JavaScript

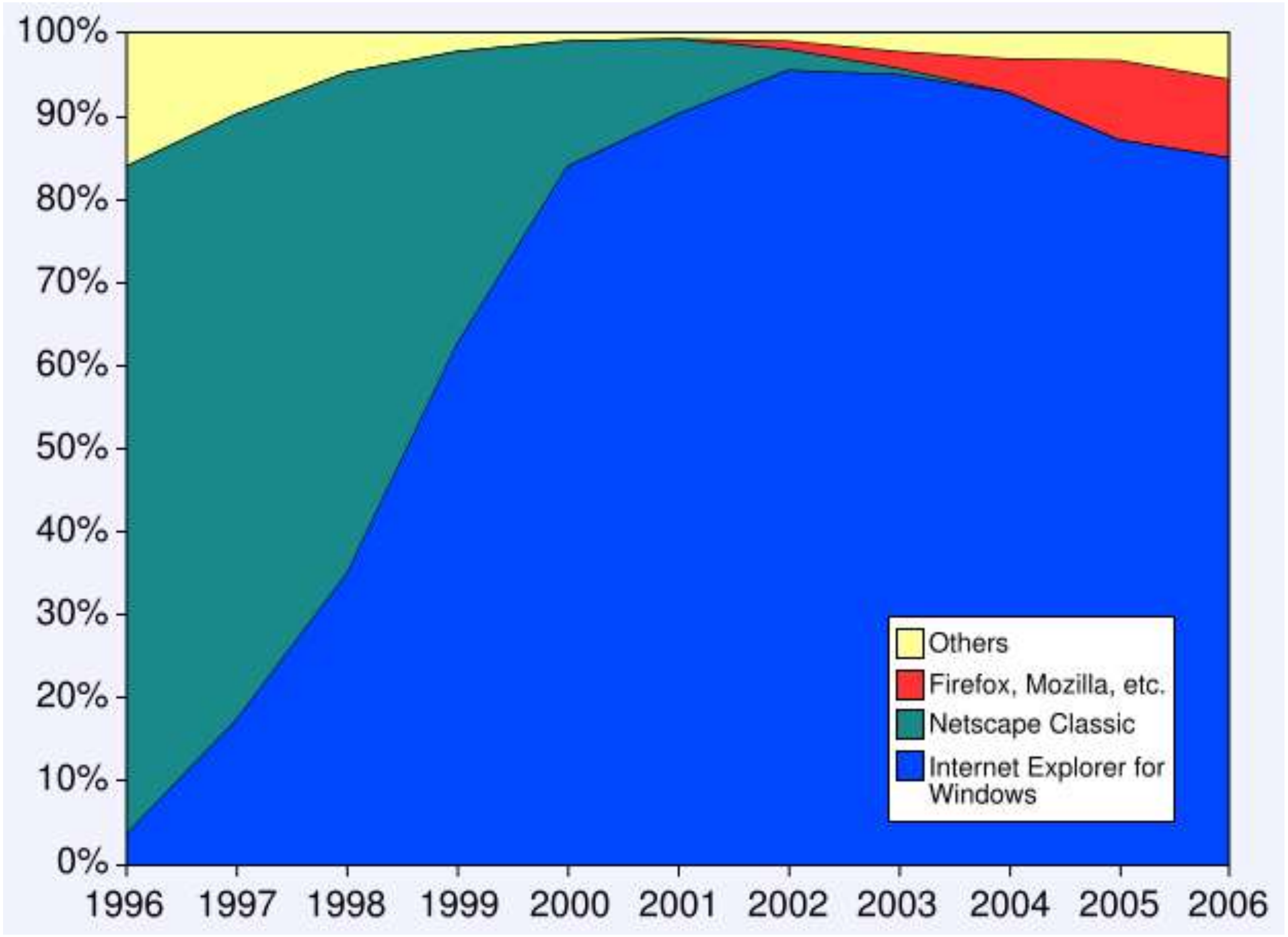
JavaScript

- Interpreted, untyped, object-oriented Programming Language (Scripting Language)
- First released in 1995
- Integrated with Browsers (the Browser is the Interpreter)
  - No additional software is required to run JavaScript
  - First shipped with Netscape Navigator 2.0
- Not related to the Java Programming Language!
- Standardised as ECMAScript (ECMA: European Computer Manufacturer's Association)
  - On principle, JavaScript code should behave similar on different Browser types
- Script Code can be integrated with HTML code

