

Web Technologies

Agenda

Web core technologies

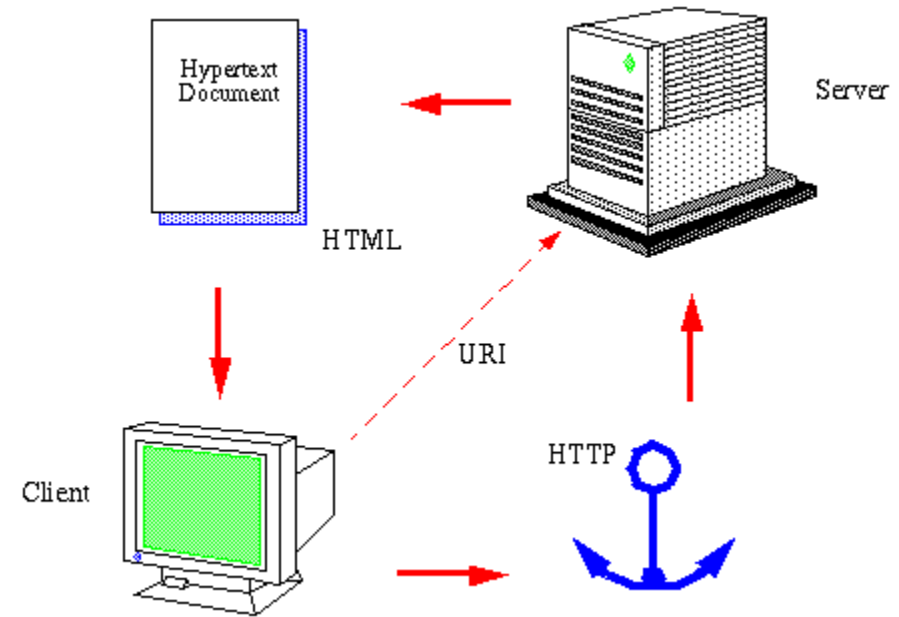
Web page key technologies

JavaScript language

Object Oriented Programming paradigm

World Wide Web

Information system where documents and other web resources are identified by Uniform Resource Locators, which may be interlinked by hypertext, and are accessible over the Internet. The resources of the WWW may be accessed by users by a software application called a web browser.



