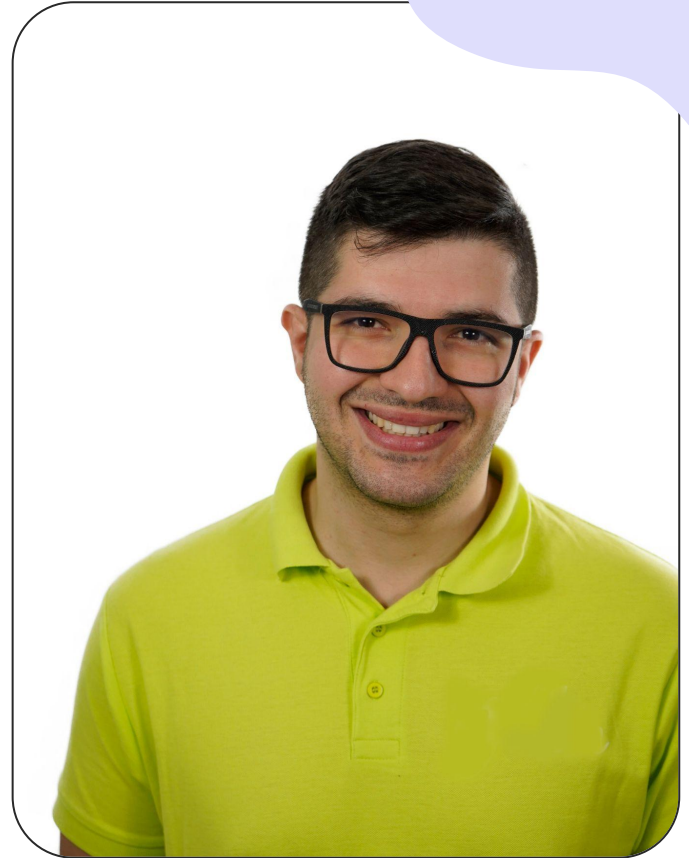


# FRANCESCO GRASSO

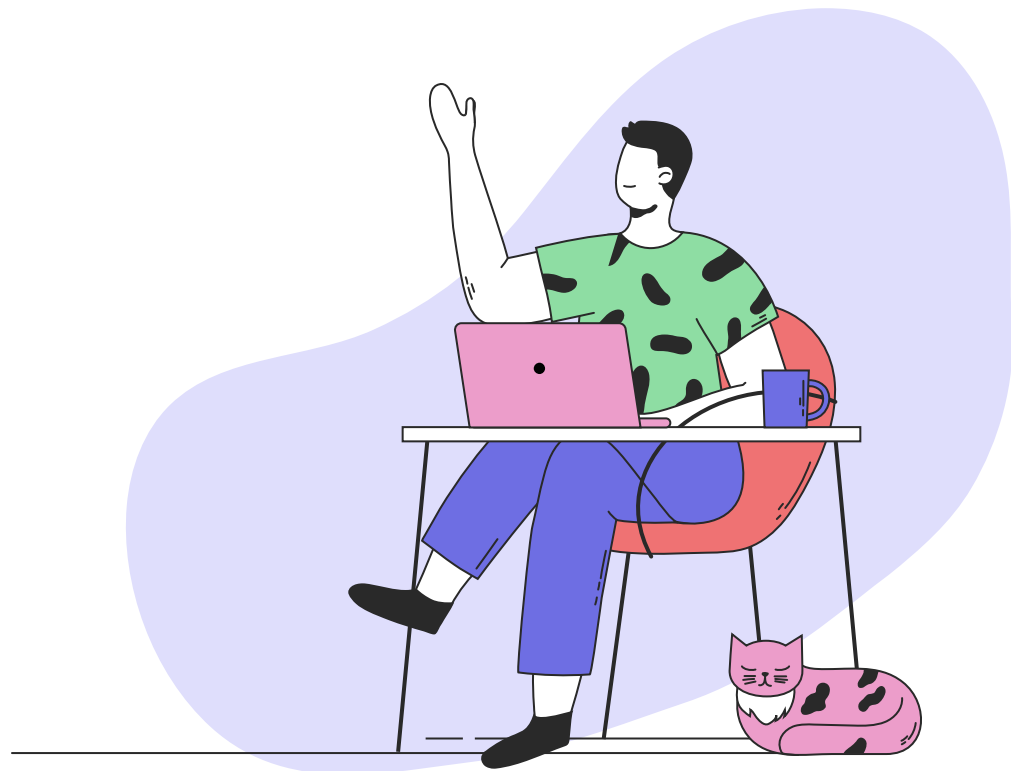
- Developer @YITH
- WordPress Meetup Catania Organizer
- WordCamp Catania 2019 Organizer
- @francgrasso