JavaScript Applications

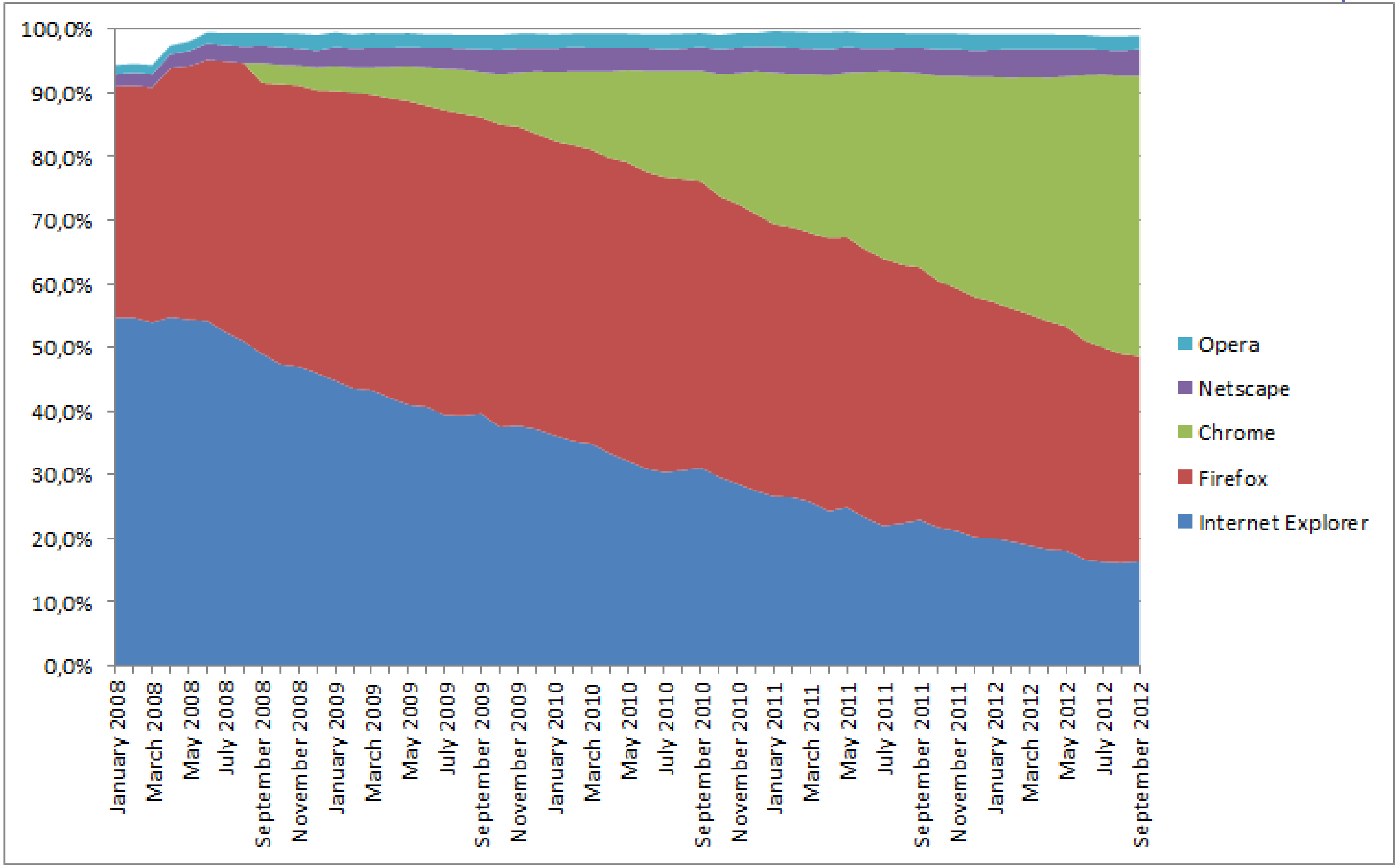
- Form Validation
- Dynamic Modification of HTML
  - Document Object Model (DOM) – will be covered later
- Dynamic Page Effects
  - Rollover Images
  - Menus
  - …
- Dynamic Page Behavior
  - Changing Hyperlinks
  - Changing Menus
- AJAX

