

## Come scrivere codice come una persona del terzo millennio, così Andrea è contento!

- ***Lo spazio è codice per la mente***

Spaziare il codice può sembrare una cosa inutile ma è un ordine tipografico importante. Orizzontalmente si può indentare il codice, cioè metterlo su piani diversi orizzontalmente, è fondamentale per

- Leggere il codice più velocemente poiché si conoscono sempre lo scope in cui si sta lavorando
- Bello da vedere

Verticalmente abbiamo l'accapo.

Spaziando verticalmente il codice per il *contenuto* che fanno, ci permette di avere una più rapida velocità di controllo del codice.

Codice scritto senza spazi:

```
funzione test(cose,cosa,cosi){
creo variabili;
creo cose;
for condizioni{
faccio altra roba
}
leggo roba;
scrivo roba;
scrivo roba;
scrivo roba diversa;
scrivo roba diversa;
scrivo roba;
ritorno;
}
```

Codice scritto con gli spazi:

```
funzione test(cose,cosa,cosi){

    creo variabili;
    creo cose;

    for condizioni{
        faccio altra roba
    }

    leggo roba;

    scrivo roba;
    scrivo roba;
```

```

    scrivo roba diversa;
    scrivo roba diversa;

    scrivo roba;

    ritorno;
}

```

- ***Dare nomi è difficile, darli sensati ancora di più***

La pigrizia è alta.

Quando diamo nomi a variabili, funzioni e simili è importante dargli nomi che spiegano già cosa sono.

Un esempio? Chi tra questi due fa la media e di cosa? *funzione  $m(x)$*  oppure *funzione media(listaNumeri)* ?

A vostro figlio daresti mai il nome  $f$ ?

- ***Commentare con criterio***

Se avete spaziato il codice già evidenziato i vari blocchetti e divisi per il cosa fanno.

Ora è il momento di commentarli! Il motivo è semplice, poter permettere ad altre persone di capire cosa abbiamo fatto.

Rispettando le regole sopra, come fare?

```

//
// funzione che fa il test di cose cosa cosi
//

```

```

funzione test(cose,cosa,cosi){

```

```

    // creo variabili

```

```

    creo variabili;

```

```

    //creo cose

```

```

    creo cose;

```

```

}

```

- Una funzione deve avere un commento tutto suo. Un commento più bello degli altri! Possiamo raccontare cosa fa la funzione, gli input che ha... gli output... e perché no... anche qualche esempio se serve!

Leggendo questo paragrafetto, dobbiamo esser in grado di capire esattamente cosa fa la funzione. **NON** come lo fa.

```

//

```

```
// test
//
// Input   :
//      cose    : intero, numero di denti nella bocca
//      cosa    : lista di nomi delle dita della mano
//
// Output  :
//      Nulla
//
// test è una funzione che non fa nulla.
//
// Esempio :
//      test(1,["mignolo"]) --> Nulla
//

funzione test(cose,cosa){
```

- Commentare a blocchi.  
Se creiamo variabili che hanno dei nomi autoesplicativi (quella roba lì), è inutile scrivere un commento per ogni riga di codice.  
Va bene spiegare ma è da fare per parti di codice.  
Ogni volta che si può generalizzare qualcosa per spiegarlo. Ogni volta che un commento spiega del codice per bene.
- Commenti lunghi, se serve.  
Avete un passaggio che è una riga di codice ma è il punto focale di tutto l'algoritmo? Un bel commento, più lungo, con esempi, disegni e tutto il possibile per spiegarlo. Non vogliamo che la parte più bella dell'algoritmo sia considerata una pezza.
- Commenti anche su più righe.  
I commenti non pesano nulla, spaziano e dividono ancora di più il codice. Commentare è uno spazio, alla fine.

- ***La forma è tutto. Con una bella estetica è più facile lavorare sul contenuto.***  
Se siete arrivati qui rispettando tutte le regole, il vostro codice (opportunamente colorato), sembrerà un quadretto post-moderno-underground molto bello da vedere e leggere.  
Se non è la sensazione che sentite, non avete seguito bene le regole di sopra!