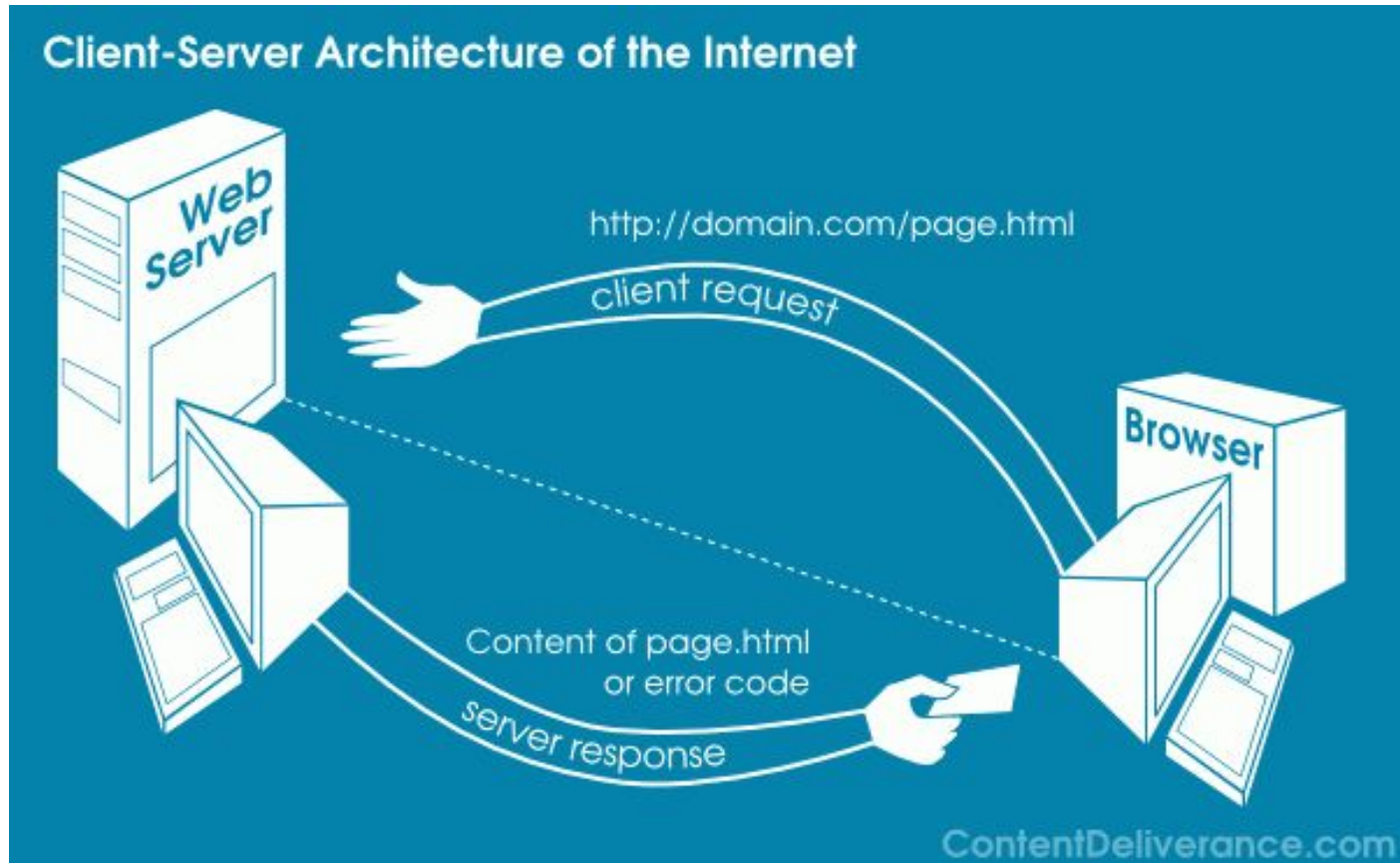


Web Technologies

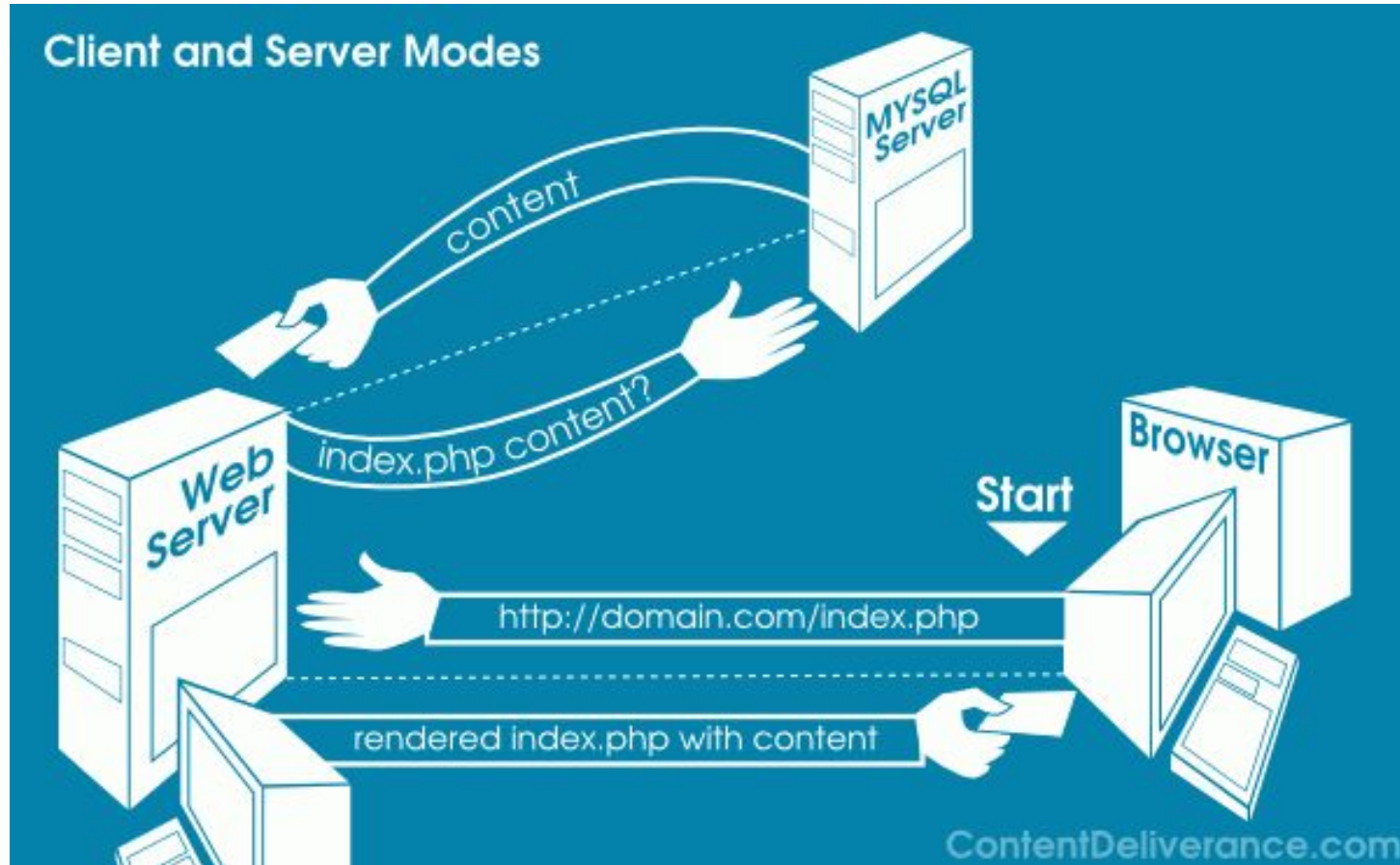
Lecture 5

Client side programming

Client-server architecture



Client-server architecture



Client-side programming

- Code runs in browser after the page is sent back from the server
 - HTML + CSS + Javascript
- Code manipulates the page or responds to user behavior
 - Form validation
 - Autocomplete
 - HTML and CSS modifications (dynamic HTML)
 - Cookies (state preservation)

Advantages

- **Usability**
 - Modify page without communicating with server
- **Efficiency**
 - Enact small and quick changes to page without waiting for server to respond
- **Event driven**
 - React to user interaction
 - Clicks, key presses

Disadvantages

- **Security**
 - Everything is stored on client computer
- **Compatibility**
 - Different browsers behave differently
- **Accessibility**
 - Cannot write local files, open server connections, connect to databases, etc.