Browser Wars



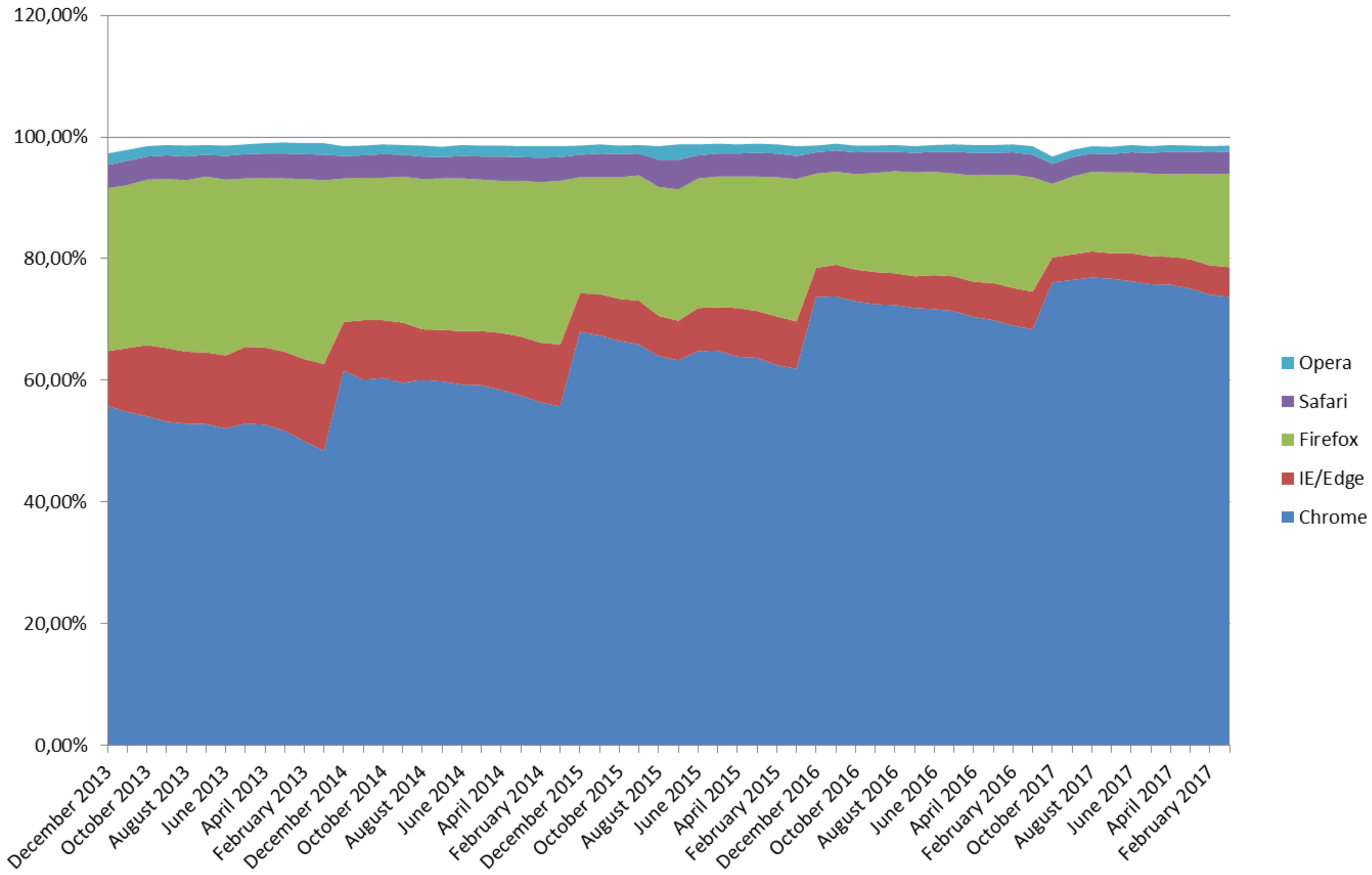
Source: de.wikipedia.org

Browser Wars continued



Data Source: www.w3schools.com/browsers/browsers_stats.asp

Browser Wars continued



Data Source: www.w3schools.com/browsers/browsers_stats.asp

JavaScript Development

- Just text editor (notepad, vi, scite, textpad, notepad++) required
  - Recommendation: use text editor with syntax coloring
- Development Steps
  1. Create text file with extension html or htm
  2. Write some JavaScript code
  3. Open html file in browser
- Different browsers show different behavior with JavaScript code
  - Code has to be adapted for Firefox, Internet Explorer, Opera, Safari, …
  - Code has to be tested with different browsers

Syntax

deprecated

```
<html>
<body>
<script language="JavaScript">
statement 1;
statement 2;
......
statement n;
</script>
</body>
</html>
```

optional in HTML5

```
<html>
<body>
<script type="text/javascript">
statement 1;
statement 2;
......
statement n;
</script>
</body>
</html>
```

Simple JavaScript Example

```
<script>


document.write ("Hello World!");
/* multi line comment: Here the methode
  write() of the object document is called.
  */
//single line comment
</script>
```

JavaScript And HTML

```
<html>
   <head>
   <title>My First Java Script</title>
   </head>
   <body>
   <p>
   <b>This is a line of code before the script</b><br />
   <script type="text/javascript"">
   var rightNow = new Date();
   document.write("This text was written with JavaScript! ");
               document.write("It is now!" + rightNow);
               /* Here the methode write() of the
                      object document is called. */
   </script>
   <br /><b>This is a line of code after the script</b>
   </b>
   </body>
</html>
```

JavaScript And HTML

- JavaScript can be embedded into XHTML/HTML…
  - Can be placed in HEAD or BODY section
  - You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section
- … but can also be used
  - from an external file or
  - within HTML links or events

