WEB1100B- WEBUTVIKLING OG HCI

Lecture 9– Introduksjon til Javascript og jQuery

Anh Nguyen Duc

LÆRINGSMÅL LEKSJON

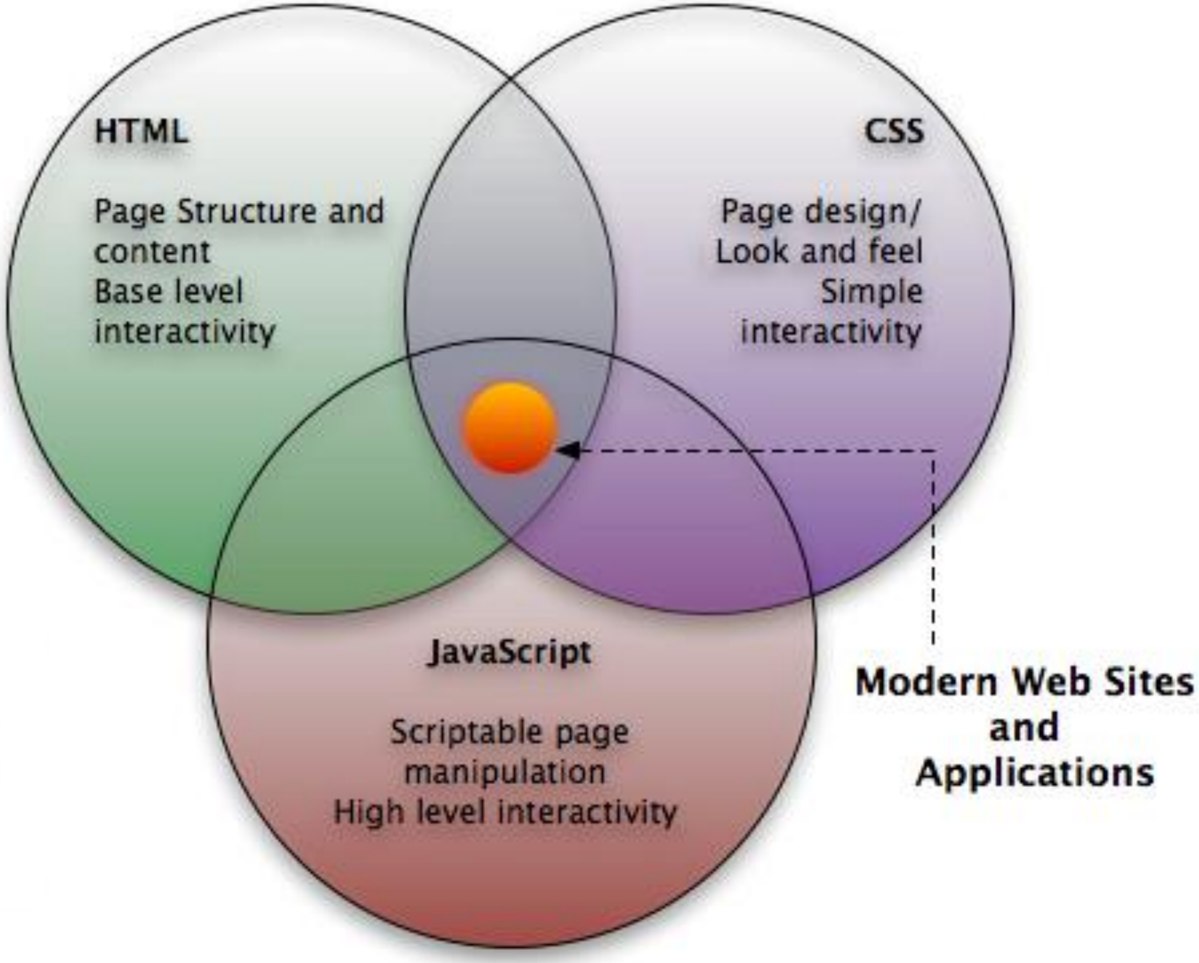
Dere skal lære dere noe om

- What JavaScript is
- Why you want to use JavaScript
- The <script> tag
- An overview of JavaScript language
- Use JavaScript to handle browser events

WHAT JAVASCRIPT IS?

