Creazione tema con Local + Wordpress + Atom

1. Dopo aver creato il sito con Local, clicco SITE FOLDER per aprire il percorso del mio sito
2. Creo la cartella del mio tema nel percorso C:\Users\NomeUtente\Local Sites\NomeSito\app\public\wp-content\theme
3. Trascino la cartella del mio tema su Atom
4. Clicco col tasto destro sul nome della cartella in Atom e creo le pagine:

style.css

index.php

single.php

page.php

404.php

Header.php

Footer.php

Sidebar.php

Functions.php

1. Inizializzo il tema scrivendo in style.css il codice di commento
2. Dopo aver fatto questo, il tema è già presente su Wordpress, procedo ad attivarlo
3. Faccio il setup del tema in linguaggio PHP inizializzando i supporti e le dimensioni delle immagini standard. Se è andato tutto a buon fine, andando in Pagine e modificando una pagina a caso, in basso a destra deve apparire il riquadro con scritto IMMAGINE IN EVIDENZA
4. Facciamo apparire un menù TRADUCIBILE attraverso il comando esc\_html andando ad aggiungere una azione nella funzione di functions.php
5. Se tutto è andato correttamente, apparirà una voce menù in ASPETTO (su WORDPRESS)
6. Clicchiamo su CREA MENU e selezioniamo il Menù HEADER prima di fare SALVA MENU (prova anche ad aggiungere una pagina dal menù a sinistra per verificare che funzioni)
7. Apro il file index.html di ns4 bootstrap con ATOM e mi prendo dall’inizio fino a <body> INCLUSO copiandolo in header.php
8. Mi copio i tag di chiusura in footer.php (quindi </body></html>) aggiugendo un paragrafo per indicare che stiamo parlando del footer
9. Se tutto è andato a buon fine, in alto apparirà il menu di wordpress ed in basso il footer
10. Andiamo ad inserire gli stili ed il javascript andando in function.php
11. Inseriamo nella cartella NX sia il CSS che il JS
12. Creiamo le funzioni per inserire il JS ed CSS in Functions.php, andiamo a verificare il corretto inserimento attraverso il menu di ispeziona elemento.
13. Inseriamo adesso il menu statico realizzato in bootstrap copiando la nav da index.html, tiriamo via l’attributo fixed-top
14. Andiamo a collegarlo al menu Header creato in precedenza; andiamo in menù e modifichiamo il core di Wordpress attraverso la classe Walker
15. Cerchiamo su google bs4navwalker e salviamo il contenuto di b4navwalker.php mettendolo nella cartella nx
16. Creiamo una cartella assets e spostiamo il nav lì
17. Includiamolo nel function.php seguendo la documentazione del readme, inserendolo all’inizio del function.php, aggiorniamo il percorso in assets/bs4… .php
