

# Tecnologie Web: Javascript

Prof. Raffaele Montella, PhD

[raffaele.montella@uniparthenope.it](mailto:raffaele.montella@uniparthenope.it)

# Sommario

- Oggetti
- Array
- Espressioni regolari
- Conclusioni

# Oggetti

- Un oggetto JavaScript è un costrutto che possiede proprietà (attributi) e metodi (funzioni)
- La sintassi per accedere alle proprietà di un oggetto è `oggetto.nomeProprietà`
- Per aggiungere una nuova proprietà ad un oggetto basta assegnarle un valore  
`obj.nome = "Pippo";`
- Per accedere ai metodi di un oggetto:  
`object.methodName( par1, par2, ... parn)`

# Oggetti

- Un oggetto JavaScript è un array associativo
- E' costituito da coppie chiave-valore
- Le chiavi sono i nomi delle proprietà
- Esempio:  

```
obj.name = "Pippo";  
obj["name"]="Pippo";
```
- Per eliminare una proprietà si usa l'operatore delete:  
delete obj.name

# Creare Oggetti

- La definizione di un oggetto avviene implementando la funzione costruttore:

```
function Persona(nome, cognome) {  
    this.nome = nome;  
    this.cognome = cognome;  
}
```

# Creare Oggetti

- In alternativa si può utilizzare un'istruzione di inizializzazione  
`var persone = { nome: "Raffaele", cognome: "Montella" };`

es:

<https://api.meteo.uniparthenope.it/products/wrf5/forecast/com63049>

- Creazione di un'istanza vuota

```
var persona = new Persona();  
persona.nome = "Raffaele";  
persona.cognome="Montella";
```

# Metodi

- Per aggiungere un metodo ad un oggetto basta assegnare al nome del metodo una funzione precedentemente definita:

```
function setAnnoDiNascita(a) {  
    this.anno = a;  
}
```

```
function Persona(nome) {  
    this.nome = nome;  
    this.setAnno = setAnnoDiNascita  
}
```

...

```
var raffaele = new Persona("Raffaele Montella");  
raffaele.setAnno(1972);
```

# Oggetti di uso comune

- String
  - Operazioni con le stringhe
- Math
  - Costanti e formule matematiche
  - E' un oggetto statico, non occorre istanziarlo
  - Esempio:  
`var area = Math.PI * Math.pow(r,2);`
- Date
  - Gestione di data ed ora

```
var d = new Date();
```



# Istruzione with

- Permette di riferirsi ad attributi e metodi di un oggetto impostando un default
- Sintassi:  
with (oggetto) statement  
usa oggetto come prefisso all'interno dello statement
- Esempio: (istruzioni equivalenti)

```
document.myForm.result.value =  
compute (document.myForm.myInput.value) ;
```

```
with (document.myForm) {  
    result.value = compute (myInput.value) ;  
}
```

# Array

- E' un dato strutturato che permette la memorizzazione di una sequenza di valori dello stesso tipo
- Si definisce come un nuovo oggetto Array

```
var nomi = new Array(3)
```

- La dimensione non è necessariamente fissa

```
var nomi = new Array();  
nomi[0]="Raffaele";  
nomi[1]="Giulio";  
nomi[2]="Angelo";
```

# Array

- Sintassi alternativa

```
var nomi = { "Raffaele", "Giulio", "Angelo" };
```

- Definizione di array con elementi vuoti

```
var colori = { "Giallo", "Rosso", , , "Blu" };
```

- Lunghezza di un array

```
var colori = new Array(7);  
var l = colori.length;
```

# Array

- La lunghezza di un array si può modificare assegnando un valore in una posizione oltre la lunghezza corrente
- Gli array sono sparsi, è allocato spazio solo per le posizioni effettivamente avvalorate
- Esempio:  
`colori[50000] = "blu londra";`

# Array

- Non esistono array multidimensionali
- Sono possibili array di array
- Esempio:

```
colori [5] [2] = "Indaco" ;
```

- Gli indici devono essere compresi fra 0 e  $2^{32}-1$

# Array

- La classe Array ha alcune funzionalità predefinite
- `myArray.sort()`: ordina alfabeticamente
- `myArray.sort(function)`: ordina l'array in base ad una funzione di confronto fra elementi

```
myArray.sort(function(a,b) { return a - b; })
```

- `myArray.reverse()`: inverte l'ordine degli elementi

# Array

- `myArray.push(...)`: aggiunge un elemento alla fine dell'array e ne incrementa la dimensione
- `myArray.pop()`: preleva l'ultimo elemento dell'array e ne riduce le dimensioni
- `myArray.toString()`: restituisce una stringa contenente i valori dell'array separati da virgole

# Espressioni regolari

- Sintassi alternative
  - Coppia di slash: `re = /ab+c/`
  - Costruttore di oggetto: `re = new RegExp("ab+c");`
- La sintassi delle espressioni regolari è uguale a quella di altri linguaggi
- Il metodo `stringa.match(regex)` ricerca l'occorrenza dell'espressione regolare all'interno della stringa



# Espressioni regolari

- Il metodo `stringa.match(regex)` restituisce `null` se non esiste alcuna occorrenza
- Se l'attributo `g` (global search) di `regex` è impostato, allora `match` restituisce un array di stringhe.
- Se l'attributo `g` è non impostato, `match` restituisce un array in cui l'elemento `0` è il testo individuato e la proprietà `index` è la posizione iniziale della sottostringa

# Conclusioni

- Javascript è un linguaggio ad oggetti
- È basato su prototipi
- Non utilizza le classi:
  - Implementazione
  - Riutilizzo
- Classless, prototype-oriented, instance-oriented