<https://www.youtube.com/watch?v=J8hzJxb0rpc>

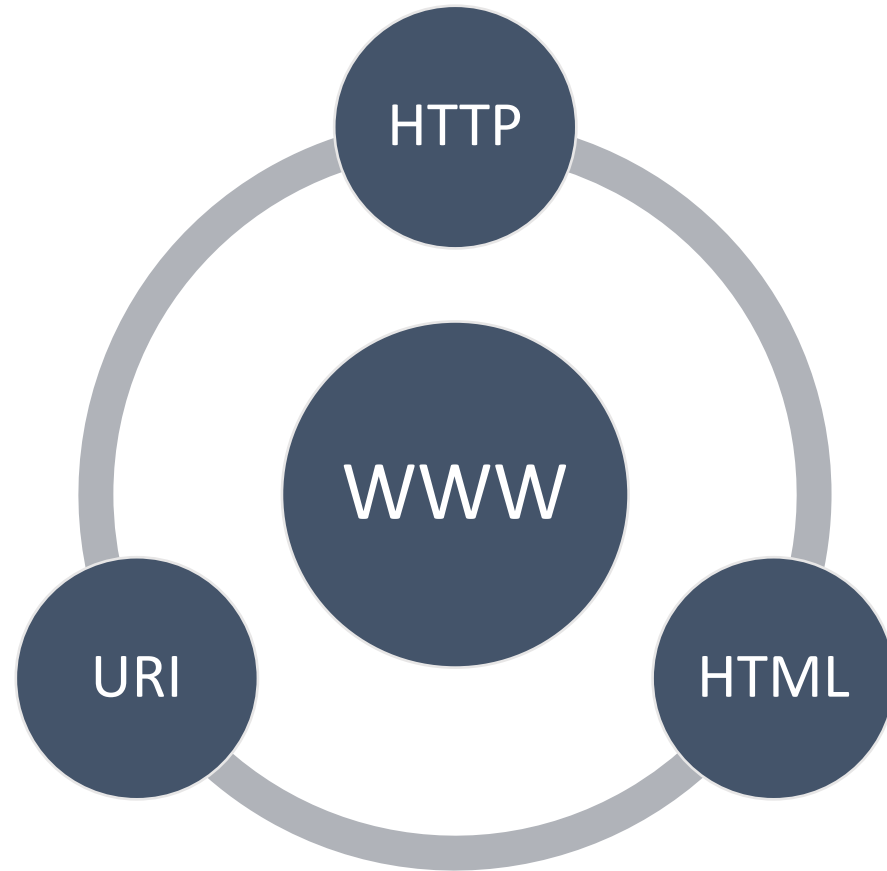
WWW Consortium

**the main international standards
organization for the World Wide
Web**



<https://www.w3.org/>

Web Core Technologies



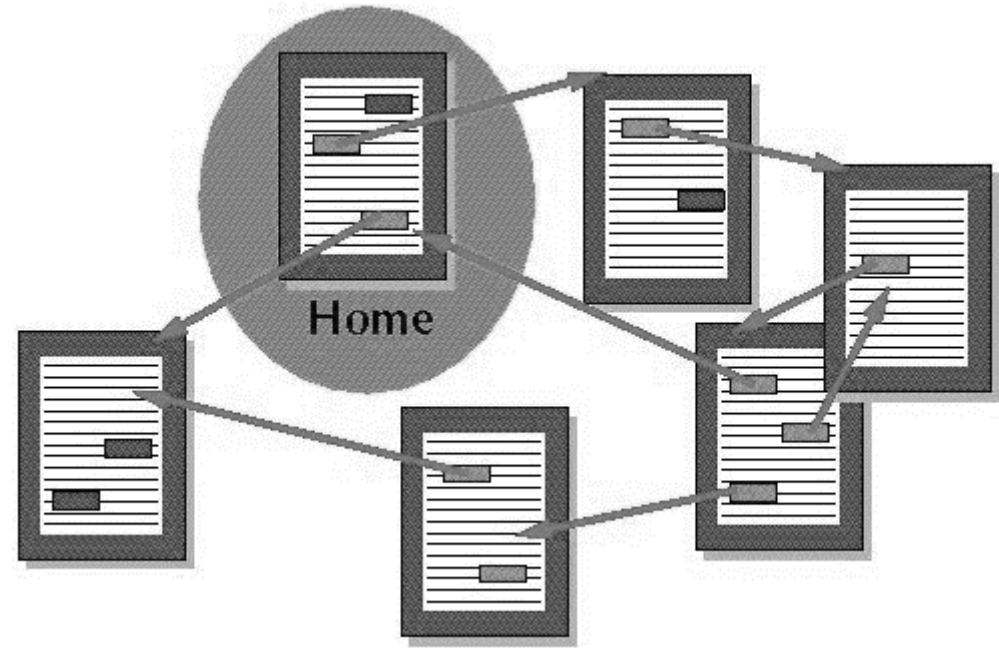
HTTP
Hypertext Transfer Protocol

HTML
Hypertext Markup Language

URI
Uniform Resource Identifier

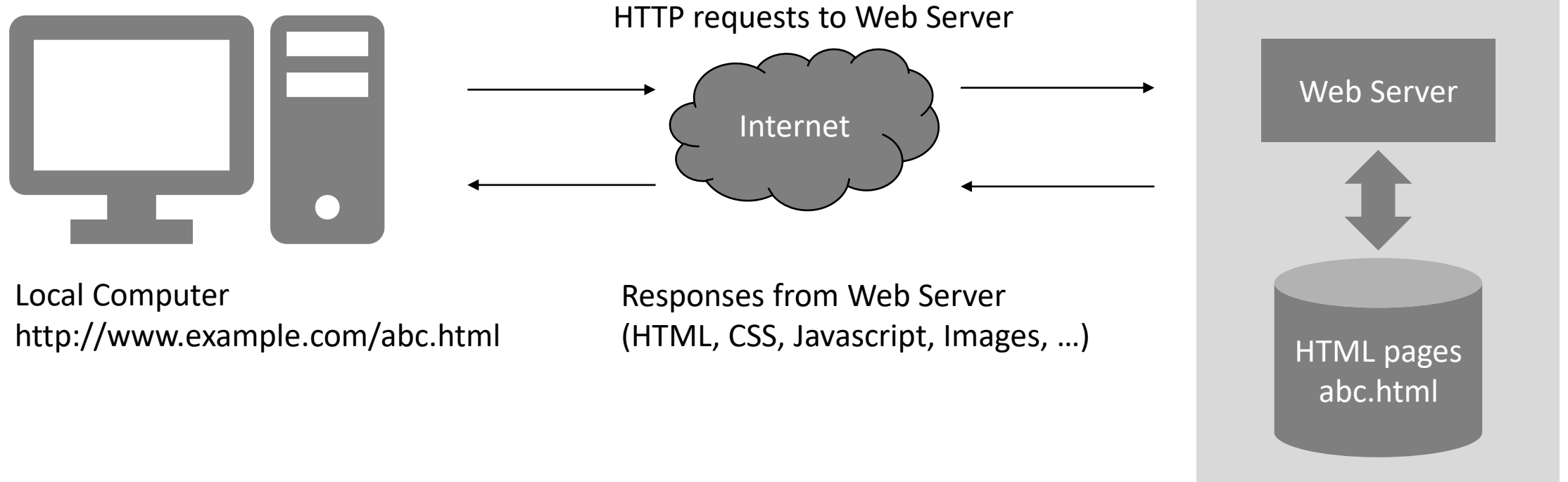
Hypertext

Hypertext is text displayed on a computer display or other electronic devices with references (hyperlinks) to other text that the reader can immediately access

