,3,6,18,30);
var year = departTime.getFullYear();
var month = departTime.getMonth() +1;
var day = departTime.getDate();

var dateText = year + “-”;
if(month<10){
   month =“0” + month;
}
dateText = +=month + “-”;
if(day<10){
   day = “0” + day;
}
dateText += day +”-”; //dateText is “2017-04-06”
```

Examples of working with dates (cont.)

- Example 2: How to calculate a due date

```
var invoiceDate = new Date();
var dueDate new Date(invoiceDate);
dueDate.setDate(dueDate.getDate + 21);
```

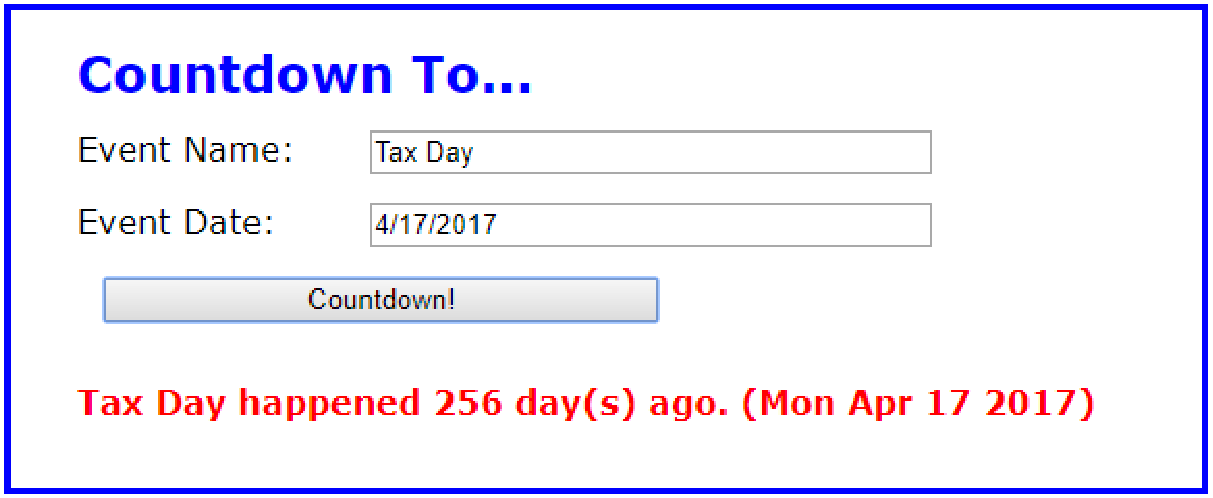
- Example 3: How to find the end of the month

```
var endOfMonth = new Date();

//Set the month to next month
endOfMonth.setMonth(endOfMonth.getMonth() + 1);