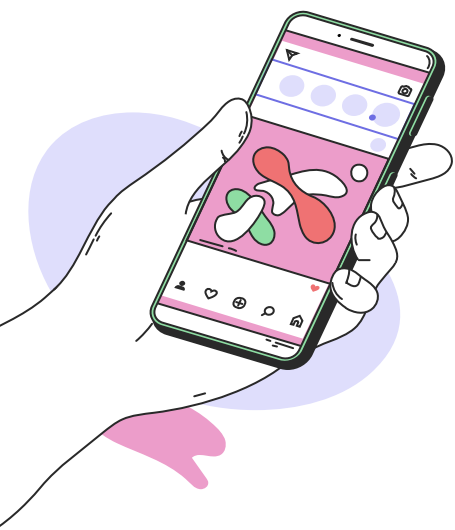


# CHILD PLUGINS

Cosa sono e come realizzarli

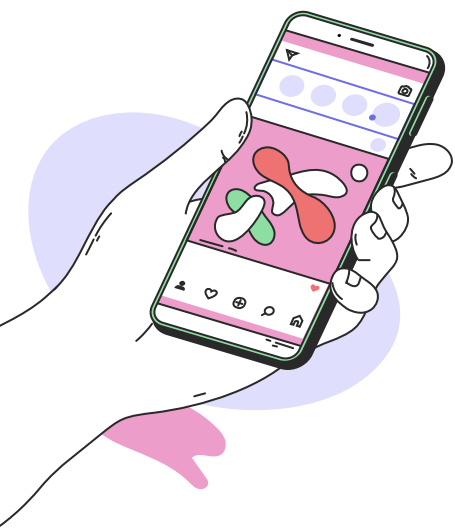
Francesco Grasso





# CHILD THEMES

Un concetto noto a quasi tutti gli utilizzatori, sviluppatori e “configuratori” di WordPress.



# CHILD THEMES

Seguono un'impostazione fissa e ben definita da WordPress e dal tema di partenza (gerarchia dei template e struttura del parent).

Hanno una sezione nel file principale del child che identifica il tema di partenza, impedendo l'attivazione qualora il "parent" (genitore) fosse non disponibile o danneggiato.

Mettere codice qui

## CHILD THEME

headers del file style.css del tema child.

# ESIGENZE IN AMBITO PLUGINS

01

## AGGIUNGERE FUNZIONALITÀ

Un plugin a noi molto utile manca di una funzionalità chiave

02

## AFFIDABILITÀ

Evitare di perdere queste aggiunte al primo update del plugin

03

## TEMPO

Inventare tutto da zero o partire da qualcosa già funzionante ed estenderlo?

04

## SICUREZZA

Aggiungere ulteriori regole/procedure di sicurezza al codice

# DA COSA È COMPOSTO

E' a tutti gli effetti un normalissimo plugin per WordPress! File php, js, css, immagini...tutto il necessario al suo funzionamento.