- High-level programming language
- One of the three core technology for modern web development
- Event-driven model
- Handling browser event without interacting with the server

Java ≠ JavaScript



HTML

Page Structure and content
Base level interactivity

CSS

Page design/
Look and feel
Simple interactivity

JavaScript

Scriptable page manipulation
High level interactivity

Modern Web Sites and Applications

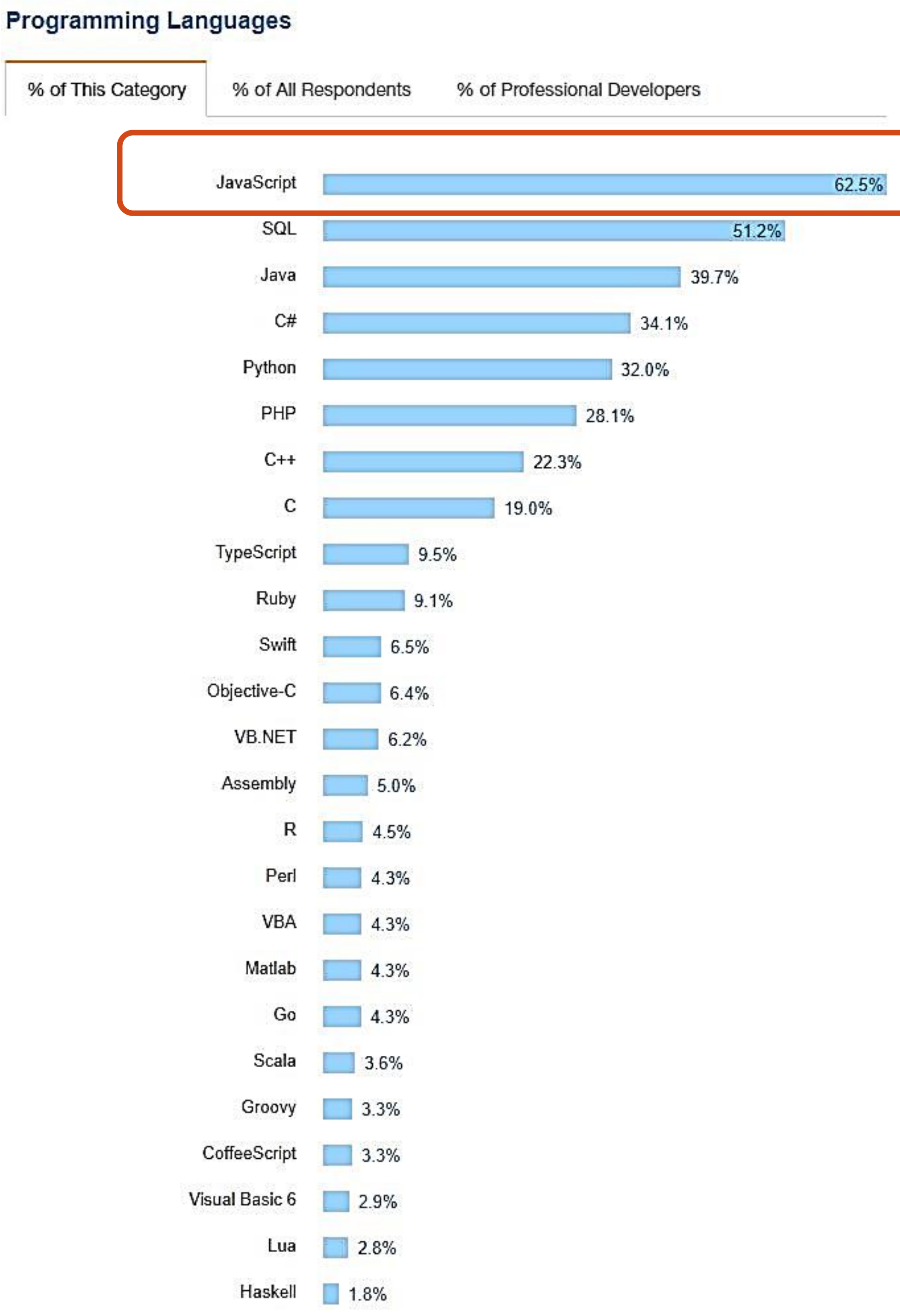
WHY WE USE JAVASCRIPTS

- Easy to use
- Improving performance
- Integration with web browsers
- Javascript now used everywhere
    - Angular, React framework for client side
    - NodeJS for server side

stackoverflow

Developer Survey Results

2017 Fra 64.000 developers



Programming Languages

THE <SCRIPT> TAG

```
<script type="text/javascript">
  Javascript code …
</script>
```

```
<script type="text/javascript" src="javascript.js"></script>
```

```
<script type="text/javascript"
src="http://www.example.com/script.js"></script>
```