18. Andiamo nell’header ed incolliamo il menu dinamico sotto body
19. Cambiamo il nome del menu e di theme location in header
20. Modifichiamo l’html che abbiamo copiato da bootstrap
21. Cancelliamo il codice statico di prima
22. Salviamo e vediamo se è apparsa la navbar, inserendo anche alcune voci dal Menu di wordpress
23. Includiamo popper negli scripts di function.php (sopra js/bootstrap.min.js)
24. Da adesso possiamo inserire un drop-down nel menu di back-end di Wordpress
25. Sistemiamo un po' il codice aggiungendo qualche dettaglio alle cose fatte in precedenza in functions ed header
26. Andiamo in index.php ed inseriamo la descrizione del nostro blog come Header
27. Dopo aver creato il codice andiamo in Impostazioni -> Generali e possiamo modificare da lì la descrizione del sito
28. Creiamo un loop di articoli andando sulla documentazione ufficiale al sito <https://codex.wordpress.org/Il_Loop>
29. Prendo il codice di inizio e fine del loop e lo aggiungo sotto il div dell’index
30. Ordino il tutto creando anche il design dei miei futuri articoli
31. Vado in Wordpress per modificare l’articolo standard caricando un’immagine dentro
32. Se tutto è andato a buon fine vedremo la nostra Home con l’immagine responsive inserita nella colonna di sinistra.
33. Andiamo adesso a creare la sidebar andando in functions.php posizionandoci dopo il setup del tema
34. Se tutto è stato scritto bene, nella sezione aspetto di Wordpress apparirà la voce: Widgets
35. Andiamo in index.php ed inseriamo il codice per visualizzare la nostra sidebar
36. Spostiamo il codice della sidebar da index.php a sidebar.php ed utilizziamo la funzione get\_sidebar nell’index per riportare la sidebar laterale dove vogliamo noi.
37. Sistemiamo adesso footer.php aggiungendo un margine, un padding ed i Copyright
38. Andiamo adesso a lavorare in single.php per creare il layout dell’articolo singolo inserendo anche in index.php il link che reindirizza alla pagina dell’articolo quando si clicca sul titolo.
39. Copiamo la classe container da index a single cambiando l’exerpt in content per visualizzare tutto il contenuto
40. Aggiungiamo la possibilità di inserire commenti all’articolo aggiungendo un template dei commenti in single.php
41. Creiamo adesso una nuova pagina nella cartella chiamata comments.php ed inseriamo dentro il codice per visualizzare i commenti cercandolo su google
42. Digitiamo su google comment template wordpress e cerchiamo il codice alla documentazione ufficiale sotto la voce: Another comments.php Example
43. Cambiamo alcune righe di codice inserendo una riga contenente il numero dei commenti cercandola su google: wordpress comments number e andando qui https://developer.wordpress.org/reference/functions/comments\_number/
44. Prendiamo la riga php sotto la voce User Contributed Notes
45. Lo sostituiamo dentro l’h2 comments-title inserendo i tag per le traduzioni automatiche
46. Inseriamo il tag per le traduzioni automatiche anche alla riga 41 appena prima di comments navigation, vicino la e: esc\_html\_e cambiando il dominio da twenty… nel vostro (nx)
47. Fate la stessa cosa alla riga dopo, appena dopo previus comment link, stavolta inserendo esc\_html\_\_ cambiando sempre il nome del dominio
48. Ripetete il passaggio alla riga successiva, dopo next comment link
49. Ripetete il passaggio 50 alla riga 48 del codice. (Prima di comments are closed)
50. Andiamo ora a customizzare la nostra area commenti andando in function.php includendo style.css nella funzione nx\_styles
51. Andiamo in single.php ed indentiamo dentro una classe comments il comment template
52. Andiamo poi in style.css e mettiamo modifichiamo la classe comments cambiando bottone e margini della text area
53. Inseriamo un filtro per modificare l’area dei commenti in functions.php (voglio cambiare il bottone class\_submit)

1. Modifichiamo ora il widget di ricerca nella sidebar andando in style.css
2. Andiamo adesso a definire il template in page.php inserendo header e footer e copiandoci il container della single.php cancellando alcune cose (cancello l’href col permalink sia in single.php che in page.php)
3. Diamo spaziatura e togliamo la sidebar in page.php accentrando il contenuto ed i testi
4. Adesso creiamo un post creando una nuova categoria (news) da Wordpress, inserendo del testo ed una immagine.
5. Adesso andiamo in index.php per mostrare la categoria di riferimento per inserire i conditional tags (i tag che mostrano ad esempio solo una categoria)
