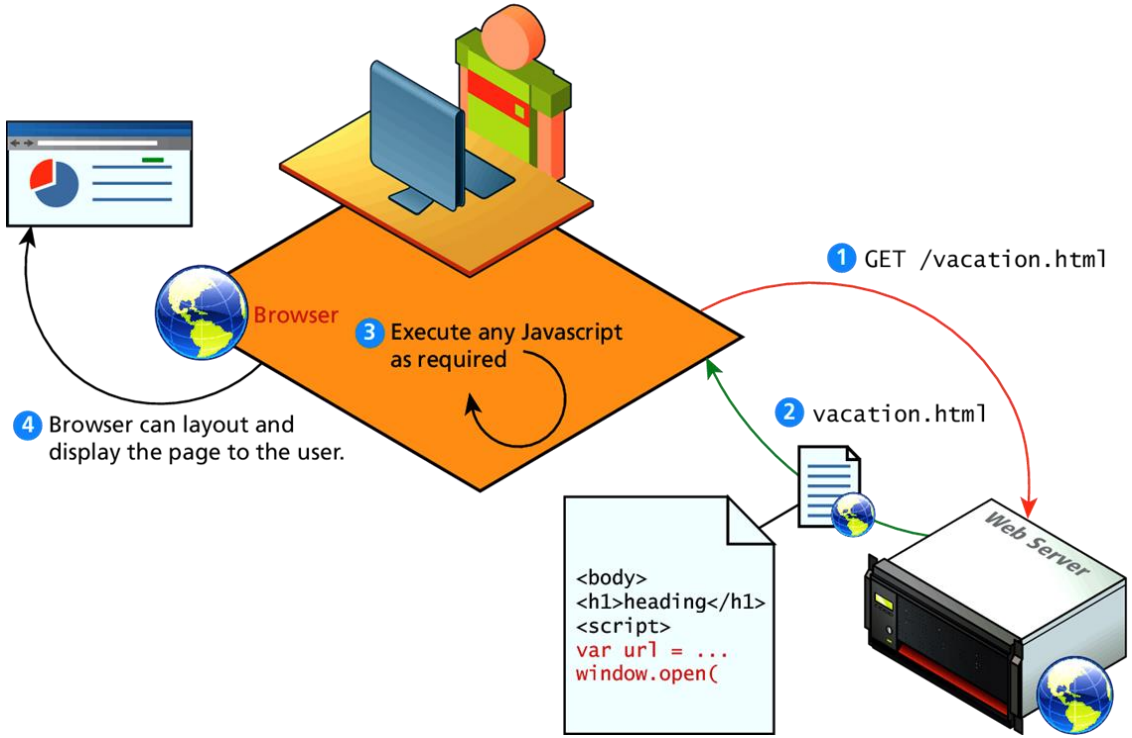
JavaScript: Client-Side Scripting

Dr. Debasish Ghose

University of Agder, Grimstad, Norway

UiA

What is JavaScript



- JavaScript runs right inside the browser
- JavaScript is dynamically typed
- JavaScript is object oriented in that almost everything in the language is an object

Variables

- **Variables** in JavaScript are dynamically typed, meaning a variable can be an integer, and then later a string, then later an object, if so desired.
- 3 ways to declare a JavaScript variable: **var, let, and const**
- var has global scope
- Variables defined with let cannot be redeclared and have Block Scope.
- Variables defined with const cannot be redeclared, reassigned and have Block Scope.

```
var x;      ← a variable x is defined
var y = 0;  ← y is defined and initialized to 0
y = 4;      ← y is assigned the value of 4
```

UiA

Datatypes

Primitive		Reference	
null	let example = null;	**Object**	let example = {hello: "world"};
undefined	let example = undefined;	**Function**	let example = () => 2 + 2;
Boolean	let example = true;	A primitive data type specifies the size and type of variable values, not an object, and it has no additional methods.	
Number	let example = 33;		
String	let example = "hello";		
BigInt	let example = 33n;		
Symbol	let example = Symbol("hello");		

UiA

Operators

Arithmetic

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

- `let x = 5; let y = 2;`
- Addition: `let z = x + y;`
- Substraction: `let z = x - y;`
- Multiplication: `let z = x * y;`
- Division: `let z = x / y;`
- Remainder/Modulus: `let z = x % y;`
- Increment: `x++; let z = x;`
- Decrement: `x--; let z = x;`
- Exponentiation: `let z = x ** 2;`