THE STRUCTURE OF JAVASCRIPT

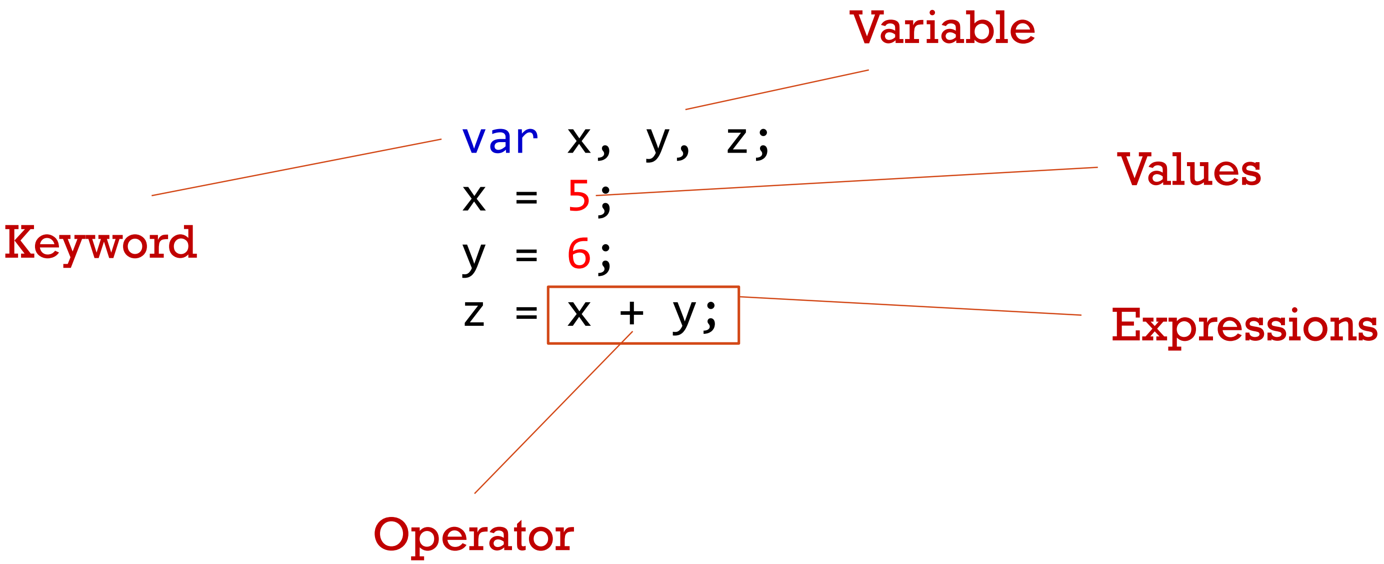
```
<html>
<head>
<title></title>
</head>
<body>
            <!– some comments -->
            Your webcontent goes here
</body>
<script type="text/javascript">
            //some comments here
            Your javascript code goes here
</script>
</html>
```

HELLO WORLD

```
<!DOCTYPE HTML>
<html>
<body>
  <p>Before the script...</p>
  <script>
    alert( 'Hello, world!' );
  </script>
  <p>...After the script.</p>
</body>
</html>
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE

- Statement: seperated by semicolons
- JavaScript statements are composed of Values, Operators, Expressions, Keywords, and Comments.



```
var x, y, z;
x = 5;
y = 6;
z = x + y;
```

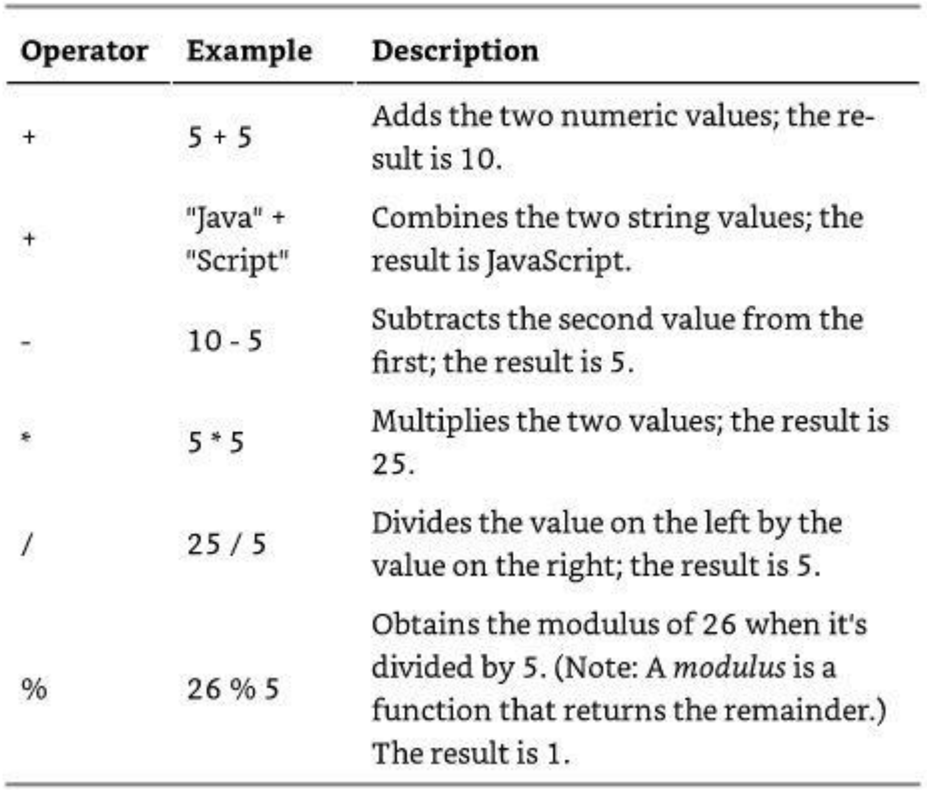
Variable

Keyword

Values

Expressions

Operator



Operator	Example	Description
+	5 + 5	Adds the two numeric values; the result is 10.
+	"Java" + "Script"	Combines the two string values; the result is JavaScript.
-	10 - 5	Subtracts the second value from the first; the result is 5.
*	5 * 5	Multiplies the two values; the result is 25.
/	25 / 5	Divides the value on the left by the value on the right; the result is 5.
%	26 % 5	Obtains the modulus of 26 when it's divided by 5. (Note: A *modulus* is a function that returns the remainder.) The result is 1.

JAVASCRIPT AS A PROGRAMMING LANGUAGE (2)