6. Se tutto è andato correttamente, cliccando sulla categoria, vedremo come titolo il nome della categoria.
7. Mettiamo il link anche sull’immagine
8. Aggiungiamo un supporto adesso per i tag, aggiungiamo tag all’articolo nel menu di Wordpress
9. Per visualizzarlo nel post singolo andiamo a modificare single.php
10. Modifichiamo il temple tags per far apparire il nome del tag come per le categorie andando in index.php
11. Facciamo la stessa cosa per far apparire il testo digitato nel bottone di ricerca
12. Se tutto è andato a buon fine il titolo cambierà in base alla nostra ricerca.
13. Andiamo adesso a lavorare sulla pagina 404
14. Per evitare che il footer salga troppo usiamo una flexbox nel body andando in style.css
15. Inseriamo la classe creata nel container della 404 in modo da allungare il footer, facciamo lo stesso con page.php
16. Adesso andiamo a fare delle rifiniture partendo dal mettere il link sul logo in header.php
17. Adesso stilizziamo i commenti in style.css
18. Adesso abbiamo finito la creazione di tutte le pagine, andiamo a vedere eventuali errori in wp\_config di Wordpress.  
    Rechiamoci nelle cartelle del nostro sito ed in wp-admin apriamo wp-config inserendo la riga di debug (se c’è qualche errore verrà visualizzato con la relativa riga, attenzione, se andiamo online con il sito attraverso un dominio, questa riga VA ELIMINATA per non far vedere al cliente/visitatore l’errore e la riga)
19. Creiamo adesso il file front-page.php per la nostra home inserendo header e footer
20. Apriamo il file index.html della cartella bootstrap per trasferirne il contenuto, ma prima creiamo una categoria “in evidenza” nel back-end di Wordpress e pubblichiamo un post di test spuntando la categoria
21. Creiamo un loop di articoli con una wp-query per prendere gli articoli con una determinata caratteristica (potete trovare tutti i codici cercando su google wp\_query Wordpress ed aprendo la documentazione ufficiale)
22. Aggiungiamo le caratteristiche dell’articolo prendendo i meta tag da index.php e li riportiamo in front page.php
23. Settiamo i tag per vedere l’url dell’immagine in modo da dinamicizzare il tutto
24. Possiamo adesso andare a creare lo slider prendendo il codice da index.html (quello con commento Slider) incollandolo in front page.php eliminando i tag ol, 2 dei tre div carousel-caption dove andremo a loopare gli elementi.
25. Andiamo quindi a prendere il loop sotto e lo inserisco dentro il carousel inner
26. Successivamente prendo l’html dello slider e lo inserisco sotto a query -> the\_post andando a dinamicizzare le caratteristiche (titolo, data, etc)
27. Prendo anche l’immagine con l’url incollandola sotto query -> the\_post andando ad inserire in url dell’immagine
28. Andiamo in style.css della struttura html in bootstrap copiando le 3 righe di codice ed incollandole nel style.css del nostro tema
29. Se tutto è andato bene avremo lo slider, adesso andiamo ad aggiungere un altro post su Wordpress (articolo), impostiamo immagine e mettiamo categoria in evidenza. Il secondo post dovrebbe andare ad apparire al posto del primo, ma lo slider non girerà perché entrambi avranno la classe active impostata.
30. Andiamo in front page.php ed inseriamo un contatore per farlo girare nel while
31. Adesso voglio far vedere il testo del excerpt solo in desktop e non in mobile inserendo una classe nell’excerpt
32. Specifichiamo nella WP\_query quanti articoli vogliamo visualizzare (settiamo 3 al massimo)
33. Andiamo a realizzare ora il componente con due colonne con una WP-Query che prende il contenuto della pagina. Creiamo una pagina da Wordpress inserendo un tag more e pubblicando
34. Vediamo l’id pagina aprendo la pagina e facendo visualizza sorgente pagina dopo averla aperta, l’id si trova nel tag body
35. Andiamo adesso in Front Page e mi copio il loop nello slider (da nx\_counter) e lo incollo sotto cancellando alcune cose che non servono
36. Vado nella documentazione per mostrare la pagina tramite ID e cambiando quindi la WP\_Query
37. Salviamo e da adesso vedremo gli estratti della pagina nella front page.
38. Adesso prendiamo la struttura 2 column dall’html di bootstrap ed inseriamolo all’interno del loop tagliando alcune cose.