UiA

Operators
Comparison

Operator	Description	Matches (x=9)
==	Equals	(x==9) is true (x=="9") is true
===	Exactly equals, including type	(x=="9") is false (x===9) is true
< , >	Less than, Greater Than	(x<5) is false
<= , >=	Less than or equal, greater than or equal	(x<=9) is true
!=	Not equal	(4!=x) is true
!==	Not equal in either value or type	(x!=="9") is true (x!==9) is false

UiA

Operators
Logical

- The Boolean operators and, or, and not and their truth tables are listed in Table 6.2. Syntactically they are represented with && (and), || (or), and ! (not).

A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F

AND Truth Table

A	B	A || B
T	T	T
T	F	T
F	T	T
F	F	F

OR Truth Table

A	! A
T	F
F	T

NOT Truth Table

TABLE 6.2 AND, OR, and NOT Truth Tables

Conditionals

If, else if, …, else

- In this syntax the condition to test is contained within ( ) brackets with the body contained in { } blocks.
- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the all conditions are false
- Use else if to specify a new condition to test, if the first condition is false

```
var hourOfDay;   // var to hold hour of day, set it later...
var greeting;   // var to hold the greeting message.
if (hourOfDay > 4 && hourOfDay < 12){
   // if statement with condition
  greeting =  "Good Morning";
}
else if (hourOfDay >= 12 && hourOfDay < 20){
  // optional else if
  greeting =  "Good Afternoon";
}
else{ // optional else branch
  greeting = "Good Evening";
}
```

LISTING 6.4 Conditional statement setting a variable based on the hour of the day

Loops

For Loop (Counted loops) & While Loop (Round and round we go)

- A **for loop** combines the common components of a loop: initialization, condition, and post-loop operation into one statement
- This statement begins with the **for** keyword and has the components placed between **( )** brackets, semicolon (**;**) separated as shown

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

- **While loops** go through a block of code if a specified condition is true.

```
while (condition) {
  // code block to be executed
}
```

```
while (i < 10) {
  text += "The number is " + i;
  i++;
}
```

```
do {
  text += "The number is " + i;
  i++;
}
while (i < 10);
```

UiA

Switch Statement

- **switch statement** to select one of many code blocks to be executed. The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- If there is no match, the default code block is executed.
-
```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

-
```
let x = "0";
switch (x) {
  case 0:
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

-
```
switch (new Date().getDay()) {
  case 0:
    text = "Today is Saturday";
    break;
  case 1:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the
Weekend";
}
```

UiA

Functions

- Functions are the building block for modular code in JavaScript, and are even used to build pseudo-classes, which you will learn about later.
- They are defined by using the reserved word function and then the function name and (optional) parameters.
- Since JavaScript is dynamically typed, functions do not require a return type, nor do the parameters require type.

- Therefore, a function to raise x to the yth power might be defined as:
-
```
function power(x,y){
        var pow=1;
        for (var i=0;i<y;i++){
                pow = pow*x;
        }
        return pow;
}
```