Javascript

- A **lightweight** programming language
 - Scripting language
- Enables **interactive** web pages
 - Autocomplete
 - Event handling
 - Page loads, page clicks, key presses, mouse moves, etc.
 - Form validation
 - Adapt page based on browser capabilities
- Web **standard**
 - Browsers may handle javascript differently
- **Not** related to Java

Javascript

- **Interpreted** not compiled
- **Relaxed** syntax and rules
 - Fewer and looser data types
 - No need to declare variables
 - Silent errors (few exceptions)
- Focused on **functions** not classes
- **Embedded** within a web page

Placing Javascript code

<head>

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

</head>

- Javascript code can be also placed inside HTML **but**
 - It is not recommended
 - Content (HTML), presentation (CSS), and behavior (Javascript) should be separated
- Javascript is **Case sensitive**

Javascript popups

- `alert("Welcome to my website");`
 - Displays an alert dialog with a message for the user
- `answer = confirm("Would you like to continue?");`
 - Returns a boolean value depending on whether the user clicked "Yes" or "No"
- `value = prompt("Please enter your email address", "Your name here");`
 - Returns the value entered in the prompt dialog
 - A default value can be specified as 2nd argument

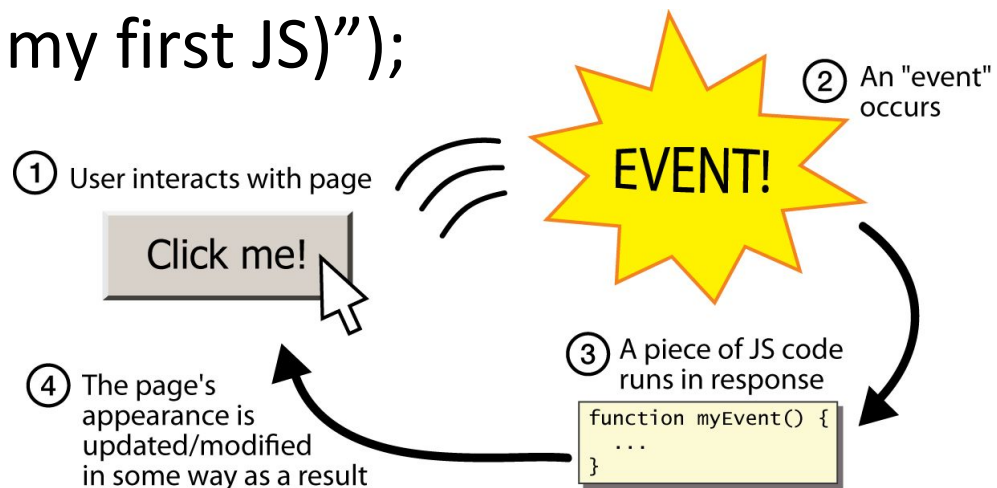
Javascript events

- **HTML code**

```
<input type="button" value="Click me!" onclick=  
  "myEvent();">
```

- **Javascript code**

```
function myEvent() {  
    alert("Hello! This is my first JS");  
}
```