39. Ora inseriamo il blocco con 4 card con una wp-query che prende gli ultimi articoli di una nuova categoria che creeremo: focus
40. Creiamo poi 2 pagine con categoria focus, aggiungiamo un tag more per far vedere solo una parte del testo nella front-page, impostiamo anche un’immagine
41. Andiamo ora in front-page e creiamo il loop copiandolo dal solito iniziale e incolliamolo a fondo pagina prima del footer eliminando alcuni elementi
42. Adesso prendiamo l’html delle cards copiandolo sotto il loop e cancellando 3 card, lasciandone solo una
43. Copio il loop dentro card\_deck ed inserisco la card dentro il ciclo while sotto the\_post cambiando il contenuto e cambiando i paragrafi in div per l’excerpt, e mettendo al posto del link dell’immagine, il php dell’immagine preso da index.php, se tutto è andato a buon fine avremo creato le card.
44. Passiamo al Jumbtron creando il contenuto all’interno di Wordpress, utilizzo una pagina mettendo testo, tag more, titolo e immagine. Pubblichiamo e vediamo l’id pagina dalla sorgente.
45. Vado nel codice di front page e mi copio il loop di 2 columns inserendolo dopo cards cancellando tutto l’html in eccesso e lasciando solo i temple\_tag
46. Vado a prendere l’html del jumbtron da index.html incollandolo sotto the\_post e dinamicizzandolo cambiando il p in div nell’excerpt
47. Per l’immagine ci prendiamo il codice dallo slider di front-page e le incolliamo sotto the\_post ripulendo un po il codice e copiando il link image\_attributes[0] nella riga dell’url in jumbotron fluid
48. Adesso avremo l’immagine nello sfondo, aggiungiamo la parola author vicino l’autore. Fatto questo abbiamo terminato gli elementi della front-page. L’homepage base è completa, con questo possiamo customizzare i nostri siti web.
49. Per tradurre il tema creiamo una cartella nel tema nx chiamandola languages, qui andremo a mettere i file delle traduzioni.
50. Andiamo in functions.php nella parte di setup per avviare il setup delle traduzioni.
51. Andiamo ora a creare i file delle traduzioni con poedit. Creiamo un nuovo file poedit e settiamo lingua italiana. Poi file -> salva, andiamo nella cartella languages del nostro tema e salviamolo come it\_IT
52. In proprietà settiamo nx come nome e successivamente andiamo in percorsi sorgente, clicchiamo + e facciamo aggiungi cartelle e selezioniamo la cartella del tema.
53. Sempre in proprietà andiamo in chiavi ricerca sorgente e come parole chiave sorgente mettiamo i famosi tag esc\_html\_\_ e esc\_html\_e  
    Salviamo e facciamo aggiorna dal codice.
54. L’app troverà le stringhe, ora possiamo tradurle in italiano. Dopo averle tradotte facciamo di nuovo aggiorna dal codice. Se tutto è andato a buon fine avremo le traduzioni inserite.
55. Inseriamo i tag di html 5 per ottimizzare il tutto. Partiamo dalla front-page. Partiamo dallo slider e sostituiamo il div con il section.  
    Facciamo lo stesso nel 2 columns, card e stessa cosa nel jumbtron.
56. Andiamo in index.php e mettiamo come tag del container il tag main alla riga 23 e 48.
57. Mettiamo il main anche in single.php alla riga 3 e 33, facciamo la stessa cosa in page.php alla riga 3 e 29 ed alla pagina 404 alla riga 3 e 15
58. Nella sidebar cambiamo il div in aside
59. Aggiungiamo ora il codice della paginazione in index.php prendendo il codice basic example dalla paginazione da paginate/links wordpress su google
60. Sistemiamo le spaziature nello style.css
61. Definiamo ora gli stili per i contenuti standard di worpress
62. Adesso per testare il sito bombardandolo di articoli andiamo nel back-end di Wordpress, in strumenti, importa e scarichiamo l’importer
63. Cerchiamo theme unit test su google e scarichiamo l’xml ed importiamolo su Wordpress (non badate agli errori, dipende da Local) spuntando importa gli allegati
64. Sistemiamo il menù aggiungendo la pagina Blog ed andando in impostazioni -> lettura, impostando come pagina principale front page e pagina blog quella creata dal file xml, se tutto è andato a buon fine vedremo da sotto la paginazione per scorrere tra le varie pagine dei vari articoli (1,2,3,…,successivo)
65. Scarichiamo adesso il plugin Theme Check per controllare eventuali errori del nostro tema ed attiviamo il funzionamento da Aspetto.