And called as

```
power(2,10);
```

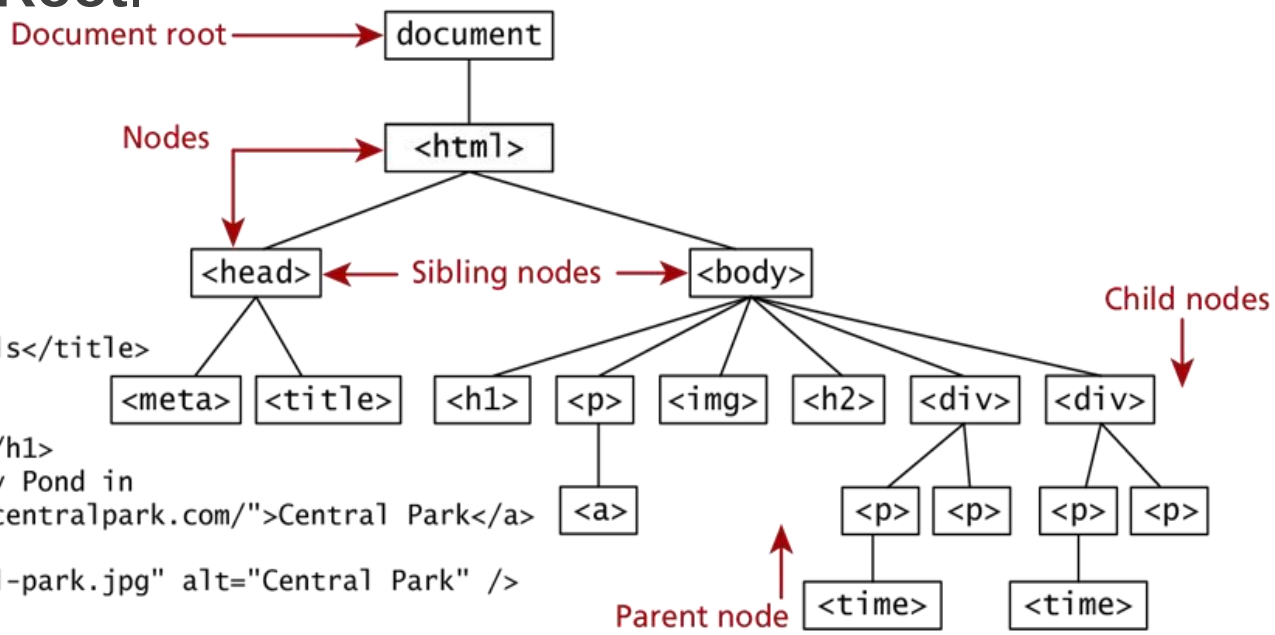
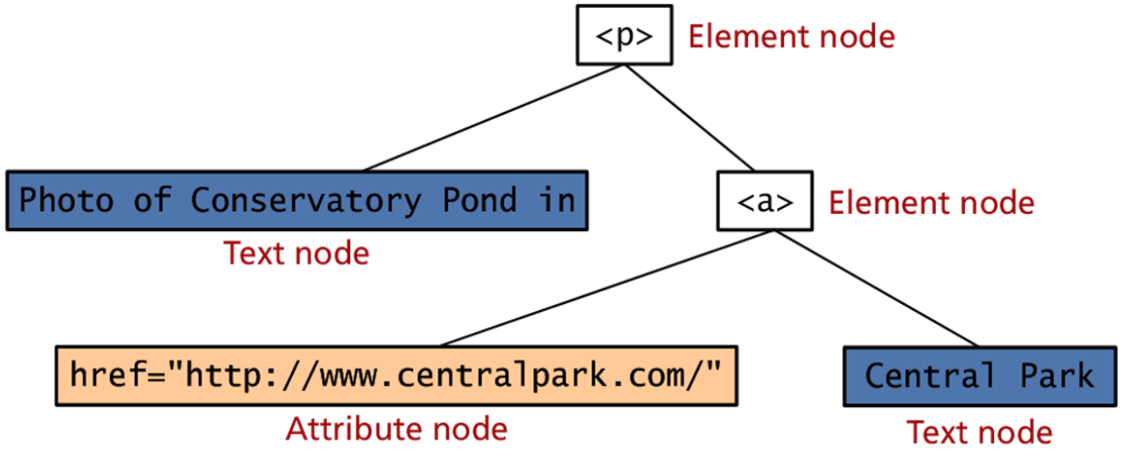
UiA

Objects

Not full-fledged O.O.

- JavaScript is not a full-fledged object-oriented programming language.
- Objects can have properties and methods associated with them
- Methods are actions that can be performed on objects.
- Methods are stored in properties as function definitions.

- ```
  const person = {
    firstName: "John",
    lastName : "Doe",
    id       : 5566,
    fullName : function() {
      return this.firstName + "
  " + this.lastName;
    }
  };
  ```
- Accessing Object Properties:
  - `person.lastName;`
  - `person["lastName"];`
- Accessing Object Methods:
  - `name = person.fullName();`

