

Università Degli Studi di Ferrara

Corso di Laurea in Informatica - A.A. 2023 - 2024

Tecnologie Web

Lez. 09 - JavaScript

Cap. 4

Tecnologie Client Side

In questa lezione...

- JavaScript
- DOM
- Framework JavaScript

Tecnologie Client Side

Si definisce tecnologia Client Side una tecnologia che viene eseguita sul lato Client di un sistema Client-Server dopo il trasferimento dei dati.

Tecnologie Client Side

Lo scopo delle Tecnologie Client Side nello sviluppo di applicazioni web è quello di arricchire le funzionalità di HTML

Migliorare gli aspetti multimediali

- Riproduzione di audio e video

Aggiungere funzionalità ai moduli online

- Ampliare la varietà campi di input (date)
- Verifica dati inseriti

Dare dinamicità delle pagine

- AJAX

Browser

Queste tecnologie vengono eseguite sul lato Client di un sistema Web-based, quindi vengono eseguite dal Browser.

- Il Browser ha tre compiti principali:
- Generare ed inviare le richieste
 - Ricevere e interpretare le risposte del server
 - Visualizzare i contenuti ricevuti

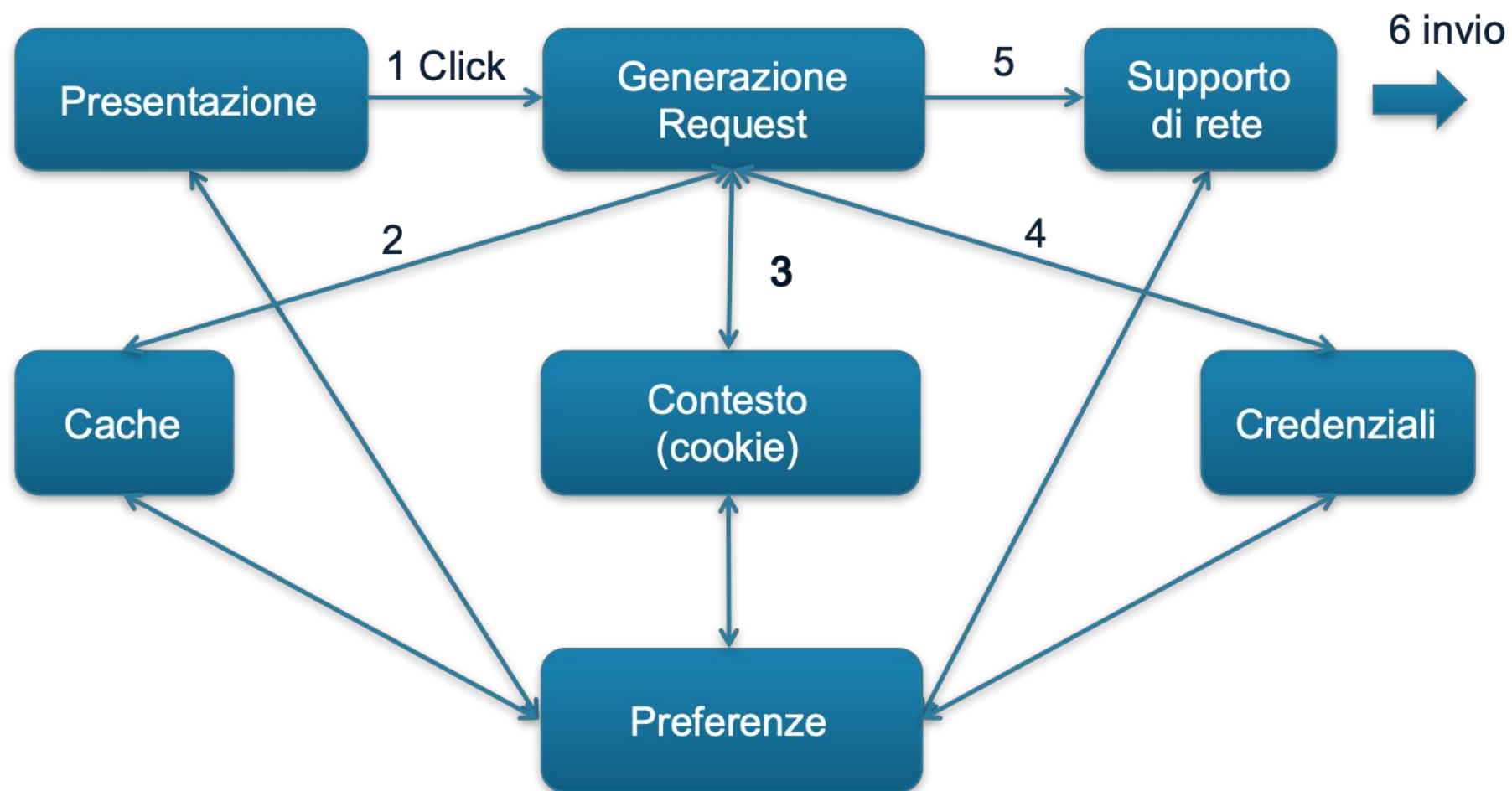
Browser

In realtà il browser si occupa anche di altri compiti, fra cui:

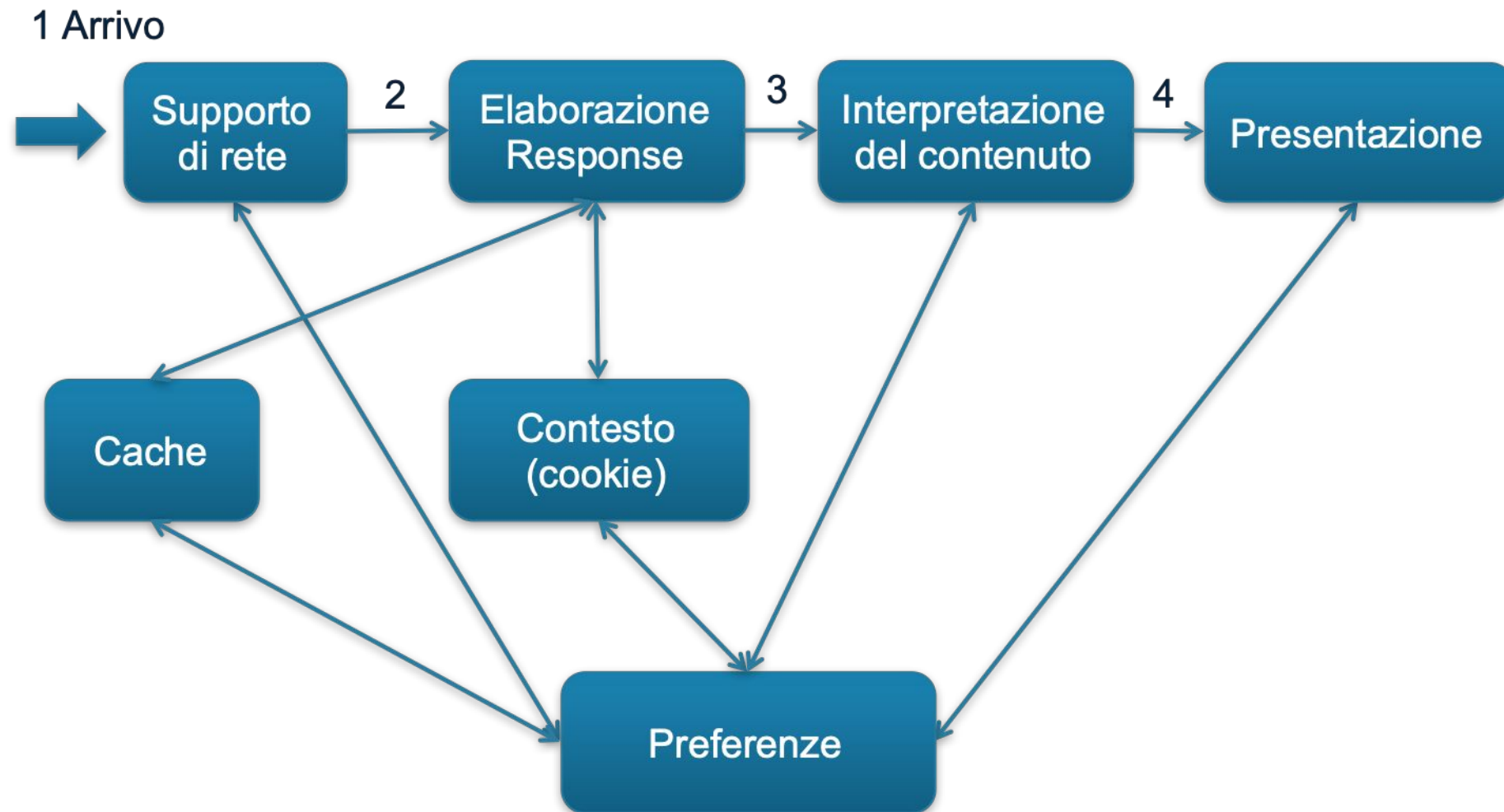
- Le richieste di autenticazione
- Il contesto (cookie)
- La cache
- I codici di risposta HTTP
- Gli errori
- La visualizzazione di contenuti complessi
- Gestire le password