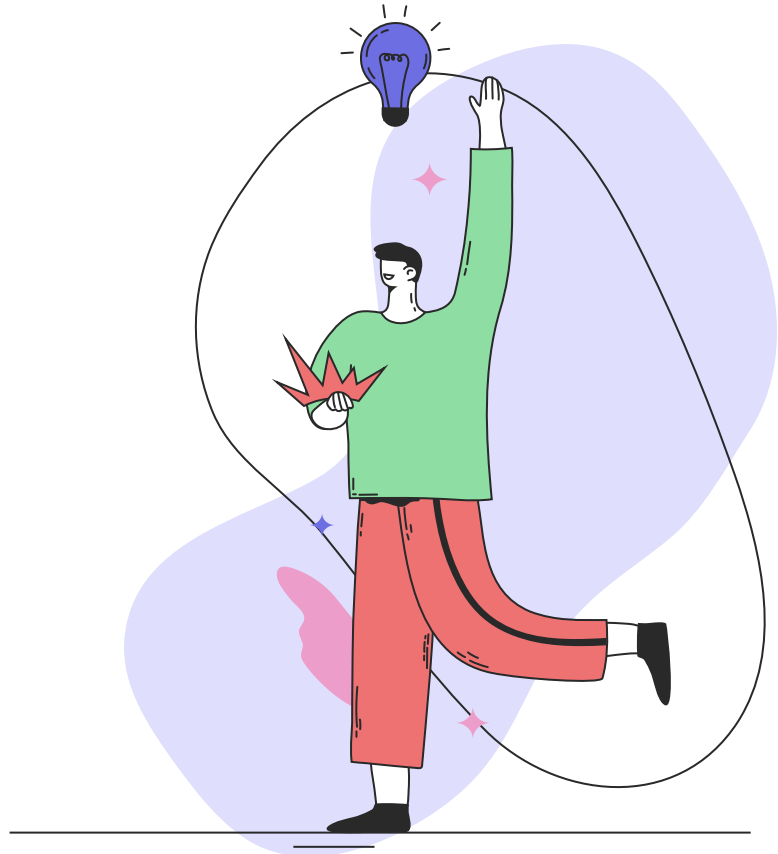
Quindi, come ogni plugin, ha un file di “avvio” contenente una heading section che ne dichiara le generalità (nome, descrizione, autore, versione, ecc).

Assicurati ovviamente, quando realizzi il tuo child plugin, che questo non abbia lo stesso nome del plugin di partenza.

## NOTA:

Non è richiesto un esplicito riferimento al plugin di partenza nella heading section del plugin (contrariamente a quanto avviene nei temi).

# COME REALIZZARLO



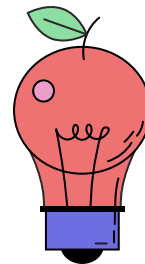


# PREREQUISITI

## !!! ATTENZIONE !!!

Il plugin di partenza deve essere scritto rispettando i più basilari standard di programmazione ed i suggerimento del modello di sviluppo WordPress.

- Se una funzione/metodo restituisce un valore, questo deve avere un filtro.
- Se il plugin fa uso di template, questi devono essere richiamati secondo una procedura che prevede un override (vedi WooCommerce).
- All'interno dei template prevedere sempre delle action per “iniettare” nuovi elementi.



# È SOLO... ...UN PLUGIN!

Lo crei esattamente come qualsiasi altro plugin.

# DA COSA È COMPOSTO

Mettere codice qui



**COME  
PROCEDERE**



# TECNICHE A DISPOSIZIONE

**HOOKS**

**OVERRIDE  
FUNZIONI/METODI**

**OVERRIDE  
TEMPLATE**

**FORK**

# HOOKS

## **ACTION HOOKS**

“Fai qualcosa qui”.

Possono completamente stravolgere il normale flusso della procedura.

## **FILTER HOOKS**

Si usano per “modificare” il valore di ritorno di una procedura.

Accettano uno o più parametri in ingresso.