```
var message = "message" ;
message
var sum;
sum = 5+5;
var firstName = "John"
var lastName = "Kennedy"
var name = firstName + " " + lastName;
```

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

JAVASCRIPT AS A PROGRAMMING LANGUAGE (3)

```
var 1stvariable;

var firstVariable, firstvariable;

var $mydollar;

var true;

var while;
```

•Names can only contain letters, digits, underscores, and dollar signs.
•Names must begin with a letter
•Names can also begin with $ and _ (but we will not use it in this tutorial)
•Names are case sensitive (y and Y are different variables)
•Reserved words (like JavaScript keywords) cannot be used as names

JAVASCRIPT AS A PROGRAMMING LANGUAGE (3)

- IF statement

```
if (condition) {
    block of code to be executed if
the condition is true
}
```

```
 <script type="text/javascript">
       var age = 20;
       if( age > 18 ){
           document.write("<b>Lets drink!</b>");
        }
</script>
```

Note that **if** is in lowercase letters. Uppercase letters (If or IF) will generate a JavaScript error.

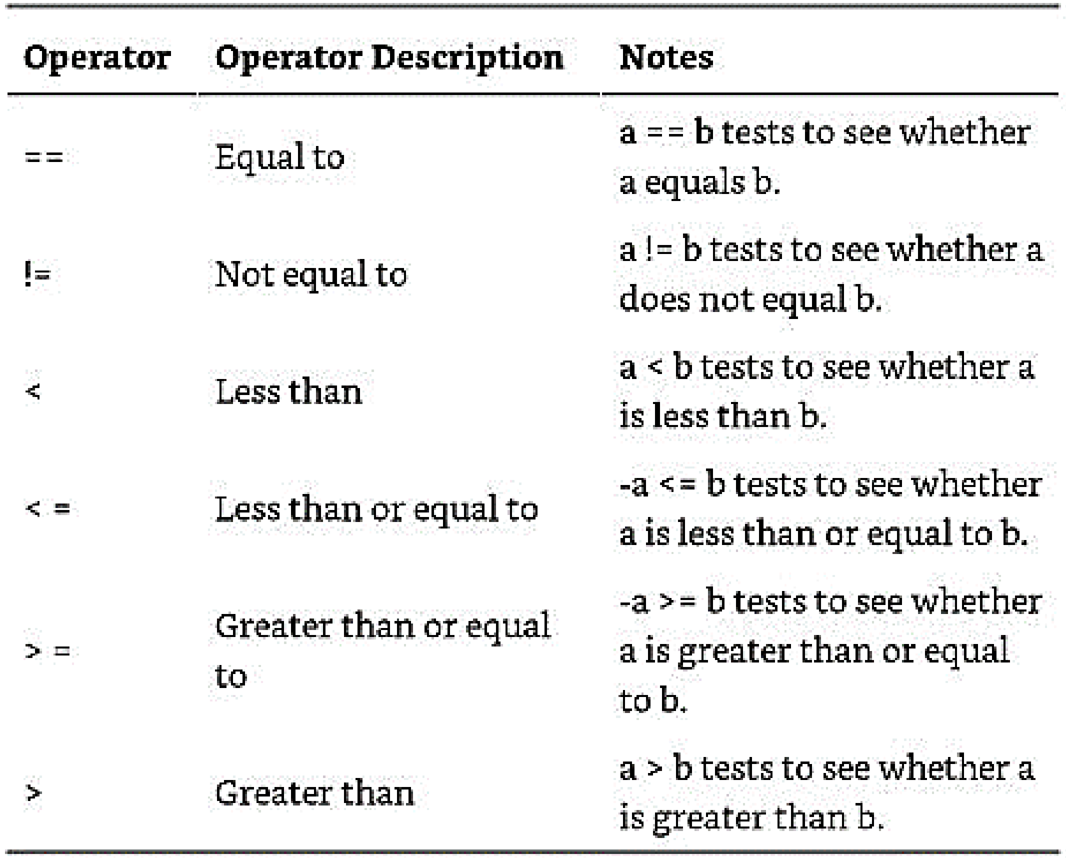
JAVASCRIPT AS A PROGRAMMING LANGUAGE (3)

- IF statement