... Rispettando le preferenze dell'utente

Browser: Gestione della richiesta



Browser: Gestione della risposta



JavaScript

JavaScript è un linguaggio di scripting orientato agli oggetti originariamente sviluppato da Netscape Communication.

- | | | | | |
|----|---|--|-------|------------|
| Ha | • | cambiato | più | nomi |
| | | Mocha | (nome | originale) |
| | | • | | LiveScript |
| | • | JavaScript (per la sintassi ispirata a Java) | | |
| | • | ECMAScript (fra il 1997 ed il 1999) | | |

JavaScript

- È un linguaggio di scripting
 - È un linguaggio interpretato
 - Non necessita di licenza
 - Rende le pagine HTML interattive
- È solitamente integrato nelle pagine HTML

JavaScript

Attraverso JavaScript possiamo

- Inserire dinamicamente porzioni di codice HTML nei documenti
 - Gestire gli eventi sul documento
- Leggere e modificare il contenuto degli elementi
 - Validare il contenuto di un form
- Creare e leggere cookies

JavaScript

La sintassi di JavaScript è una sintassi ispirata a Java ed è molto familiare a chi ha studiato C/C++ e Java.

In JavaScript esistono:

- Loop: for e while
- Costrutti di selezione: if, if/else e switch

Inoltre esistono anche:
Array, Stringhe, Numeri, Date e Oggetti

JavaScript

Noi non studieremo approfonditamente JavaScript, ne faremo un uso semplice, quasi intuitivo.

Lo useremo in abbinamento a librerie che ci permetteranno di manipolare il documento HTML.

Se volete approfondire potete consultare il sito di W3School: <http://www.w3schools.com/js/default.asp>

JavaScript

Esistono 2 modi per integrare codice JavaScript all'interno di una pagina HTML:

- Integrando il codice JavaScript nella pagina.
- Importando il codice JavaScript da un file esterno.

Per entrambi i metodi si usa il tag `<script>`

Esempio 1

JavaScript

Le basi del linguaggio

JavaScript è un linguaggio molto semplice, non tipizzato e con una sintassi molto simile a Java.

Variabili:

```
var x = 4;
```

Operatori:

```
var y = x + 4;  
var fullName = fname + " " + lname;
```

Le basi del linguaggio

Array:

```
var ages = Array(); // Array vuoto  
var ages = [16, 43, 11];
```

Loop For:

```
for (var i=0; i<ages.length; i++) { .. }
```

If statement:

```
if (x == y) { ... }
```

Le basi del linguaggio

Funzioni:

```
function something(x) {  
    // faccio qualcosa con x  
}
```

```
function sum(x, y) {  
    return x+y;  
}
```

Le basi del linguaggio

Oggetti:

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

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

Le basi del linguaggio

Accesso alle proprietà:

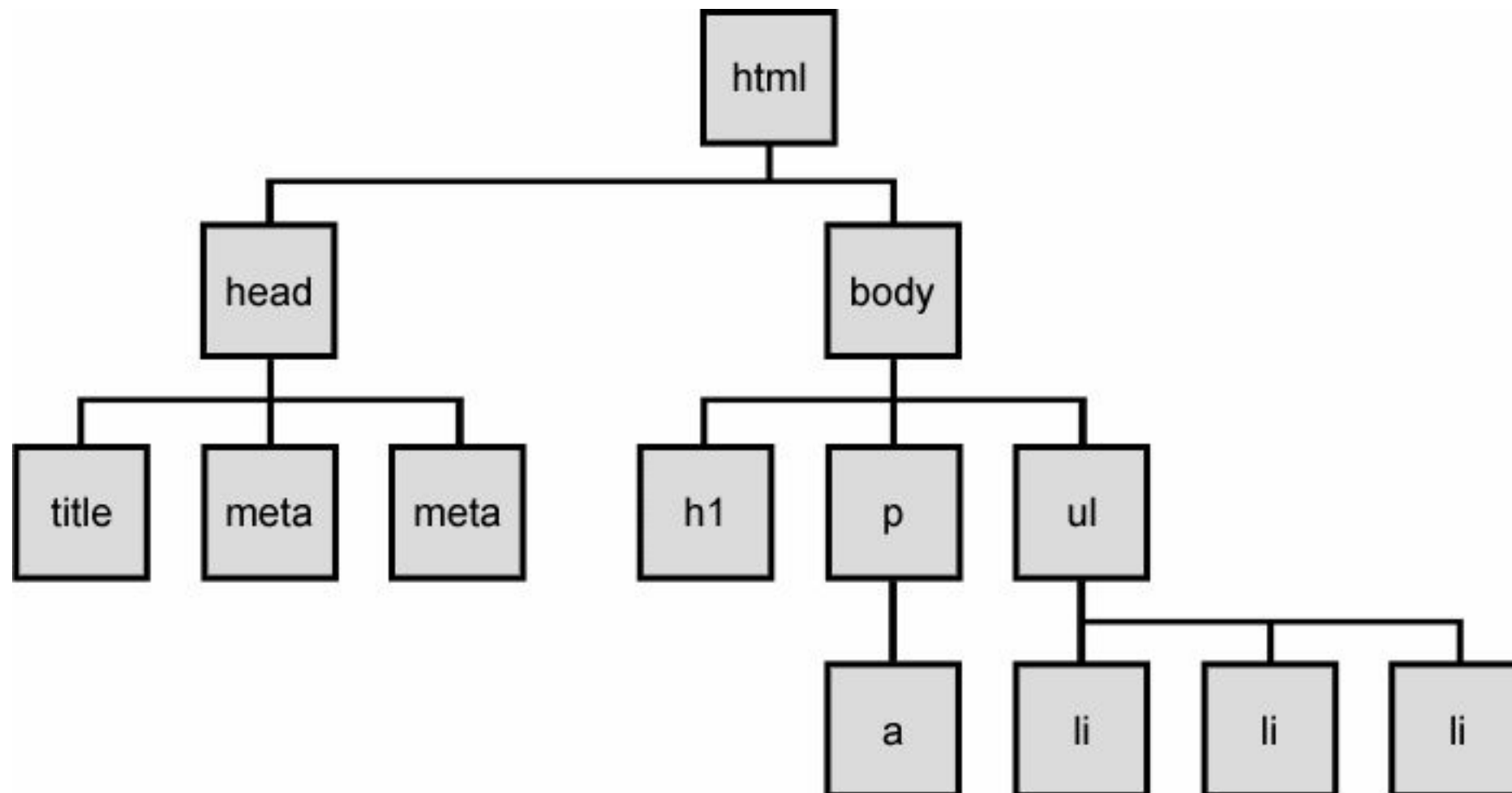
```
car.type;  
car['type'];
```

Metodi:

```
var car = {  
    type: function() { ... },  
};
```

DOM

La sigla DOM sta per Document Object Model, ovvero la rappresentazioni ad Oggetti del Modello di Documento.