UiA
  ```

Object Constructors

- A "blueprint" for creating many objects of the same "type". The way to create an "object type", is to use an object constructor function. In the example below, function **`Person()`** is an object constructor function. Objects of the same type are created by calling the constructor function with the new keyword:
- Example:
  - ```
    function Person(first, last, age, eye) {
      this.firstName = first;
      this.lastName = last;
      this.age = age;
      this.eyeColor = eye;
    }
    ```
  - ```
    const myFather = new Person("John", "Doe", 50, "blue");
    const myMother = new Person("Sally", "Rally", 48, "green");
    ```

Objects Included in JavaScript

A number of useful objects are included with JavaScript including:

- Array
- Boolean
- Date
- Math
- String
- Dom objects

Arrays

- An array is a special variable, which can hold more than one value. That is, it can hold many values under a single name, and you can access the values by referring to an index number.
- The easiest way to add a new element to an array is using the push() method.
- Arrays use numbered indexes whereas objects use named indexes.

- `const cars = ["Saab", "Volvo", "BMW"];`
- `const cars = new Array("Saab", "Volvo", "BMW");`
- Accessing Array Elements:
  - `let car = cars[0];`
- Array Properties and Methods:
  - `cars.length // Returns the number of elements`
    `cars.sort() // Sorts the array`
- Adding Array Elements:
  - `const fruits = ["Banana", "Orange", "Apple"];`
    `fruits.push("Lemon"); // Adds a new element (Lemon) to fruits`

UiA

Date Objects

- By default, JavaScript will use the browser's time zone and display a date as a full text string: Tue Nov 02 2021 11:08:31 GMT+0100 (Central European Standard Time)
- When a Date object is created, a number of methods allow you to operate on it.

- Creating Date Objects
  - `new Date()`
    `new Date(year, month, day, hours, minutes, seconds, milliseconds)`
    `new Date(milliseconds)`
    `new Date(date string)`
- For Instance: `const d = new Date(2018, 11, 24, 10, 33, 30, 0);`
- Date Methods:
  - `const d = new Date();`
    `d.toDateString();`

UiA

The DOM

- The Document Object Model (DOM) is the objects that make up a web page. This tree structure is formally called the **DOM Tree** with the root, or topmost object called the **Document Root**.




```
<html>
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels</title>
</head>
<body>
   <h1>Share Your Travels</h1>
   <p>Photo of Conservatory Pond in
      <a href="http://www.centralpark.com/">Central Park</a>
   </p>
   <img src="images/central-park.jpg" alt="Central Park" />
   <h2>Reviews</h2>
   <div id="latestComment">
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>
</body>
</html>
```

UiA

DOM Nodes

Element, text and attribute nodes

```
<p>Photo of Conservatory Pond in
   <a href="http://www.centralpark.com/">Central Park</a>
</p>
```



Property	Description
attributes	Collection of node attributes
childNodes	A NodeList of child nodes for this node
firstChild	First child node of this node.
lastChild	Last child of this node.
nextSibling	Next sibling node for this node.
nodeName	Name of the node
nodeType	Type of the node
nodeValue	Value of the node
parentNode	Parent node for this node.
previousSibling	Previous sibling node for this node.

UiA

Accessing nodes

getElementById(), getElementsByTagName()

```
var abc = document.getElementById("latestComment");
```

```
<body>
   <h1>Reviews</h1>
   <div id="latestComment">
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <hr/>
   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>
   <hr/>
</body>
```

```
var list = document.getElementsByTagName("div");
```

UiA

Modifying a DOM element

- Using the DOM document and HTML DOM element objects, we can do exactly that using the innerHTML property
- <script>

```
var latest = document.getElementById("latestComment");
var oldMessage = latest.innerHTML;
latest.innerHTML = oldMessage + "<p>Updated this div with JS</p>";
```

LISTING 6.8 Changing the HTML using innerHTML

</script>

UiA

JavaScript Events

- A JavaScript event is an action that can be detected by JavaScript.
- We say then that an event is triggered and then it can be caught by JavaScript functions, which then do something in response.

- ```
  <!DOCTYPE html>
  <html>
  <body>

  <h1 onclick="this.innerHTML =
  'Ooops!'">Click on this
  text!</h1>

  </body>
  </html>
  ```

UiA
  ```

Event Listener

- The addEventListener() method attaches an event handler to the specified element.
- The addEventListener() method attaches an event handler to an element without overwriting existing event handlers.
- You can add many event handlers to one element.

