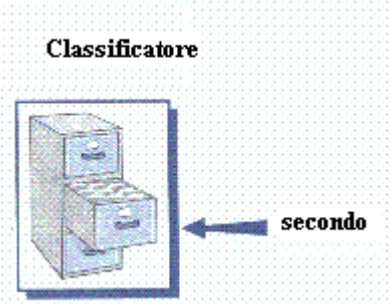
Array



Una matrice (**array**) è un *insieme ordinato* di dati identificato con unico *nome*: è una variabile che contiene uno o più valori in sequenza ordinata. Questi valori si chiamano "elementi". I singoli elementi sono individuati tramite indici.

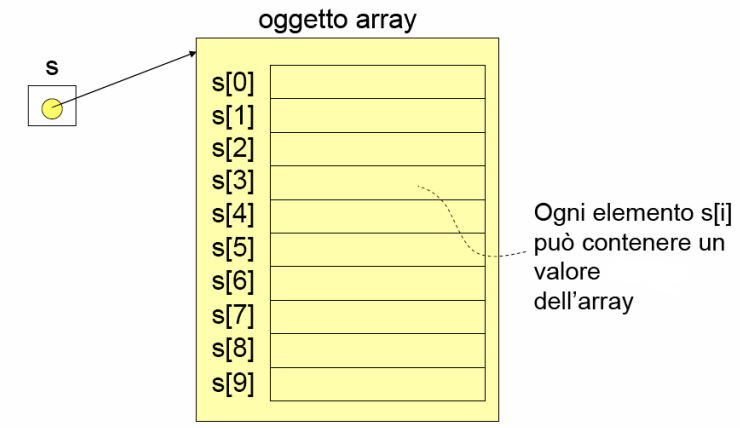
Variabile di tipo strutturato, dunque, in quanto l'indice introduce delle relazioni all'interno dell'insieme, stabilendo un nuovo ordine che non è necessariamente quello dei valori degli elementi componenti.

Il tipo dell'indice deve essere scalare, non reale perché, per ogni elemento, deve essere possibile definire "il precedente" ed "il seguente".

Il *nome* dell'insieme è l'*indirizzo* della locazione di memoria dove è memorizzato il valore del primo elemento; gli altri elementi della lista occuperanno locazioni di memoria consecutive.

In JavaScript un **array** è un oggetto di **tipo** **Array** con attributi (tra cui **length** che memorizza la dimensione dinamica dell'array) e metodi che ne permettono manipolazioni. Un **oggetto** in JavaScript è una collezione di valori (*nomi* di attributi e metodi), un'entità a sé stante contenente dei dati (***attributi***) ed un insieme di procedure per manipolare questi ultimi (***metodi***). Per far riferimento ad una *proprietà* di un oggetto si fa riferimento al nome dell'oggetto seguito dal punto (.) e dal nome della proprietà. Esempi di tale *dot-notation* sono, per riferirsi agli attributi di un oggetto immagine, immagine.altezza oppure immagine.larghezza o ancora, se l'oggetto rende disponibile una funzione detta **metodo** dell'oggetto, *nomeOggetto.metodo*().

Schema generico di un array



Implementazione di un **array** (matrice) **monodimensionale**

La *sintassi* per creare una **matrice monodimensionale** è la seguente:

nome = new Array() ; // crea array vuoto

Ad esempio, per creare un **array** di nome **s** si usa la seguente sintassi: var **s = new Array()** ;
che raccoglie sotto lo stesso nome un insieme di elementi ***anche eterogenei***, in numero ***anche variabile***

Per accedere ad un elemento dell'array, per leggerlo o modificarlo, usiamo la notazione *nome* [i]
dove "i" è l'indice che indica la posizione all'interno dell'array. Nel contare gli elementi di un array si comincia sempre dallo 0.

Ad esempio, per assegnare un valore ad un elemento si usa la seguente sintassi : *nome* [i] = *valore*

Per creare l'**array** e contemporaneamente inizializzare i valori degli elementi si usa la sintassi:

nome = new Array (*valore1, valore2, ...*) ;

Il metodo costruttore può essere usato in maniere differenti:

```
var s = new Array (10);   // crea un array contenente 10 elementi
var vocali = new Array ("A", "E", "I", "O", "U");  // crea un array contente le vocali
var lettere_straniere = ["J", "K", "W", "X", "Y"];   // modo breve per inizializzare
```

In JavaScript ci sono diversi metodi che aiutano a manipolare gli array. Fra tutti, alcuni particolarmente utili:

Proprietà **attributo o metodo**	**Descrizione**	**Esempio**
length	Conoscere la lunghezza di un array	var alunni=new Array("Mario", Gianni","Monica"); alert (alunni.**length**);

push(elemento)	Aggiungere un elemento in coda all'array e restituire la nuova lunghezza	alunni=new Array("Mario", "Gianni","Monica");
		aggiungi=alunni.**push**("Davide");
		alert (aggiungi);

concat (elementi da aggiungere)	Aggiungere elementi ad un array e restituire la nuova lunghezza. Restituisce un nuovo array formato dalla somma degli elementi	var alunni=new Array("Mario", "Gianni","Monica");
		aggiungi=alunni.**concat**("Davide","Giovanni");
		alert (aggiungi);

pop()	Eliminare un elemento dalla fine dell'array e restituisre il nome dell'elemento eliminato.	var alunni=new Array("Mario", "Gianni","Monica");
		togli=alunni.**pop**();
		alert(togli);
		alert(alunni.length);

shift()	Eliminare un elemento dall'inizio dell'array e restituire il nome dell'elemento eliminato	var alunni=new Array("Mario", "Gianni","Monica");
		togli=alunni.**shift**();
		alert(togli);
		alert(alunni.length);

reverse()	Invertire l'ordine degli elementi di un array	var alunni=new Array("Mario", "Gianni","Monica");
		alunni.**reverse**();
		alert(alunni[0]);

slice(inizio,fine)	Dividere l'array in un array più piccolo e restituire il nuovo array	var alunni=new Array("Mario", "Gianni","Monica","Davide");
		alert(alunni.length);
		alunni2=alunni.**slice**(0,2); // per copiare porzione di array
		// tra 0 incluso e 2 escluso
		// o fino alla fine dell'array se non indicato 2
		alert(alunni2.length);

sort()	Ordina l'array secondo l'ordine alfabetico	var vocali = new Array ("e", "a", "u", "i", "o"); vocali.**sort**() *// ora le vocali sono ordinate nell'array vocali* */ attenzione: lavora direttamente sull'array*

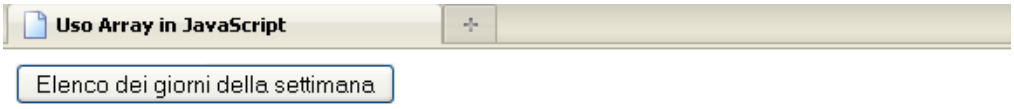
Attenzione: nel codice Unicode (già in ASCII) le minuscole seguono alle maiuscole (ad esempio sono ordinate *E U a i o*); per evitare inconvenienti nell'uso del metodo sort() in JavaScript, si può lavorare su stringhe solo minuscole, usando il metodo **toLowerCase** dell'oggetto **String**

Codice esempio

```
<html>
<head>
	<title>Uso Array in JavaScript</title>
	<meta name="Generator" content="Alleycode HTML Editor">
	<script>
	<!--
	 var settimana = new Array ("Lunedì", "Martedì", "Mercoledì", "Giovedì",
	                            "Venerdì", "Sabato", "Domenica");

	  function scrivi() {
	                    msg="";
	                    for (i=0; i<settimana.length; i++) {
	                            msg = msg + settimana[i] + "\n";  // accodamento di stringhe
	                    }
	                   window.alert(msg);
	    }
	 //-->
	</script>
</head>

<body>
	<form>
		<input type="button"
		    value="Elenco dei giorni della settimana" onClick="javascript: scrivi();">
	</form>
</body>
</html>
```



Nb: JavaScript **non supporta le classi** e, a differenza di linguaggi OO, usa oggetti di cui può modificare *proprietà*. Ad esempio una pagina web viene scomposta da JavaScript in un modello ad oggetti (ognuno con le sue proprietà) in relazione reciproca.

Per creare un nuovo oggetto è necessario partire da un modello (in JavaScript un oggetto chiamato **prototipo** definito con un **costrutttore**) che indichi come creare altri oggetti dello **stesso tipo** (ogni oggetto è una **istanza** del prototipo).

Gli oggetti possono dunque possedere delle caratteristiche (**attributi**): nel caso ad esempio di macchina (per fare un paragone con la realtà), saranno la cilindrata, le dimensioni, il costo, ecc....

Ciascuna istanza espone inoltre la possibilità di effettuare delle operazioni (**metodi**) che si possono richiedere come *servizi*: per la nostra macchina, metterla in moto o guidare. Queste operazioni modificheranno delle caratteristiche come il livello del suo carburante o la velocità.