DOM

È una rappresentazione ad albero del documento in cui i nodi possono avere tipi diversi:

Tipo	Descrizione	Esempio
Documento	Rappresenta la radice dell'albero. Esiste un solo elemento di questo tipo	<!DOCTYPE>
Elemento	Individua un tag HTML	<p> <div> <h1>
Attributo	Corrisponde ad un attributo di un Elemento	src nel tag
Testo	Corrisponde al contenuto testuale di un Elemento	<p>Testo</p>

JavaScript e DOM

JavaScript ci permette di manipolare il DOM per migliorare l'esperienza utente nella navigazione del sito Web oppure nell'utilizzo dell'applicazione Web.

Ottenere gli elementi

HTML:

```
<p id="paragrafo">Questo è un paragrafo</p>
```

JavaScript:

```
var par = document.getElementById("paragrafo");  
console.log(par);
```

Esempi 2 e 3

JavaScript

Eventi

È possibile invocare una funzione JavaScript allo scatenarsi di un evento nel documento HTML

Evento	Descrizione
onclick	Quando viene cliccato su un elemento
onchange	Quando cambia il valore di un elemento <select>
onmouseover	Quando il mouse “entra” sopra a un elemento
onmouseout	Quando il mouse “esce” da un elemento
onkeydown	Quando viene premuto il tasto della tastiera
onload	Quando viene caricato l’elemento

Eventi

```
<a href="..." onclick="check()" >...</a>
```

```
<script type="text/javascript">  
function check() {  
    ...  
}  
</script>
```