//Set the date to one day before the start of the month
endOfMonth.setDate(0);
```

The Count Down application

The Count Down application

- The User Interface



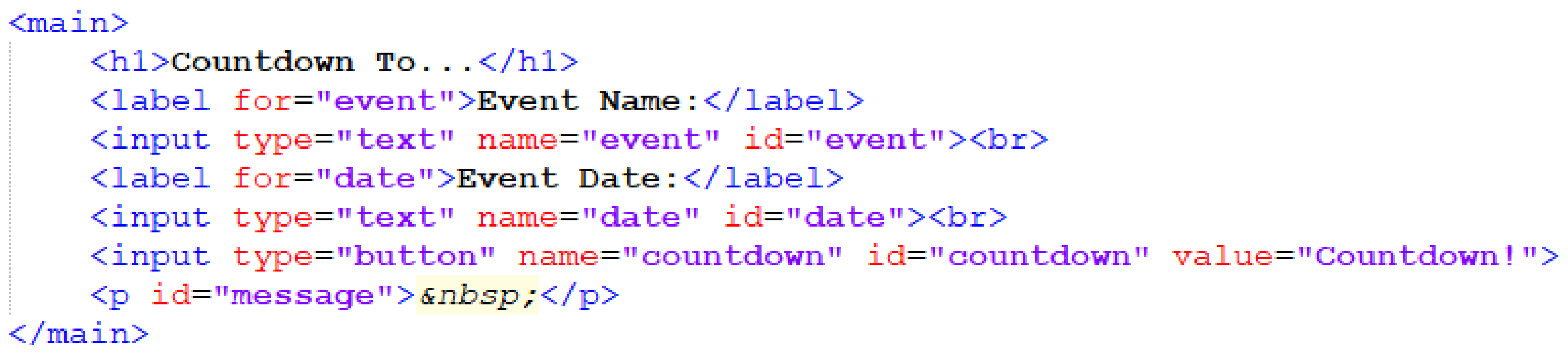
Countdown To...

Event Name:	Tax Day
Event Date:	4/17/2017

Tax Day happened 256 day(s) ago. (Mon Apr 17 2017)

The Count Down application

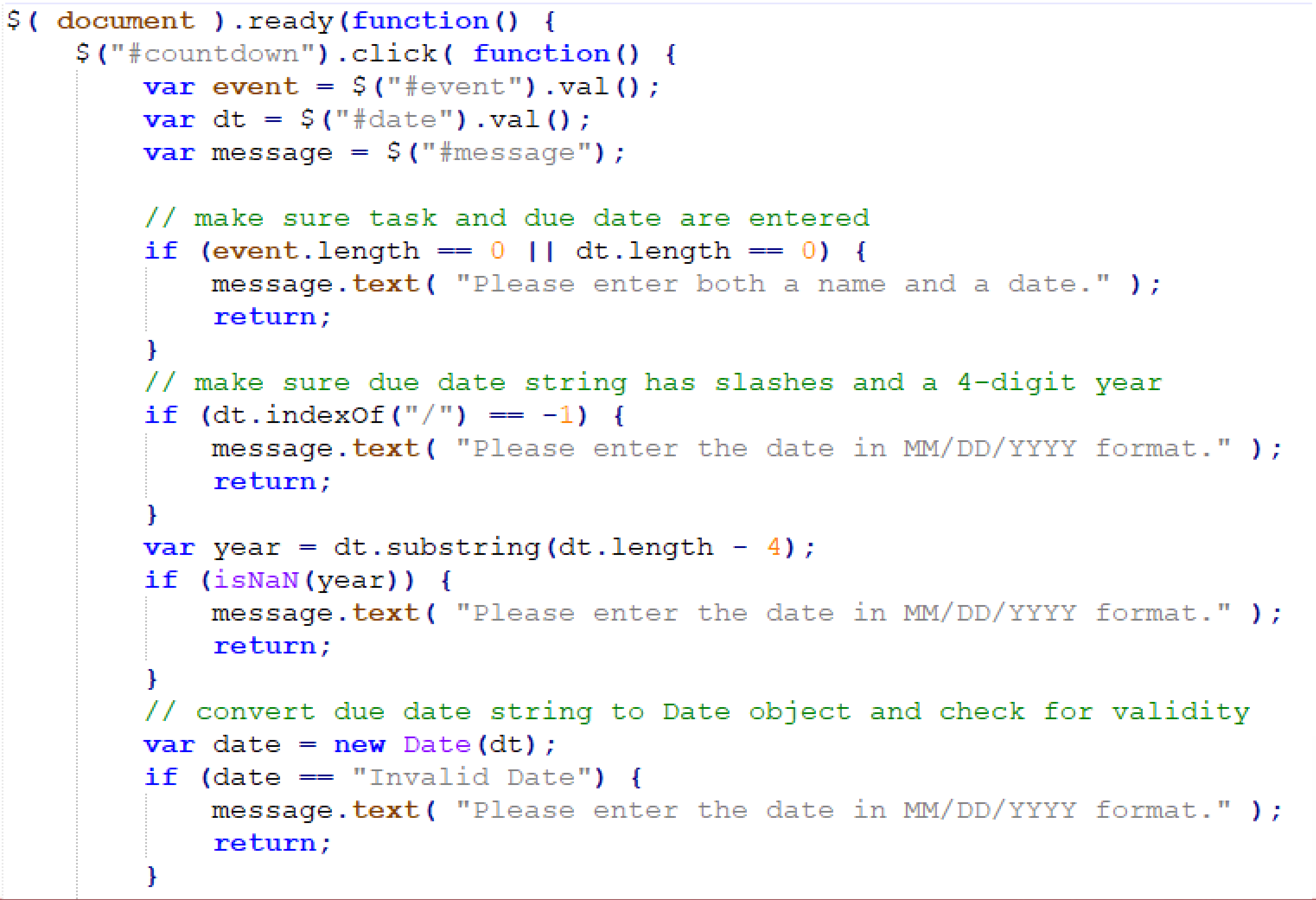
- The HTML code