Non modificano il normale flusso della procedura.

# ESEMPIO ACTION HOOKS

Mettere codice qui



# ESEMPIO FILTER HOOKS

Mettere codice qui





# !!! IMPORTANTE !!!

## USA SEMPRE

```
if ! function_exists( 'nomefunzione' ){  
    function nomefunzione(){ ... }  
}
```

## OPPURE

```
if function_exists( 'nomefunzione' ){ ... }
```

# OVERRIDE FUNZIONI/METODI



**DA UTILIZZARE SOLO SE IL PLUGIN PARENT NON SFRUTTA CORRETTAMENTE HOOKS**

E' il metodo più furbo e allo stesso tempo “sporco” per sovrascrivere logica e funzionamento di una funzione del plugin (o una classe).

# ESEMPIO PLUGIN SENZA HOOKS

Mettere codice qui



# ESEMPIO OVERRIDE FUNZIONE

Mettere codice qui



## PRO

- Anche senza gli opportuni hook il plugin viene modificato dal child
- Non è necessario apportare modifiche al parent (anche se ne sei l'autore)

## CONTRO

- Ad ogni update del parent devi verificare che la funzione “sovrascritta” sia ancora utilizzata in quel modo
- Il “fatal error” è dietro l'angolo
- Potresti essere costretto a scrivere un sacco di codice e riscrivere mezzo plugin

# OVERRIDE TEMPLATE



**PRATICA POSSIBILE SOLO SE IL PLUGIN È PREDISPOSTO PER QUESTO.**

Alcuni riferimenti:

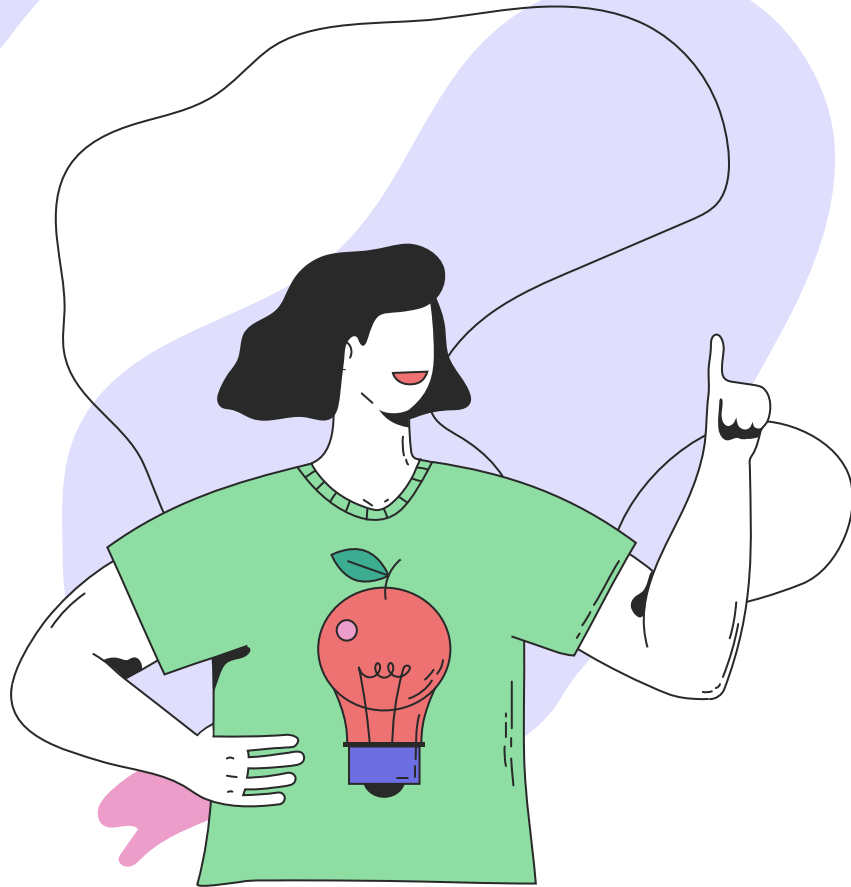
- [https://docs.woocommerce.com/wc-apidocs/function-wc\\_locate\\_template.html](https://docs.woocommerce.com/wc-apidocs/function-wc_locate_template.html)
- [http://hookr.io/functions/wc\\_locate\\_template/](http://hookr.io/functions/wc_locate_template/)