Esempio 4

Eventi

Eventi

Potete trovare ulteriori informazioni sugli eventi JavaScript su

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

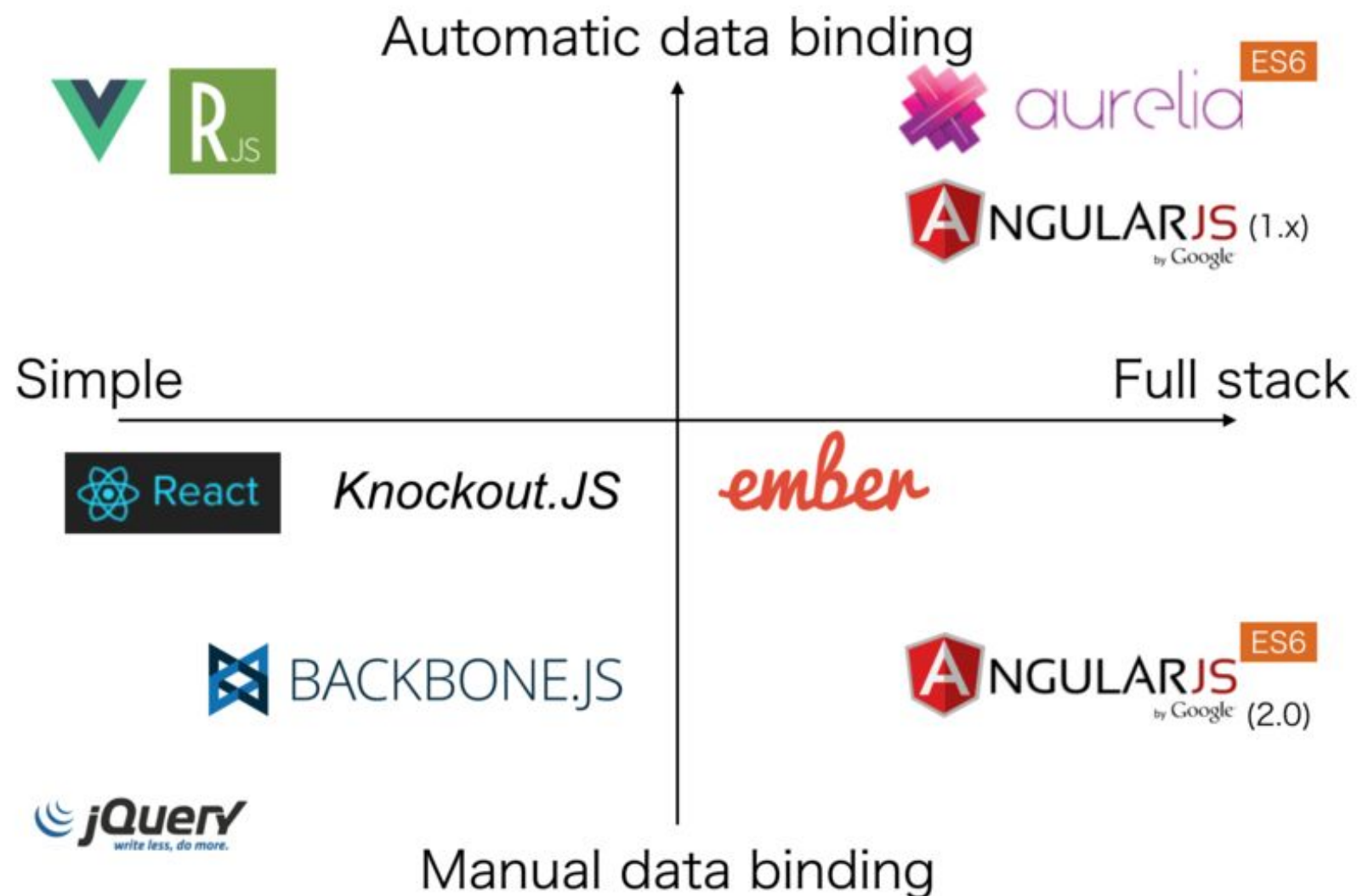
Framework JavaScript

Nonostante ECMA lavori costantemente a una standardizzazione, ogni browser ha una diversa implementazione del linguaggio JavaScript, per questo non possiamo essere sicuri che uno script abbia lo stesso comportamento su tutti i browser.

A questo problema pongono rimedio una serie di strumenti il cui scopo è proprio quello di occuparsi di queste differenze di implementazione e anche di rendere qualche funzionalità più facile da utilizzare

Framework JavaScript

Oggi esistono centinaia di framework (o librerie) JavaScript



Framework JavaScript

I commenti su
Hacker News
riguardo
all'imminente uscita
di Angular 4.0





Publicato per la prima volta nel 2005 da John Resig.

Semplice e largamente utilizzato, jQuery è una libreria di funzioni scritta in JavaScript che astrae ad un livello più alto la programmazione lato client.

Permette di manipolare le proprietà di un elemento (dimensioni, colore, posizione, ...) e anche di creare animazioni.

jQuery

La libreria è composta da 3 progetti separati

jQuery Core

- Accesso agli elementi;
- Funzioni primitive;
- AJAX.

jQuery UI

Estensione per interfacce utente

Plugins

Estensioni sviluppate dalla community

jQuery

L'installazione della libreria è banale:

Dal sito <https://jquery.com/> si scarica lo script e si include nelle pagine HTML o PHP come uno script javascript esterno.

Funzione \$

Permette di selezionare gli elementi del DOM:

- `$ ("#title")` : seleziona l'elemento con id "title";
- `$ (".top")` : seleziona gli elementi con classe "top";
- `$ ("p")` : seleziona tutti i paragrafi.

Restituisce una collezione di elementi individuati dal selettore

Funzione \$

```
$ ( 'document' ) .ready (function () {  
  
} ) ;
```

Ed ecco ritornare le funzioni anonime.

Il codice JavaScript viene passato come funzione anonima al metodo `ready()` che esegue la closure quando il documento (`document`) ha terminato il caricamento

Esempio 5

jQuery

jQuery Funzioni

Una volta ottenuti gli elementi con il selettore, jQuery ci consente di manipolare il DOM ottenendo valori ma anche cambiandoli

Ovviamente le funzioni di jQuery sono moltissime e non potremmo elencarle tutte.