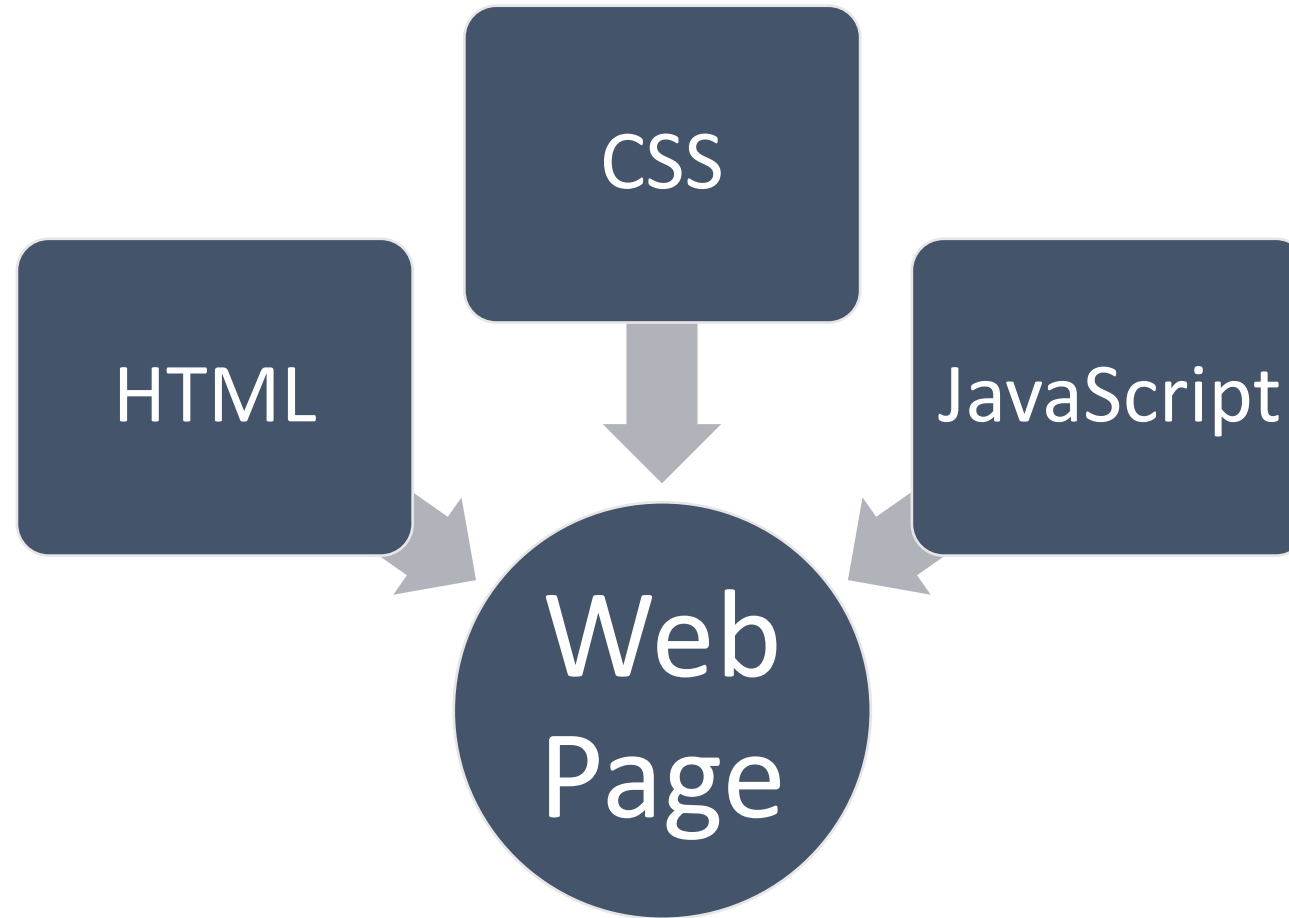


https://www.youtube.com/watch?v=1TVW9wE_jdw

Web Page delivering



Web Page Key Technologies



Web Technologies Basic Resources



**Educational website for learning
web technologies online**

<https://www.w3schools.com/>



**Resources for developers, by
developers**

<https://developer.mozilla.org/>

HTML document structure

```
<!DOCTYPE html>
<html>

  <head>
  ...
  </head>

  <body>
  ...
  </body>

</html>
```



