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)