66. Per sistemare alcuni errori copiamo il codice che troviamo qui <https://developer.wordpress.org/advanced-administration/wordpress/css/> alla voce Wordpress Core inserendolo alla fine dello style.css aggiungendo   
    .gallery-caption{ padding: 5px; font-size: 10px;}  
    .bypostauthor{ border-left: 3px solid #ddd; padding-left: 20px;}
67. Risolviamo gli altri OBBLIGATORI aiutandoci con l’intelligenza artificiale (elimina DS STORE da cartella CSS)
68. Adesso che la versione base del tema è completa, per alleggerire un po' il sito, creiamo la versione 1.0 del nostro sito da trasformare poi in BluePrint (una versione che importa già il tema che vogliamo sul sito)
69. Clicchiamo col tasto destro sull’installazione di local nuova e facciamo Save as Blueprint
70. Da adesso se creiamo un nuovo sito abbiamo la possibilità di impostare NX come tema da Blueprint. Creiamo quindi un Nx Design Version partendo dall’installazione Blueprint
71. Completata questa installazione avremo il sito con il nostro tema su, possiamo eliminare la V.1.0 che diventa ormai inutile.  
    Con questa versione lavoreremo per customizzare la grafica del tema base.
72. Per customizzare il nostro tema custom creeremo una tassonomia (un raggruppamento di categorie), aggiungiamo il plugin “custom post type ui” ed installiamolo
73. Clicchiamo la nuova voce del menù e andiamo su aggiungi/modifica tassonomie.   
    Settiamo come slug: home\_visibility, come nome singolare Home Visibility e plurale Home Visibilities e spuntiamo Pagine aggiungendo la tassonomia che apparirà nella tendina di fianco.
74. Scendiamo in basso e settiamo TRUE a Gerarchico per poter creare categorie e sotto categorie. Se adesso andiamo ad aggiungere una pagina dovremmo vedere il box home\_visibility
75. Se non è già settato, impostate TRUE su Mostra REST API (nel menù della tassonomia)
76. Creiamo adesso una pagina con la nuova categoria home visibility: Slider e settiamo l’immagine slider (se pesa tanto ridimensionatela con tinypng)
77. Andiamo adesso nel codice per cambiare la query e far prendere le immagini della categoria slider, non più quelle in evidenza. Aggiungiamo la query nella front-page
78. Andiamo adesso a graficizzare nello slider il nostro menù aprendo style.css sotto main-content
79. Andiamo in header e fissiamo in alto il menù (fixed-top)
80. Facciamo in modo che la navbar non venga nascosta dal menu di wordpress per non avere problemi di visualizzazione andando di nuovo in style.css
81. Stabiliamo un padding superiore per il body (affinchè il menu non stia troppo attaccato al corpo della pagina.
82. Inseriamo adesso il logo creando un folder img su atom inserendo le due immagini di logo e bottone ricerca.
83. Andiamo poi in header ed al posto di blog info mettiamo il percorso dell’immagine
84. Rimpiccioliamolo un po' nello style.css e cambiamo anche il font
85. Inseriamo adesso il font montserrat andando su Google Font, cerchiamo Montserrat e ci copiamo il link https fino a Montserrat.
