LARAVEL

Laravel e' un framework Php con una architettura MVC *(Model View Controller)*.
Modelli (Classi di Php) e *Vista (Pagine Web)* si interfacceranno direttamente al *Controller (Logica)*.

< Plug In:
- Laravel Blade Snippets
- PHP Namespace Resolver
- File Utils

D'ora in poi per refreshare utilizzeremo la combinazione ***CMD+Shift+R />***

Per creare un nuovo progetto con Laravel dopo averlo correttamente installato diamo il comando nella Bash:

```
laravel new nomeProgetto
```

Così facendo si creerà una cartella al cui interno verranno scaricati dalla Cache tutti i file necessari a Laravel a meno che non siano da aggiornare.

composer.json	*Contiene l'elenco di tutte le librerie necessarie.*
composer.lock	*Contiene l'elenco di tutte le librerie installate.*
.env	*Contiene tutte le configurazioni che daremo al nostro progetto per collegarle a servizi esterni.*
.env.example	*E' il file che Laravel utilizzerà per creare il file .env quando verrà a seguito di un pull su GitLab.*
.gitignore	*Contiene l'elenco dei file che non verranno pushati su GitLab quando si fanno progetti in gruppo.*
.server.php	*Serve per far partire un server in locale. Necessario prima inviare il comando "php artisan serve" nella Bash.*
web.php	*E' l'instradatore delle richieste che le indirizza dove devono andare.*

All'interno di *web.php* troveremo una classe *Route* con un metodo astratto *::get* e una funzione anonima.
-- Quando ottieni questo tipo di URI "instradalo" alla funzione, e ritorna la pagina chiamata 'welcome' nella cartella views.

```
Route::get('/',function(){
      return view('welcome');
});
```

CREARE UNA NUOVA PAGINA

Per creare una nuova pagina:

- Creare un nuovo Routing nel file *web.php*
- Modificare l'*URI* e il *nomedellapagina* all'interno delle parentesi.
- Creare un nuovo file con il *nomedellapagina* scelto con estensione *.blade.php*

blade è un *Template Engine* che ci permette di inserire della logica Php nel nostro Html con una sua sintassi.

CREARE UN COLLEGAMENTO

Per creare un collegamento con *href* è sufficiente inserire l'*URI* della pagina a cui vogliamo indirizzare il link.

AGGIUNGERE IMMAGINI

Per aggiungere immagini creiamo una cartella *img* all'interno della cartella public.
In essa possiamo mettere le immagini.
Possiamo inserirle nel nostro documento tramite il percorso che occupano all'interno del progetto, esempio:

```
'img' => '/img/nome.jpeg'
```

La cartella ***public*** e' l'unica cartella visibile dal browser.

CARICARE UN TEMPLATE

Quando ci troviamo a scaricare un template come punto di partenza di un nostro progetto, troveremo spesso uno zip da scaricare con al suo interno HTML, CSS, JS.
Mettiamo le cartelle CSS e JS in Public, e l'HTML nel corpo del documento.

NB. Utilizza il plugin File Utils con CMD+Shift+P, *rinomina* per modificare in maniera intelligente il nome file.

NAMED ROUTE

Per facilitare il processo di modifica di un URI per tutte le pagine del nostro progetto utilizziamo la named Route.
All'interno del file *web.php* aggiungiamo dopo le parentesi un nome che indicherà l'URI che rappresenta:

```
Route::get('/',[PublicController::class, "homepage"])->name('homepage');
```

Adesso negli href possiamo riferirci a questo URI semplicemente con il nome scelto con la sintassi di Blade.

```
href= "{{route('homepage')}}"
```

DEBUGGING
Una modalità' per effettuare il debug e' tramite il ***dd()*** che fa un return di cio' che e' inserito nelle parentesi per controllare che ciò' avvenga.

```
{{dd(...)}}
```