# OGGETTI & FUNZIONI

### OGGETTI

Collezione non ordinata di proprietà/metodi, si possono memorizzare valori attraverso il nome della proprietà

In Javascript gli oggetti possono esistere anche in assenza di classi

* Sono **dinamici**
  + posso aggiungere/modifica/creare metodi in ogni momento
  + posso aggiungere/modifica/creare proprietà in ogni momento
* non ci sono metodi di controllo degli oggetti
  + proprietà e metodi sono sempre pubblici

#### Definizione di un oggetto

*Immagine che contiene testo, Carattere, schermata, tipografia

Descrizione generata automaticamenteNel caso in cui nel nome di un oggetto ci sono dei valori non permissibili da javascript, bisogna mettere il nome tra virgolette (caratteri speciali/spazio)*

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

Si può anche creare usando le funzioni costruttrici

#### Accedere ai valori delle proprietà

Si può accedere usando le virgolette o con la dot notation(in assenza di caratteri speciali)

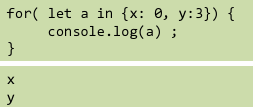
#### Aggiunta/eliminazione/modifica proprietà



*NOTA 🡪 Se scrivo movie[title] → title scritto così è una variabile quindi non funzia (tranne se esiste una variabile title che al suo interno abbia il nome di una proprietà)*

***NOTA*** *🡪 se una proprietà non è definita → undefined*

#### Iterare su oggetti

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

#### Ottenere tutte le chiavi

#### Ottenere chiavi-valore

#### Copia degli oggetti - assign

**assign**: assegna tutte le proprietà da un oggetto sorgente (movie) ad un oggetto target e restituisce l’oggetto target come risultato  
**NOTA**: assign si può anche usare per aggiungere proprietà

#### Unione di oggetti in un nuovo oggetto

#### Verifica di esistenza di proprietà in oggetti

NOTA: LE COPIE SONO SHALLOW

## FUNZIONI

Le funzioni possono avere un numero variabile di parametri

**PARAMETRI**: lista separata di elementi da virgole che possono avere dei valori di default

* Assegna valore di default se non è passato, se non c’è valore di default 🡪 undefined

arr prende tutti gli altri parametri e li mette in un array (deve essere ultimo)

#### Immagine che contiene testo, Carattere, verde, calligrafia Descrizione generata automaticamenteDichiarazione delle funzioni

* **Immagine che contiene testo, Carattere, calligrafia, verde

  Descrizione generata automaticamenteclassico**
* **function** **expression**: definisco una variabile a cui assegno una funzione
* **Immagine che contiene testo, Carattere, calligrafia, verde

  Descrizione generata automaticamenteImmagine che contiene testo, Carattere, verde, calligrafia

  Descrizione generata automaticamentenamed** function expression
* **Arrow function**: funzioni usa e getta, infatti molto usate nelle callback
  + Alcune accortezze:
    - Immagine che contiene testo, Carattere, schermata, bianco

      Descrizione generata automaticamenteSe non ho nessun parametro devo mettere parentesi tonde
    - Se voglio tornare qualcosa devo usare le return (max 1 valore)
    - *Se c’è solo una riga/istruzione, si può omettere parentesi graffe e return*
      * *Capisce in automatico che deve ritornare il risultato dell’unica operazione fatta*

#### Immagine che contiene testo, schermata, Carattere, linea Descrizione generata automaticamenteFunzioni annidate

Le funzioni interne **esistono solo all’interno** della madre

* Può accedere alle variabili esterne
* Non è visibile dall’esterno

NOTA: se una funzione è all’interno delle proprietà di un oggetto 🡪 metodo

NOTA: se è passata all’interno di un parametro di un oggetto → callback

NOTA: esistono anche:

* **Closure:** funzione interna che può anche essere usata esternamente
* **Immagine che contiene testo, schermata, Carattere, numero

  Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, numero

  Descrizione generata automaticamenteIIFE** 🡪 creo funzione ed eseguo immediatamente.
  + Non posso più richiamarla in futuro

#### Function construction

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamenteDefiniscono un tipo di oggetto

* Lettera maiuscola iniziale
* Setta le proprietà con this
* Se avessi scritto duration=0
  + Di default, se assente 0
  + Però dovrebbe essere ultimo

Per creare un’istanza di un oggetto, si usa new

## DATE

La maggior parte dei metodi dell’oggetto date funzionano in local time (nel fuso orario che gira sul nostro pc).

La formattazione delle date è quella locale; confronti tra date sono difficili.

* + Date ha diversi vincoli, usiamo librerie esterne

**DAY.js**

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteTutti gli oggetti in day.js sono immutabili

Immagine che contiene testo, schermata, Carattere, numero

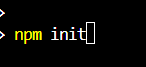
Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamenteEsistono **plugin esterni** che si possono aggiungere:

* Anno bisestile
* .isBetween
* .isBefore .isSame .isAfter
* .daysInMonth

Metodi per modificare una data

* .add
* .substract
* .startOf
* .endOf
* d1.diff(d2, ‘unit’)

**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteesempio**

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente→

Con common js → file.js

Nota: l’orario è quello inglese (1 ora in meno rispetto ad italia; $M è due perché parte da 0)

→

Nota: con ES module → .mjs e non serve specificare ‘use strict’;