```
if (condition) {
       block of code to be executed if
       the condition is true
}
else {
       block of code to be executed if
       the condition is false
}
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (4)

- IF statement



Operator	Operator Description	Notes
==	Equal to	a == b tests to see whether a equals b.
!=	Not equal to	a != b tests to see whether a does not equal b.
<	Less than	a < b tests to see whether a is less than b.
< =	Less than or equal to	-a <= b tests to see whether a is less than or equal to b.
> =	Greater than or equal to	-a >= b tests to see whether a is greater than or equal to b.
>	Greater than	a > b tests to see whether a is greater than b.

```
<script type="text/javascript">
      var age = 20;
      if( age > 18 ){
          document.write("<b>Lets drink!</b>");
       }
</script>
```

Comparison Operator

1 == 2 .... False

5 > 3 .... True

JAVASCRIPT AS A PROGRAMMING LANGUAGE (5)

- Loop statement

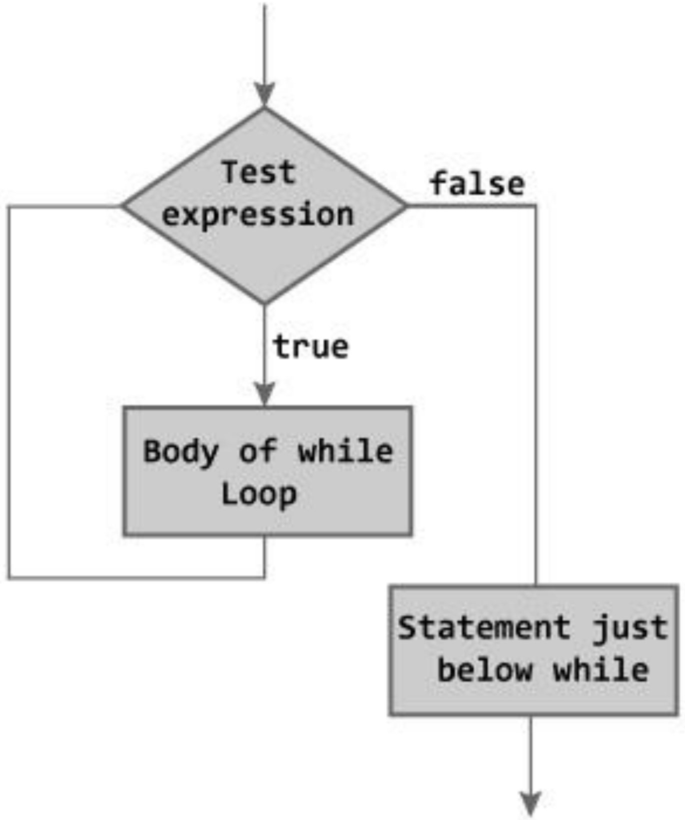
```
text += "The number is 1 <br>";
text += "The number is 2 <br>";
text += "The number is 3 <br>";
text += "The number is 4 <br>";
text += "The number is 5 <br>";
text += "The number is 6 <br>";
text += "The number is 7 <br>";
text += "The number is 8 <br>";
text += "The number is 9 <br>";
```

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

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

JAVASCRIPT AS A PROGRAMMING LANGUAGE (6)

- Loop statement



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

✓ Forget to initial variables used in the loop
✓ Never finish loop
✓ Use the wrong comparison in the condition
✓ Include statements inside the loop that belong outside the loop

JAVASCRIPT AS A PROGRAMMING LANGUAGE (7)

- Loop statement (Iteration)

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


for (var i = 0; i < 10; i++) {
    text += "The number is " + i + " <br>";
}
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (8)

- Function declaration

```
function myFunction(p1, p2) {
    return p1 * p2;
    // The function returns the product of p1 and p2
}

function name(parameter1, parameter2, parameter3) {
    code to be executed
}
```

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

- Function invocation
  - When an event occurs
  - When it is called from other Javascript code
  - Automatically (Self invoked)

JAVASCRIPT AS A PROGRAMMING LANGUAGE (9)