```html
<main>
    <h1>Countdown To...</h1>
    <label for="event">Event Name:</label>
    <input type="text" name="event" id="event"><br>
    <label for="date">Event Date:</label>
    <input type="text" name="date" id="date"><br>
    <input type="button" name="countdown" id="countdown" value="Countdown!">
    <p id="message">&nbsp;</p>
</main>
```

The Count Down application

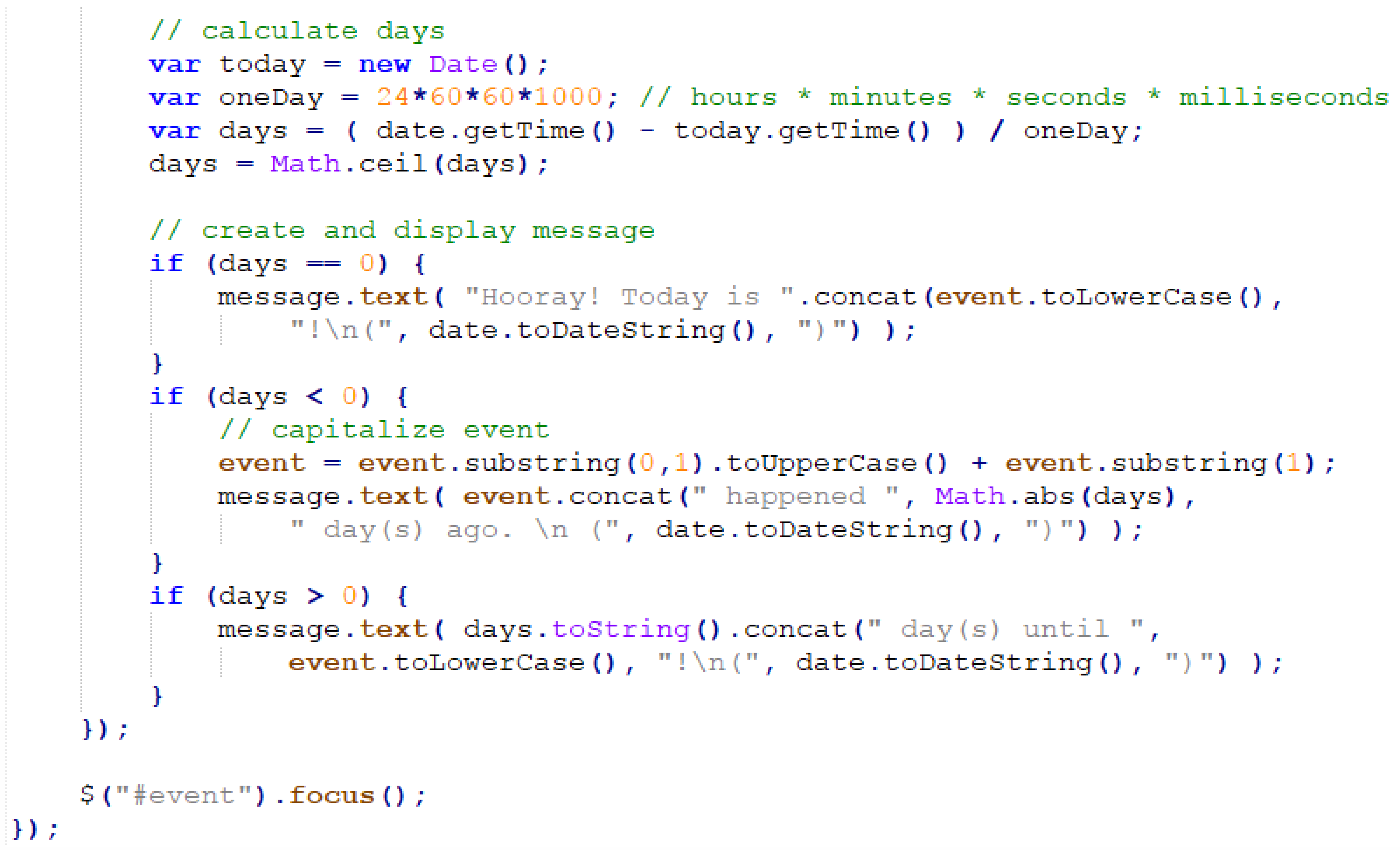
- The JavaScript code



```
$( document ).ready(function() {
    $("#countdown").click( function() {
        var event = $("#event").val();
        var dt = $("#date").val();
        var message = $("#message");

        // make sure task and due date are entered
        if (event.length == 0 || dt.length == 0) {
            message.text( "Please enter both a name and a date." );
            return;
        }
        // make sure due date string has slashes and a 4-digit year
        if (dt.indexOf("/") == -1) {
            message.text( "Please enter the date in MM/DD/YYYY format." );
            return;
        }
        var year = dt.substring(dt.length - 4);
        if (isNaN(year)) {
            message.text( "Please enter the date in MM/DD/YYYY format." );
            return;
        }
        // convert due date string to Date object and check for validity
        var date = new Date(dt);
        if (date == "Invalid Date") {
            message.text( "Please enter the date in MM/DD/YYYY format." );
            return;
        }
```

The Count Down application

- The JavaScript code



```
        // calculate days
        var today = new Date();
        var oneDay = 24*60*60*1000; // hours * minutes * seconds * milliseconds
        var days = ( date.getTime() - today.getTime() ) / oneDay;
        days = Math.ceil(days);

        // create and display message
        if (days == 0) {
            message.text( "Hooray! Today is ".concat(event.toLowerCase(),
                "!\n(", date.toDateString(), ")") );
        }
        if (days < 0) {
            // capitalize event
            event = event.substring(0,1).toUpperCase() + event.substring(1);
            message.text( event.concat(" happened ", Math.abs(days),
                " day(s) ago. \n (", date.toDateString(), ")") );
        }
        if (days > 0) {
            message.text( days.toString().concat(" day(s) until ",
                event.toLowerCase(), "!\n(", date.toDateString(), ")") );
        }
    });

    $("#event").focus();
});
```

Summary

- To working with numeric data, you can use properties and methods of the Number and Math object.
- To working with string data, you can use properties and methods of the String object.
- In JavaScript, dates are stored in Date objects, and they are represented by the number of milliseconds since midnight, Jan 1, 1970.

The End.