86. Andiamo in functions.php ed inseriamo il link negli nx\_styles
87. Poi in style.css inseriamo font-family nel body
88. Inseriamo il menù di destra con search e icone, andando a modificare prima il nav menù dell’header inserendolo in un div
89. Andiamo a prendere il codice bootstrap-4 della documentazione navbar e prendiamo la form class a fine codice.  
    https://getbootstrap.com/docs/4.0/components/navbar/
90. Modifichiamo lo stile rendendo la search trasparente, andiamo quindi in style.css
91. Attiviamo il tutto tornando in header.php
92. Per inserire le iconcine andiamo a scaricare font-awesome.io
93. Copiamo tutta la cartella fontawesome nel tema
94. Andiamo ad inserire il link dello stile in functions.php
95. Se è andato tutto a buon fine, andiamo su Font-Awesome e copiamoci l’icons del search incollandolo nell’header dentro il tag button
96. Coloriamo il placeholder della search di bianco sempre nello style.css andando su css-tricks placeholder e copiando il codice
97. Mettiamo adesso le icone con font-awesome in header.php
98. Stilizziamole in style.css con la classe navbar-social
99. Sistemiamo adesso il menù su mobile, andiamo in style per fare le media-query
100. Sistemiamo ora il menù che durante lo scroll ha problemi di trasparenza, aggiungiamo uno script in functions.php
101. Andiamo nella cartella js e creiamo un nuovo file scripts.js ed inseriamo dentro il js per aggiungere la classe is-scrolled nel momento in cui stiamo scendendo verso il basso della pagina
102. Andiamo in style.css per impostare il colore rgba scuro con una trasparenza quando stiamo scrollando la pagina
103. Risolviamo i problemi di adattabilità del placeholder sulla versione mobile modificando prima il form nell’header e poi successivamente alcune funzionalità nello style.css
104. Possiamo salvare una blueprint della versione ottenuta fino a questo momento andando su local e cliccando col tasto destro: save as blueprint impostando il nome: Nx + Logo + FontAwesome + Google Font
105. Ora possiamo stilizzare lo slider con gli elementi che vogliamo mettendo la classe slider-big su front-page.php per renderlo personalizzabile su style.css.  
     Cancelliamo la riga contenente categoria e data (35), eliminiamo display-3 da the\_title lasciandolo solo dentro h3 e per l’excerpt utilizziamo la classe carousel-text al posto della lead d-none…  
     In carousel caption aggiungiamo text-left  
     In functions.php includiamo le grandezze dei font nella riga del font Montserrat (200,300,400,700)
106. Torniamo in style.css per modificare gli elementi
107. Torno in front-page per costruire il bottone per la call to action
108. In style modifichiamo l’hover settandolo con un colore sfumato grazie a css gradient generator https://www.colorzilla.com/gradient-editor/  
     Lasciamo solo due quadratini dei colori mettendo a sinistra #03c4b4 ed a destra #85bc0d  
     Orizzontale come orientazione e copiamo il codice nello style.css
109. Aggiungiamo gli ultimi ritocchi di stile, un trattino colorato come il bottone, posto in alto
110. In front page aggiungo un po di margine in basso alla classe carousel-caption ed alzo l’altezza dello slider
111. Sistemiamo il container del titolo dello slider
112. Rimpiccioliamo il menù quando stiamo scrollando la pagina
113. Animiamo il menu con la proprietà transition di css, quindi inseriamo le righe trovate in rete su google: transition all css
114. Inseriamole nello style css dentro una classe animate
115. Aggiungiamo la classe agli elementi dell’header
116. Ritorniamo su style.css e modifichiamo la grandezza degli elementi allo scroll
117. Sistemiamo lo slider nella versione mobile facendo una media query dello smartphone per ridurre la grandezza del titolo
118. Impostiamo anche le grandezze massime per titolo e paragrafi, sempre in style.css
119. Andiamo adesso a creare un blocco a 4 colonne sotto lo slider con pre-titolo, titolo e bottone scopri di più con il plugin advanced custom fields
120. Creiamo un campo Intro Text ed aggiungiamolo. Poi aggiungiamo una regola in fondo alla pagina dicendo: Post Type è uguale a Pagina e salviamo le modifiche, se è andato a buon fine vedremo il campo Intro Text quando creiamo una nuova pagina
121. Creiamo la pagina: Primo punto di forza con testo lungo ed aggiungiamo del testo e nell’intro text mettiamo: Intro test.  