- Function declaration

```
function myFunction(p1, p2) {
    return p1 * p2;
    // The function returns the product of p1 and p2
}

function name(parameter1, parameter2, parameter3) {
    code to be executed
}
```

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

- Function Return
  - The function stops executing when reaching a return statement
  - Function often return a value

```
var x = myFunction(4, 3);
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (10)

- Data Types
  - A classification of data tells how they should be used
  - Define the operations that can be done on the data

```
var x = 16 + "Volvo";
```

```
var x = "16" + "Volvo";
```

```
var foo = 42;    // foo is now a Number
var foo = 'bar'; // foo is now a String
var foo = true;  // foo is now a Boolean
var foo = null;  // foo is now a null
```

Primitive type

JAVASCRIPT AS A PROGRAMMING LANGUAGE (11)

- Data Types
  - A classification of data tells how they should be used
  - Define the operations that can be done on the data

Javascript attempts to convert data to whatever type it needs in a given context (autocasting)

```
var x = 16 + "Volvo";

var x = "16" + "Volvo";
```

```
var foo = 42;    // foo is now a Number
    foo = 'bar'; // foo is now a String
    foo = true;  // foo is now a Boolean
    foo = null;  // foo is now a null
```

Primitive type

JAVASCRIPT AS A PROGRAMMING LANGUAGE (12)

- Array

Initialize an array as an Object:

```
var nums = new Array(10);
var cars = new Array("Saab", "Volvo", "BMW");


…


nums[0] = “Number 1”;
var tourname = " Second car: " + cars[1];
```

Initialize an array as a variable:

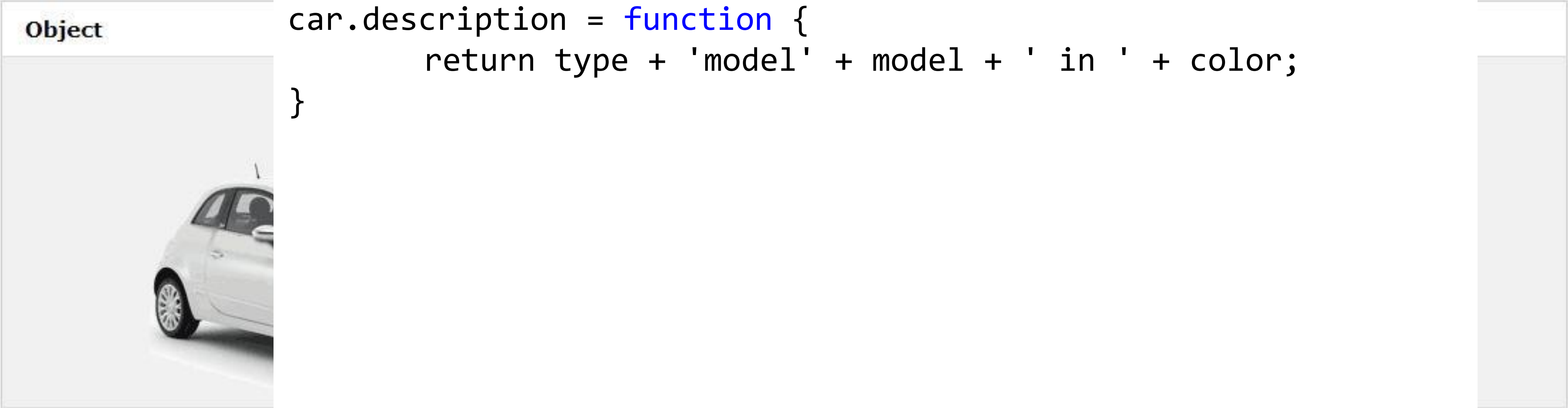
```
var cars = ["Saab", "Volvo", "BMW"];
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (13)

- Object

```
var car = {type:"Fiat", model:"500", color:"white"};

car.description = function {
        return type + 'model' + model + ' in ' + color;
}
```



Object

https://www.w3schools.com/js/js_objects.asp

JAVASCRIPT AS A PROGRAMMING LANGUAGE (14)