A questo indirizzo: <http://api.jquery.com/> potete trovare l'elenco completo, noi ci limitiamo a vederne alcune fra le più comuni

jQuery Funzioni

```
$ ( 'div#container' ) .html ( ) ;
```

Ottiene TUTTO il codice HTML contenuto nel div che ha id="container"

```
$ ( 'div#container' ) .html ( "<h1>Titolo<h1>" ) ;
```

Sostituisce il codice HTML (in questo esempio un titolo) nel div che ha id="container"

Esempi 6 e 7

.html()

jQuery Funzioni

```
$ ( 'p#p1' ) .text ( ) ;
```

Ottiene il CONTENUTO del tag HTML selezionato

```
$ ( 'p#p1' ) .text ( "...") ;
```

Sostituisce il contenuto del tag HTML selezionato

Esempi 8 e 9

.text()

Altre Funzioni

`.attr() ;`

`.addClass() ;`

`.removeClass() ;`

`.css() ;`

`.hide() ;`

`.show() ;`

`.append() ;`

`.prepend() ;`

`.isEmpty() ;`

`.parent() ;`

`.children() ;`

`.val() ;`

Binding

È possibile intercettare gli eventi sui vari elementi usando il selettore e la funzione `bind`

```
$( '.link' ).bind( <evento>, function() {  
    ...  
} ) ;
```

Ad esempio:

```
$( '.link' ).bind( 'click', function() {  
    ...  
} ) ;
```

Binding

Gli eventi gestibili sono tutti quelli gestibili da JavaScript, come:

- `click`
- `mouseenter`
- `mouseleave`
- `submit`

Potete trovare l'elenco completo a questo link:

https://www.w3schools.com/jsref/dom_obj_event.asp

.each()

Quando un selettore ritorna più di un elemento, ci viene restituito un array, su cui possiamo usare il metodo `each()` per fare un loop su tutti gli elementi

```
$( 'ul#list li' ).each( function() { ... } );
```

Ad ogni iterazione del ciclo possiamo riferirci all'elemento corretto con `$(this)` e utilizzarlo come selettore.

Esempio 10

`.each()`

Semplici animazioni

Uno degli scopi più comuni nell'utilizzo del framework JavaScript è il migliorare la user experience durante la navigazione di un sito Web o della applicazione.

Migliore user experience significa anche animazioni:

```
.fadeIn ( "slow" ) ;
```

```
.fadeOut ( "fast" ) ;
```

```
.fadeToggle ( "fast" ) ;
```

Esempio 11

Animazioni

Animazioni più complesse

È possibile combinare più cambiamenti ed eseguire un'animazione.

È necessario costruire un oggetto JavaScript in cui per ogni proprietà CSS che si vuole cambiare deve essere presente una proprietà:

```
var props = {  
    left: '300px',  
    opacity: '0.5'  
}
```

Esempio 12

Animazioni più complesse

Esercizio 1

Realizzate un modulo di inserimento (form) in cui vengono richiesti nome, cognome e data di nascita (formata da giorno/mese/anno inseribili con 3 select).

Attraverso una funzione JavaScript, intercettate l'evento `click` sul `button` e verificate che la data di nascita appartenga a una persona maggiorenne prima dell'invio del modulo visualizzando un `alert()` in caso di fallimento del check.

Usate la funzione `bind()` sull'evento `click` del bottone. Usate `e.preventDefault()` per impedire l'invio del form. Lanciate l'evento `submit` sull'elemento `<form>` per continuare l'invio.

Esercizio 2

Usando l'esercizio 1 della lezione 04, disponibile sul sito del corso, aggiungete la possibilità di inserire nuove righe nella tabella attraverso una sezione di inserimento (con input e button o link) attraverso jQuery implementando la funzione `addRow()`.

Esercizio 3

Aggiungete un po' di stile alla pagina: Usate le classi di Bootstrap per migliorare l'aspetto del Form

Esercizio 4

Implementate la funzione `delRow()` che esegue l'eliminazione di una riga.

In ogni riga deve essere presente il link 'Cancella' che esegue l'eliminazione dal DOM della riga corrispondente.

Domande?