HTML Elements and attributes

```
<tagname> ... </tagname>
```

opening and closing tag

```
<tagname attrname="value" attrname="value"> ... </tagname>
```

name/value pair
additional info about element
in opening tag

CSS Content styling

Inline

```
<h1 style="color:blue;">text</h1>
```

Internal

```
<style> ... </style>
```

External (CSS document)

```
<link rel="stylesheet" type="text/css"  
      href="mystyle.css">
```



JavaScript

Programming language

Object-oriented, imperative,
functional programming

Prototype-based ([class-based vs. prototype-based](#))

Interpreted (just-in-time
compilation)

WWW core technology

Tutorials: [MDN](#) [w3schools](#)



How Javascript works

<https://www.youtube.com/watch?v=b1ieJtlx1NY>

JS Location

Internal

HTML files (head / body)

```
<script>  
  ... JavaScript code ...  
</script>
```

External

JavaScript files

```
<script  
  src="/js/myScript.js">  
</script>
```

Input / Output

Dialog / popup windows: [alert, confirm, prompt](#)

[Console](#): Chrome DevTools

HTML DOM

```
var x = document.getElementById("elementID").textContent;
```

```
document.getElementById("elementID").innerHTML = "text";
```

Data types and variables

Boolean

Null

Undefined

Number

String

Object

Dynamic types

Variable declaration

```
var abc = value;
```

Variable scope

global, local

[Grammar and types](#)

Strings and numbers

Strings

```
var name = "John May";
```

Numbers

```
var price = 120.5;
```

String functions

[JavaScript String Reference](#)

Number functions

[JavaScript Number Reference](#)

[Numbers and dates](#)

Arrays

Store multiple values in a single variable

Array declaration

```
var array-name = [item1, item2, ...];
```

Array methods to operate on array

Flow control

Block statement

```
{ statements; }
```

Conditional statements

```
if...else, switch
```

Loops and iteration

```
for, do...while, while
```

Functions

```
function name(parameter1, parameter2, ...) {  
    code to be executed  
    return value; // optional  
}
```

Defining and calling functions

JavaScript Functions

Events

HTML code

```
<HTML-element some-event="some JavaScript">
```

JS code

```
element.addEventListener(event, function);
```

```
eg. var el = document.getElementById("h1");
```

```
    el.addEventListener("click", myFunction);
```

[Event reference](#)

Software development tools

Static code analysis:

JSLint (lint, linter)

Text editor:

Microsoft Visual Studio Code

Authoring and debugging:

Chrome DevTools

<https://developers.google.com/web/tools/chrome-devtools/>

Object Oriented Programming

JSON

<https://www.youtube.com/watch?v=GpOO5iKzOmY>

OOP

<https://www.youtube.com/watch?v=pTB0EiLXUC8>

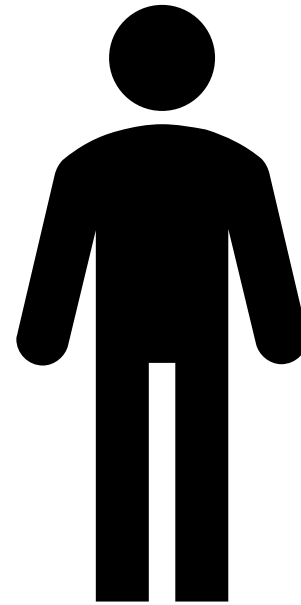
Objects

Properties

name
age
student
interests
address

Methods

greeting()



Object definition

```
var objectName = {  
    name1: value1,  
    name2: value2,  
    name3: value3,  
    ...  
}
```

value – number, string, array, object, function, ...

[Introducing JavaScript objects](#)

[JavaScript Objects](#)

Object definition

```
var person = {  
    name: "Adam",  
    age: 23,  
    student: true,  
    interests: ["swimming", "reading"],  
    address: {city: "Krakow", street: "Rakowicka"},  
  
    greeting: function() {  
        alert("Hi, I'm " + this.name);  
    }  
}
```

Object manipulation

```
alert (person.name) ;  
  
person.student = false;  
  
alert (person.fullname ()) ;  
  
person.age = 24;  
  
person.address.city = "Zakopane";  
  
person.interests[0] = "shopping";  
  
person.phone = "566 534 039";
```

Constructor functions

```
function Person(name, age, student) {  
    this.name = name;  
    this.age = age;  
    this.student = student;  
    this.greeting = function() {  
        alert("Hi, I'm " + this.name);  
    }  
}  
  
var person1 = new Person("Adam", 23, true);
```

Object prototypes

JavaScript – prototype-based language

Prototype-based ([class-based vs. prototype-based](#))

[Introduction to object prototypes](#)