     Prima di pubblicarla aggiungiamo una nuova categoria nella home visibilities: focus e pubblichiamo
122. Adesso andiamo in front-page, copiamo il loop dello slider ed incolliamolo sotto creando una nuova sezione con classe container ed una row tirando via il contatore e l’incremento del contatore. Eliminiamo anche le due righe delle immagini, lasciamo solo titolo e permalink. Creiamo anche una colonna con 3 spazi sotto the\_post ed andiamo ad inserire titolo e permalink lì.  
     Correggiamo anche la traduzione di Guarda il video siccome manca (sia qui che nello slider)  
     Metto poi nel loop la nuova categoria, non slider, ma focus.
123. Sistemiamo il bottone, inserendo al suo posto la classe text-link
124. Andiamo poi nello style e creiamo il nuovo modulo focus con titolo, sottotitolo ed iconcina della freccia (angle-right) presa da FontAwesome  
     Modifichiamo anche front-page aggiungendo la freccetta e modificando alcune classi
125. Aggiungiamo un po di margine alla classe focuses
126. Adesso dalla documentazione di advanced custom fields aggiungiamo l’intro  
     Andiamo qui: <https://www.advancedcustomfields.com/> e prendiamo la riga di codice: the\_field mettendola in front-page sopra the\_title cambiando sub\_heading in intro\_text
127. Andiamo in style.css per sistemare il design
128. Aggiungiamo un altro punto di forza
129. Aggiungiamo il link anche al titolo andando su front-page e stilizziamolo in style per evitare che diventi blu
130. Adesso stilizziamo il box a due colonne creando una nuova pagina: Alcuni casi di successo dei nostri servizi in azione  
     Aggiungo del testo e una nuova categoria nella Home Visibility: pre evidenza e pubblichiamo
131. Modifichiamo adesso la WP\_Query utilizzando la variabile args piuttosto che l’id della pagina
132. Stilizziamo la section adesso aggiungendo la classe two-columns in front page ed andando a lavorare nello style.css
133. Stilizziamo adesso le cards creando una nuova pagina con Titolo: Nome della case e paragrafo: Grazie al servizio XYZ siamo riusciti ad ottenere il seguente risultato, con tag more e testo riempitivo  
     Creiamo una nuova home visibilities: in evidenza e settiamo una immagine per la pagina.  
     Facciamo lo stesso procedimento per una seconda pagina
134. Adesso andiamo a cambiare la WP\_Query in front page copiandoci l’args da 2 column section, aggiungiamo la classe card-cover ed eliminamo l’immagine, la categoria e la data  
     Aggiungendo lo Scopri di più della sezione precedente sotto l’excerpt (in un paragrafo con classe card-link)  
     Copiamo poi il codice per mettere l’immagine di sfondo dallo slider
135. Eliminiamo il container delle card per far prendere tutta la larghezza alle stesse, la nuova section diventa card-deck mt-5 ed anziché card-deck mettiamo card-group
136. Inseriamo un link dentro l’immagine (card-body)
137. Andiamo adesso a stilizzare le card in style.css
138. Stilizziamo adesso un nuovo elemento Blockquote partendo dalla copia del blocco a due colonne sotto le cards in front-page
139. Creiamo una nuova pagina: Two Blockquote con testo riempitivo ed una nuova Home Visibility: post evidenza
140. Andiamo adesso a stilizzare la classe two-blockquote
141. Adesso andiamo in advanced custom fields ed apriamo i page fields, aggiungiamo un campo Second content e come tipo di campo: Editor Wysiwig. Spostiamoci sulla pagina Two Blockquote, è apparso un nuovo editor per la modifica
142. Aggiungo del testo: 2987 clienti soddisfatti negli ultimi 2 anni di attività   
     Vado a capo ed aggiungo un altro paragrafo: 116 Lavori customizzati sulle esigenze del cliente  
     Metto entrambi in blockquote ed il numero lo imposto in grassetto  
     IMPORTANTE: voglio due blockquote separati, quindi controlliamo col codice html se sono separati, se non dovessero esserlo aggiungiamo noi il tag blockquote per ognuno di essi