Embedding of JavaScript into XHTML

DIRECT

```
<script>
document.write("Hello World");
</script>
```

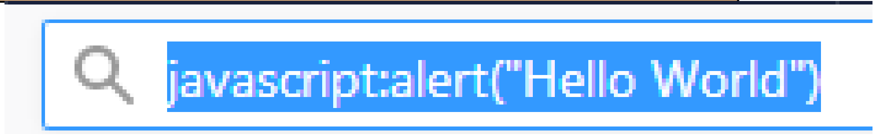
EXTERNAL FILE

```
<html>
<head>
<title>JavaScript</title>
</head>
<body>
<script type="text/javascript" src="hello.js">
</script>
</body>
</html>
```

Embedding of JavaScript into XHTML

LINK

```
<a href="javascript:window.alert(
'Hello World!');">
```



javascript:alert("Hello World")

EVENT HANDLER

```
<a href="#" onclick="alert('Hello World!');">
Hello 1</a>

<!-- OR -->

<a href="page2.html" onclick="alert(
'Hello World2!');">Hello World2</a>
```

Variables

- Storing of Data
- Variables have a name
  - Characters, Numbers, and Underline "_"
  - First Character has to be a letter
- Variables have a value
  - Is set during Initialisation or by applying an Operation
- Differentiation between local and global Variables
  - Local Variables are, e.g., defined within a function

Example

```
var i = 1; //define and initialise i
i++;       //(add 1 to i)
var x = 2.71;
x = x-1;
var y = Math.sqrt(25);
```

Variable Types

- Numerical/Number Type
  - PI = 3.14156;
- String Type
  - sCourse = "Web Engineering I"; // or 'Web Engineering I'
  - Control characters: \r – carriage return, \n – new line, \t – tab, \b – backspace, \f – page forward
- Boolean Type
  - bIsNumeric = true;
- Object Type
  - myObject = new object();
  - … will be explained in detail later
- Null Type (The null type has exactly one value ‘null’)
- Undefined Type (Any variable that has not been assigned is by default "undefined").

Arrays

- Arrays are Objects, not language elements
- Expose dynamic behavior
- Can store any data type

ARRAYS

```
var arr = new Array();
arr[0] = "Element1";
arr[2] = "Element1";
//…
```

String Operators

- + – String concatenation
  - sComplete = "Hello" + "World" + "!";
- .length – returns the number of characters
  - sMyString = "123456"; sLen = sMyString.length; //=6
- .charAt(x) – returns the char at position x
  - sMyString = "654321"; sLen = sMyString.charAt("2"); //=4
- .substring(start, end) – returns a substring
  - sVar1 = "Hello World"; sVar2 = sVar1.substring(6,11);

Arithmetic Operations

- Addition +
- Subtraction -
- Multiplication *
- Division /
- Modulo %

Example

```
var i = i + 1;

i = 13 % 5; //result: 3
```

Assignment Operators

- Combination of arithmetic Operators and equality sign "="
  - += : i = i + 1; is equal to i += 1;
  - -= : i = i - 2; is equal to i -= 2;
  - *= : i = i * 3; is equal to i *= 3;
  - /= : i = i / 4; is equal to i /= 4;
  - % : i = i % 5; is equal to i %= 5;

Comparison Operators

- == : is equal
- === : equal value and equal type
- != : is not equal
- !==: not equal value or not equal type
- < : is smaller than
- > : is greater than
- <= : is smaller than or equal to
- >= : is greater than or equal to

Boolean Operators

- ! - NOT
- || - OR
- && - AND

Example

```
var t = true;
var f = false;
var bool1 = t && f;  //-> false
var bool2 = t && t;  //-> true
```

Control Structures

IF...ELSE

```
if (condition) {
     Statements;
} else {
     Statements;
}
```

SWITCH

```
switch (Variable) {
     case value1 :
          Statements;
          break;
     case value2 :
          Statements;
          break;
     default :
          Statements;
}
```

Important!

Examples: Control Structures

```
var n1 = 10;
if (n > 20) {
   n -= 10;
} else {
   n +=5;
}
```

```
switch (menu_coice) {
   case 1: var choice = "Menu 1
   selected"; break;
   case 2: var choice = "Menu 2
   selected"; break;
   case 3: var choice = "Menu 3
   selected"; break;
   default: var m = "Please choose
   number between 1 und 3";
}
document.write(choice);
```

Works in JS!

Exception Handling

TRY CATCH

```
try {
   // Statements 1
} catch (ex) {
   // Statements 2
}
```

- Code in "Statements 1" is executed until error occurs ("exception")
- Code does not top on error, but continues with "Statements 2"
- "Throwing" of own exceptions: `throw("My Exception");`
- Important Concepts here:
  - "Design by Contract"
  - "Graceful Degradation"

Exercises 4.3 - 4.6