# FORK DEL PLUGIN PARENT

## DA CONSIDERARE SE NON È POSSIBILE SFRUTTARE HOOKS E SOVRASCRIVERE TEMPLATE

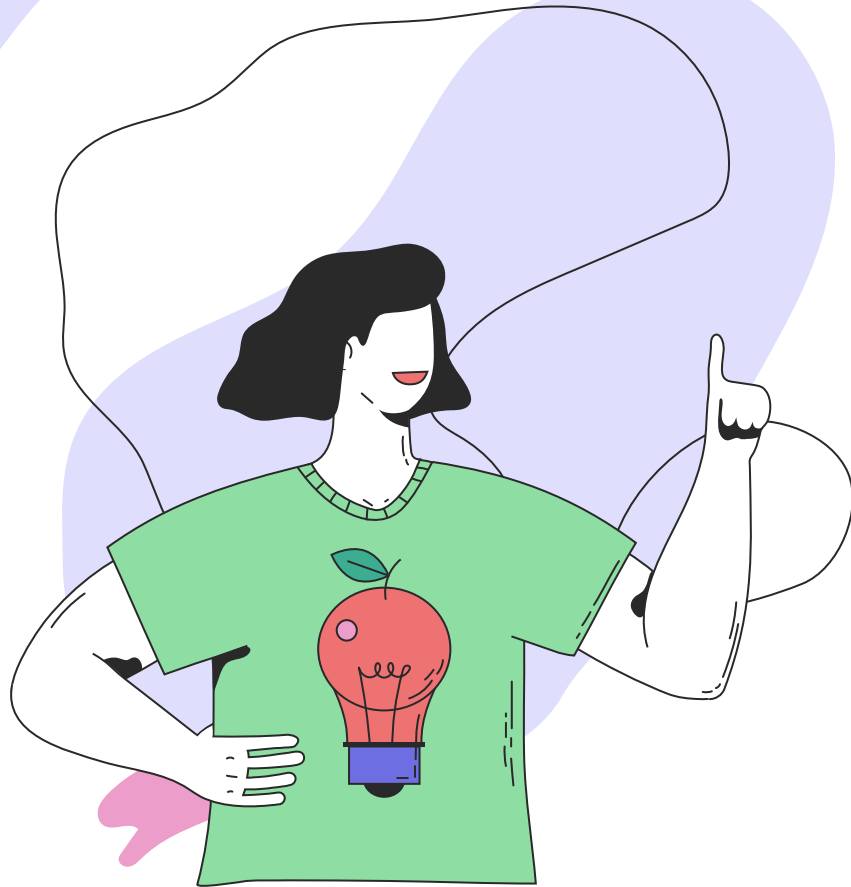
Come agire:

- Assicurarsi di avere sotto controllo di versione il parent (GIT/SVN)
- Creare un branch
- Rinominare il plugin e apportare le modifiche
- Attribuire il codice di partenza all'autore del parent
- Sottoporre le tue modifiche all'autore del plugin per eventuali integrazioni tramite pull request (se sotto git)



**QUINDI...**





**QUINDI...**  
**...I CHILD PLUGIN**  
**ESISTONO**

# FINISH!

Domande?

facebook:

<https://www.facebook.com/francesco.aci>

twitter:

[https://twitter.com/franc\\_grasso](https://twitter.com/franc_grasso)

slack:

francgrasso



CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#) and infographics & images by [Freepik](#)