- Here are some examples of HTML events:
  - An HTML web page has finished loading
  - An HTML input field was changed
  - An HTML button was clicked
- Binding a javascript code to events

```
<button onclick="document.getElementById('demo').innerHTML = Date()">
       The time is?
</button>
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (14)

- Binding to HTML elements' attribute
  - Cons: hard to maintain
- Binding to HTML elements' property
  - Cons: bind only one function to the event

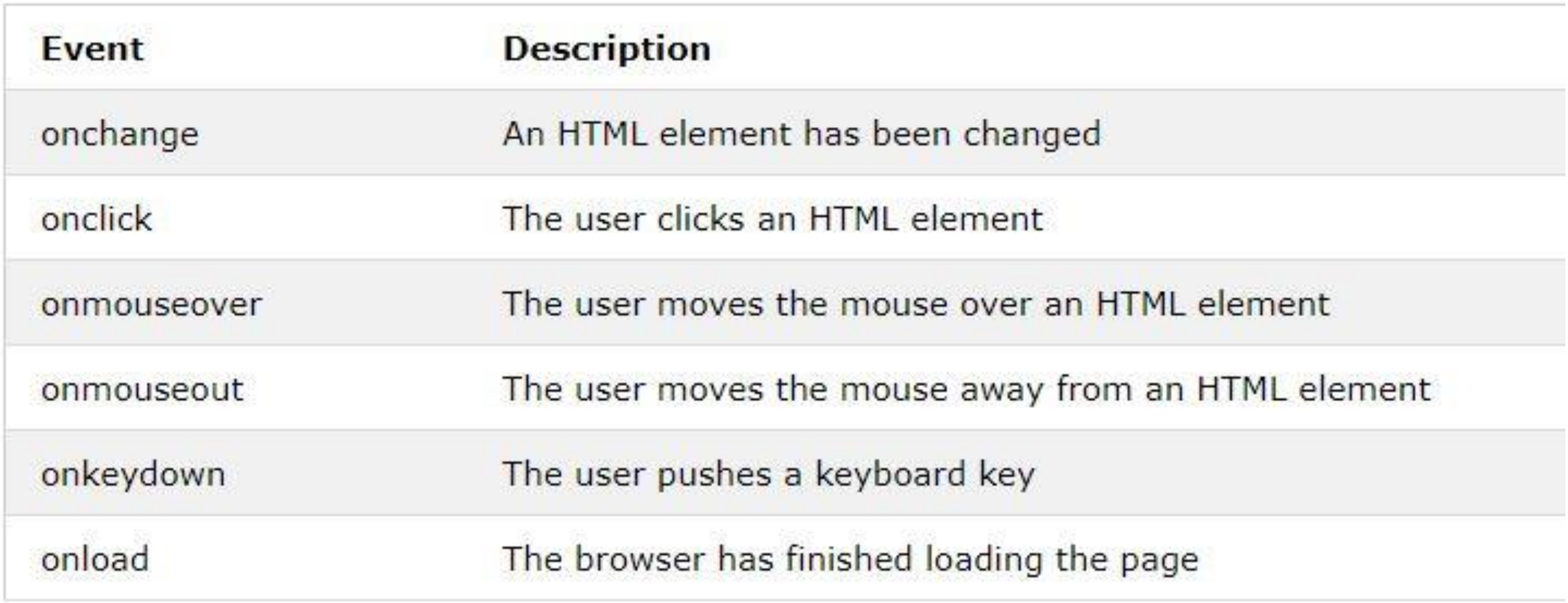
```
myLink.onclick = function(event) { alert('Be careful'); }
```

- Use addEventListener()

```
myButton.addEventListener('click', function(){ alert('Be careful!');
}, false);
```

JAVASCRIPT AS A PROGRAMMING LANGUAGE (15)

- Binding to HTML elements' attribute
  - Cons: hard to maintain



Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Mer: https://www.elated.com/articles/events-and-event-handlers/

USING JQUERY

WHAT ARE JAVASCRIPT LIBRARIES

- Pre-written JavaScript which allows for easier development of JavaScript
- Your web can "link" to a library to allow you access to that functionality
- jQuery is a popluar Javascript library
  - wrap a lot of common tasks into methods that you can call with a single line of code
  - simplifies complicated things from JavaScript, like AJAX calls and DOM manipulation

jQuery

write less, do more.

https://jquery.com/

HOW TO CALL A JQUERY LIBRARY FUNCTIONS?

