OGGETTI

Gli oggetti sono variabili, ma possono contenere molti valori di tipi diversi.
Un oggetto è una collezione di coppie, il cui ordinamento non è garantito.

(**proprietà : valore**) oppure (**chiave : valore)**
La proprietà/chiave è una stringa, il valore può essere qualsiasi cosa.

La sintassi di un oggetto è questa:

```
let nomeOggetto = {
      ‘proprietà’ : “valore” ,
      ‘proprietà-2’ : “valore-2”,
      ‘proprietà-3’ : “valore-3”,
}
```

Esempio di oggetto:

```
let persona = {
     'nome' : "Massimiliano",
     'cognome' : "Falcone",
     'skill' : ["html" , "css" , "javascript"],
     'saluto' : function(){
           return `Ciao sono ${this.nome}`
      }
     'caratteristiche' : {
           'occhi' : "castani",
           'capelli' : "castani",
           'altezza' : 1.73,
      }
}
```

Come da esempio, vediamo che negli oggetti è possibile annidare qualsiasi cosa:
- *Stringhe (es. “Massimiliano”)*
- *Numeri (es. 1.73)*
- *Array (es. [“html”, “css”, “javascript”]*
- *Oggetti (es. ‘caratteristiche’ : { … })*
- ~~*Funzioni*~~ ***Metodi*** *(es. function() { …])*

Le ~~***Funzioni***~~ *negli oggetti si chiamano* ***Metodi.***
All’interno delle ~~*funzioni*~~ *Metodi se dobbiamo riferirci ad un elemento dell’oggetto, ci riferiamo al nome dell’oggetto con il termine* ***this****.*

*Nell'esempio di prima con this ci riferiamo a **persona**.*

Per accedere ai dati del nostro oggetto, o per invocare un metodo, usufruiremo della **dot syntax.**

```
console.log(...)
      persona.nome
            // "Massimiliano"
      persona.skill
            // " ["html" , "css" , "javascript"]
      persona.skill[1]
            // javascript
      persona.saluto()
            // Ciao sono Massimiliano
      persona.caratteristiche.altezza
            // 1.73
```

In questo modo possiamo anche aggiungere nuove proprietà al nostro oggetto:

```
persona.nazionalità = "Italiana";
      // nazionalità : "italiana" verrà aggiunto all'oggetto
        persona
```

"COPIARE" UN OGGETTO

```
let altraPersona = persona
console.log(altraPersona)
// l'oggetto luca avrà tutte le proprietà di Marco

altraPersona.nome = "Luca"
// in questo modo modifichiamo il nome di altraPersona ma anche
di persona
```

Possiamo copiare un oggetto creandone una coppia identica come nell'esempio 1, ma quando poi modifichiamo una proprietà dell'oggetto appena creato, modificheremo anche la proprietà dell'oggetto vecchio.
Questo succede perché quando copiamo un oggetto non creiamo una copia, ma **creiamo una chiave** per accedere all'altro oggetto.