Variables

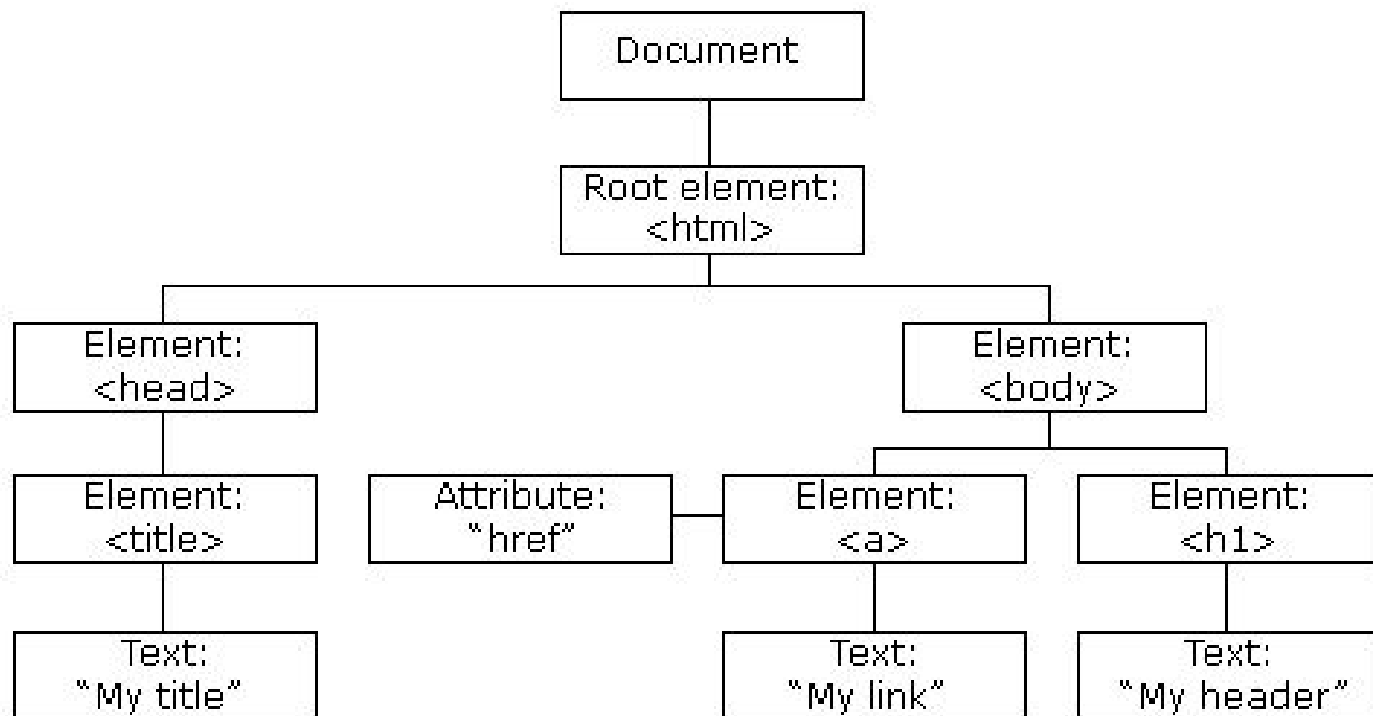
- `var name = expression;`
 - `var age = 32;`
 - `var name = "John";`
 - `var weight = "85.5";`
- Variables are declared using the *var* keyword
- Data types not specified
- Javascript is loosely typed
 - Can find the type by calling *typeof*
 - Number, Boolean, String, Array, Object, Function, Null, Undefined

Document Object Model

- DOM
 - A set of Javascript objects that represent each element on a page
 - Can be used to manipulate HTML
 - Dynamic HTML
 - Examine an element's state
 - Checked vs. unchecked
 - Change state
 - Insert text in a div
 - Change style
 - Make a paragraph red

DOM hierarchy

- When a page is loaded the browser creates a tree of objects (the DOM model)



Manipulating HTML

- HTML code

```
<input type="button" value="Change paragraph text"
onclick="changeValue();" >
<p id="text">Click button to change me!</p>
```

- Javascript code

```
function changeValue() {
    var element = document.getElementById("text");
    element.innerHTML = "You just changed me!";
    element.style.color="blue";
}
```

Manipulating HTML

- HTML content
 - element.**innerHTML**
 - element.**value**
 - element.*attribute*
 - Where *attribute* is the attribute name of the element
 - Example: element.**src**
- CSS
 - element.**style.property**
 - *property* is the CSS property to be changed
 - Example: element.style.**color**

Event listeners

addEventListener() method

document.getElementById("myButton").addEventListener("click", displayDate);

- Attach event listeners to the specified element
- Does not override existing event handlers
- Many handlers can be added to the same element
- Handlers can be added to any DOM elements
 - Example: the window object
- Handlers can be removed by using the **removeEventListener()**

What's next?

- State vs. stateless
 - Server vs. client side
 - Cookies and Web storage
- AJAX
 - Synchronous vs. asynchronous
- JQUERY
- Server side programming
- Web services