```
$(document).ready(function() { // do stuff when DOM is ready  })
```

```
<html>
<head>
      <title>The jQuery Example</title>
      <script type = "text/javascript"
         src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
      </script>
      <script type = "text/javascript" language = "javascript">
         $(document).ready(function() {
            $("div").click(function() {alert("Hello, world!");});
         });
      </script>
</head>
<body>
   <div id = "mydiv"> Click on this to see a dialogue box. </div>
</body>
</html>
```

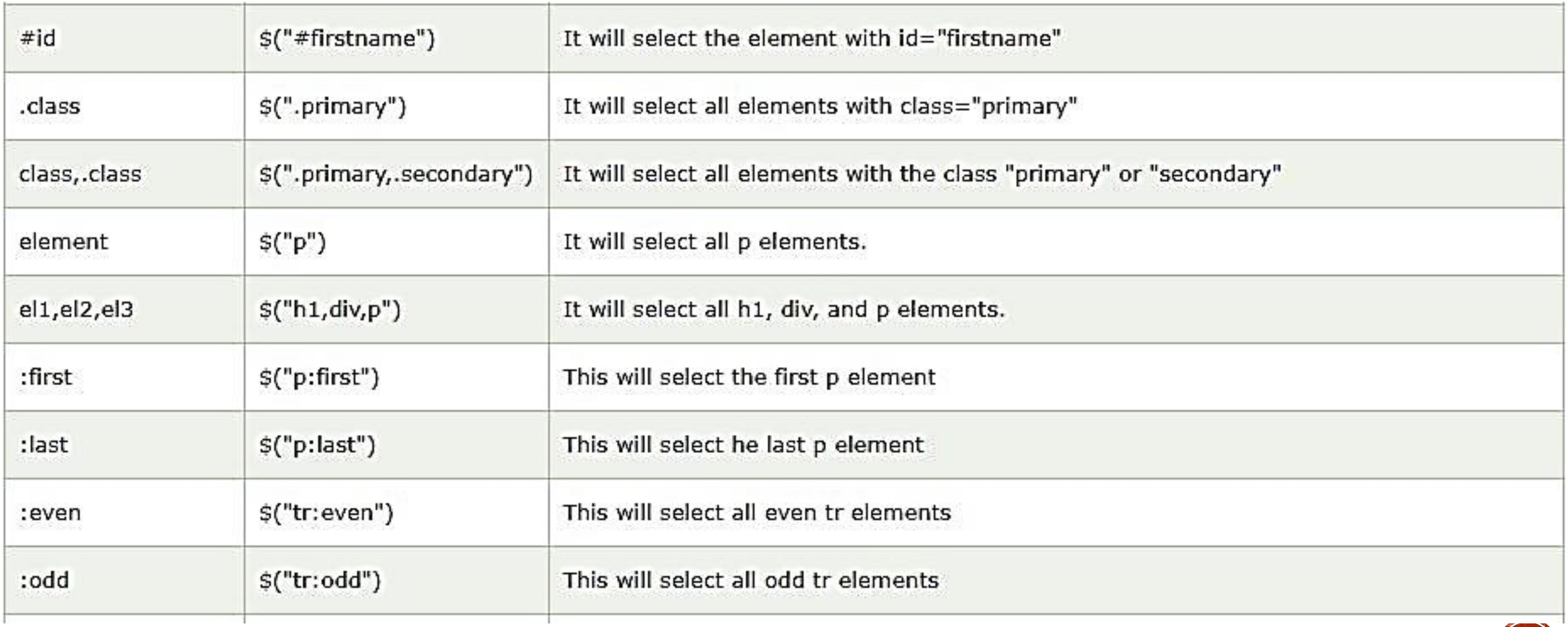
SELECTING ELEMENTS FROM THE DOCUMENT

- All starts with $( )
- It uses the three basic building blocks while selecting an element
  - Element name: $(‘p’)
  - Element ID: $(‘#myid’)
  - Element class: $(‘.class’)

SELECTING ELEMENTS FROM THE DOCUMENT

```
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="http://ajax.googleapis.com/
    ajax/libs/jquery/2.1.3/jquery.min.js">
     </script>
     <script type="text/javascript" language="javascript">
         $(document).ready(function() {
         $("p").css("background-color", "pink");
         });
     </script>
 </head>
<body>
    <p>This is first paragraph.</p>
    <p>This is second paragraph.</p>
    <p>This is third paragraph.</p>
</body>
</html>
```

DIFFERENT JQUERY SELECTORS



#id	$("#firstname")	It will select the element with id="firstname"
.class	$(".primary")	It will select all elements with class="primary"
class,.class	$(".primary,.secondary")	It will select all elements with the class "primary" or "secondary"
element	$("p")	It will select all p elements.
el1,el2,el3	$("h1,div,p")	It will select all h1, div, and p elements.
:first	$("p:first")	This will select the first p element
:last	$("p:last")	This will select he last p element
:even	$("tr:even")	This will select all even tr elements
:odd	$("tr:odd")	This will select all odd tr elements

Mer fra: https://www.javatpoint.com/jquery-selectors

MODIFYING STYLES ON A PAGE

- Hide and show elements:

```
$("#hide").click(function(){
    $("p").hide();
});
$("#show").click(function(){
    $("p").show();
});
```

- Retrieving and changing style sheet properties

```
$(this).css(“text-decoration“, “underline“);
```

THIS IN JAVASCRIPT

```
 $ ("button").click (function (event) {
    // $(this) will have the value of the button
($("button")) object
// because the button object invokes the click () method
    console.log ($ (this).prop ("name"));
});

var person = {
    firstName   :"Penelope",
    lastName    :"Barrymore",
    // "this" will have the value of the person object because the person
object will invoke    showFullName ()
    showFullName:function () { console.log (this.firstName + " " +
this.lastName);   }
}
```

WORKING WITH FORM