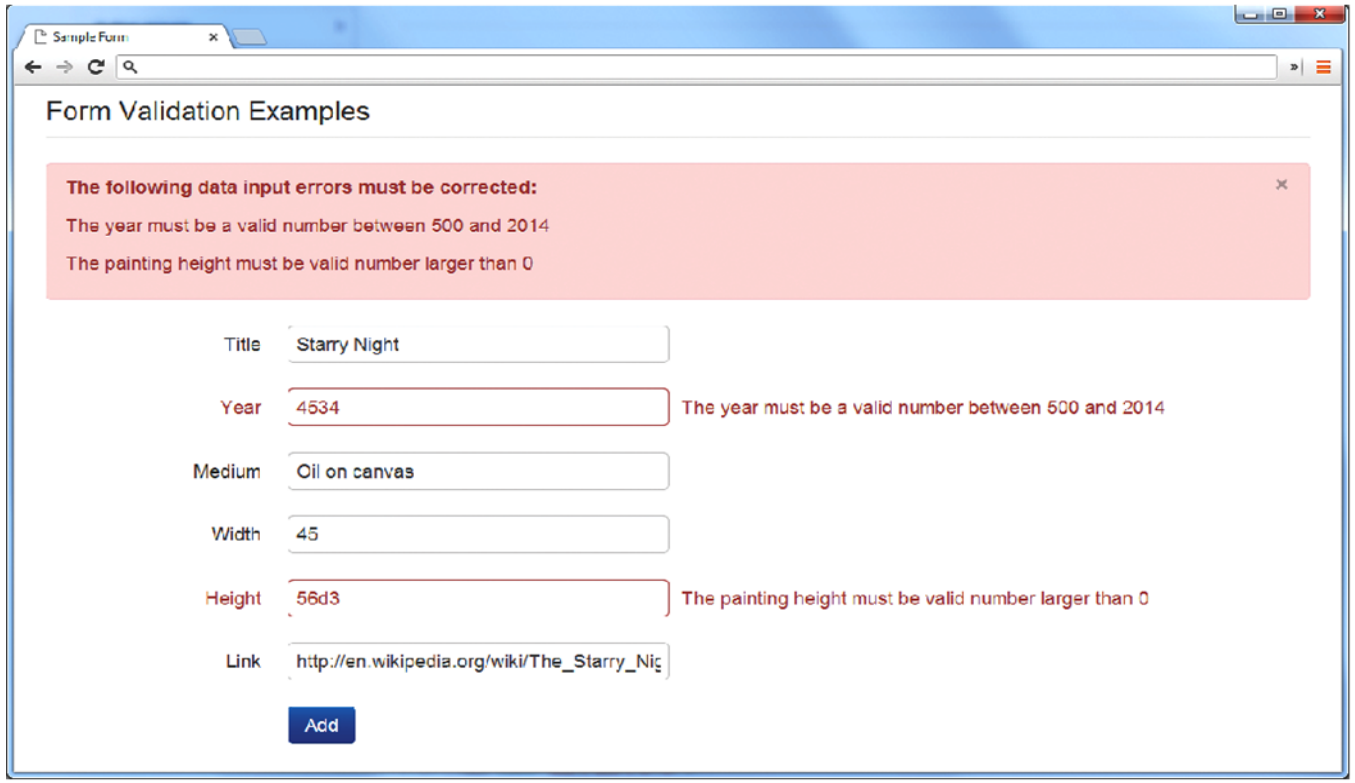
- Syntax:
  ```
  element.addEventListener(event, function);
  ```
- ```
  document.getElementById("myBtn")
  .addEventListener("click", displayDate);
  ```
- ```
  element.addEventListener("click" , myFunction);

  function myFunction() {
    alert ("Hello World!");
  }
  ```

UiA

Forms

Validating forms

- Writing code to pre-validate forms on the client side will reduce the number of incorrect submissions, thereby reducing server load.
- There are several common validation activities including email validation, number validation, and data validation.



UiA

jQuery

- jQuery was created in 2006 by John Resig. It was designed to handle Browser Incompatibilities and to simplify HTML DOM Manipulation, Event Handling, Animations, and Ajax.
- For more than 10 years, jQuery has been the most popular JavaScript library in the world.

- jQuery: `myElement = $("#id01");`
- `JavaScript: myElement = document.getElementById("Id01");`

- `myElements = $("p");`
- `myElements = document.getElementsByTagName("p");`

UiA