143. Andiamo in front-page per visualizzare il contenuto
144. Stilizziamo il contenuto in style.css

DA COMPLETARE

1. Creiamo adesso il blocco testimonials partendo dall’html di base in front-page
2. Copiamo dentro la row il loop cancellando alcune cose dell’html ed inserendo la riga dell’immagine presa da single.php con classi cambiate
3. Creiamo una nuova pagina Testimonianza con una nuova home visibilities: testimonials ed una immagine
4. Stilizziamo il blockquote adesso, prima in front-page e poi in style.css
5. Andiamo nel second content della pagina ed inseriamo il nome con un tag link mettendo un link a caso (google)
6. Definiamo il colore di default dei link nello style.css
7. Aggiungiamo le virgolette al blockquote sempre in style.css
8. Aggiungiamo il link nella front-page
9. Adesso lavoriamo sul Jumbtron, inseriamo nella section la classe call-to-action-box in front-page modificandone un po' il contenuto ed andiamo a creare la classe nello style.css
10. Creiamo ora una nuova pagina: Contattaci per una consulenza gratuita senza impegno, inseriamo testo riempitivo e tag more  
    Impostiamo una immagine in evidenza ed aggiungiamo la categoria alla home visibilities prima di pubblicare: cta box
11. Andiamo in front page e sostituiamo la wp\_query copiandoci gli args da un elemento di sopra
12. Sistemiamo le ultime cose in style.css e cambiamo il bottone in front page da read more a contact us
13. Creiamo l’ultima sezione quella delle news, copiamo il codice delle card per farla. Incolliamolo sotto il Jumbotron modificandone un po' il contenuto. Prendiamo categoria e immagini da single.php  
    (cancella la riga dell’autore dal blocco a 2 colonne ed incollala nella sezione news)
14. Andiamo adesso a stilizzarli (abbiamo una sovrapposizione di stili utilizzando la classe card di bootstrap, quindi inseriamo prima la classe card-cover nella sezione card, in modo da differenziare le sezioni)
15. Torniamo in front-page per sistemare l’html delle news
16. Andiamo a creare lo style delle news in style.css  
    Abbiamo così terminato la home.
17. Ora lavoriamo le pagine singole, siccome il tema è modulare, posso riutilizzare gli elementi che ho a disposizione, in questo caso prendo il jumbtron da front-page perché voglio mettere un’immagine a sfondo nella mia pagina singola (page.php) e modifichiamo con gli elementi che ci servono
18. Adesso andiamo a rendere il menù trasparente anche per le pagine, ma serve una modifica in functions.php per attivare una differenziazione tra front-page e tutte le altre, andiamo quindi in functions.
19. Abbiamo così aggiunto la classe navbar-transparent quando ci troviamo in un post oppure una pagina, andiamo quindi in style per sostituire tutte le classi .home con la classe .navbar-transparent (usiamo il cerca di atom). Se è andato a buon fine avremo la navbar trasparente anche sulle pagine.
20. Eliminiamo mt-5 dalla page.php per togliere il margine dalla section jumbotron e sistemiamo il codice in modo che, se la pagina non ha immagini, il menù resti blu.
21. Notiamo che il menù nella front-page è tornato blu, quindi andiamo in functions.php per aggiungere la classe navbar-transparent anche se siamo nella front-page.
22. Impostiamo il menù creando una pagina blog ed una pagina front page, settiamole nelle impostazioni come pagine predefinite per home e blog. Andiamo poi in menù ed aggiungiamo le voci blog e contatti al nostro menù.
23. Navighiamo adesso sulla sezione Personalizza, attraverso la pagina functions.php possiamo modificare i settaggi del personalizza a nostro piacimento. Lavoriamo quindi sulla functions.php, aggiungeremo una sezione Logo per cambiare il Logo
24. Attiviamo il cambio logo andando nell’header.php per rendere effettivi i cambiamenti, settiamo dunque come logo standard (se l’utente non inserisce altri loghi, il logo nx, mentre prendiamo il logo inserito dall’utente se questo lo inserisce dal personalizza di wordpress)