

Part 1: Theoretical Foundations (1 ora e 5 circa)

Introduzione a Next.js, i vantaggi che ci sono utilizzando questo framework, rispetto a una tradizionale app in react, i vari pattern di rendering come SSR, SSG e ISR.

React Component Basics: Introduzione sui componenti in react, come costruirsi ed avere una libreria di componenti riutilizzabile.

Concetti di routing su Next.js.

Come funziona il sistema basato sulla navigazione e sulle folder di file-system

Docs, miglior risorsa per andare in maniera verticale sul topic. <https://nextjs.org/learn>

Da approfondire:

Come vengono gestite le pagine:

<https://nextjs.org/docs/pages/building-your-application/routing/pages-and-layouts>

Dynamic route e wildcard:

<https://nextjs.org/docs/pages/building-your-application/routing/dynamic-routes>

Next router e next link

<https://nextjs.org/docs/pages/building-your-application/routing/linking-and-navigating>

Data fetching - che vedremo nella ultima parte del workshop:

<https://nextjs.org/docs/pages/building-your-application/data-fetching>

Next image

<https://nextjs.org/docs/pages/building-your-application/optimizing/images>

Sneak peek del progetto:

Una prima occhiata a quello che andremo a costruire:

Layout generico:

- Header
- Banner
- Footer

Character panel:

Lista, Filtri, Counter, Load More, Link

Creazione repository con primo commit “init” fatto. In modo da iniziare la prima lezione dove tutti hanno un'app funzionante installata.

Verrete invitati al bitbucket di wodka, dove dovrete andare a crearvi un progetto

“4224-**nome-cognome**-rem-next”

Creazione del progetto con cli:

<https://nextjs.org/docs/pages/api-reference/next-cli>

<https://nextjs.org/docs/pages/api-reference/create-next-app>

-Pulizia dello stile iniziale dell'app, per partire da un template "plain"
-Add remotes e push nel repository creato. Potete farlo a mano, o con un client a vostra scelta.

Part 2: Static Components and Dynamic Features (3 ore circa)

Obiettivo:

Creazione di componenti statici con stile, e feature dinamiche. Definizione del layout di pagina e gestione route. Link tra pagine e base funzionante per andare a testare il fetching dinamico.

- Creazione dei primi due componenti **Header e Footer**, inseriti all'interno del layout.
- Familiarizzazione con il concetto di **componente** e sulla gestione della **file structure** di next.
- Panoramica sulla gestione dei **componenti** e su come organizzare il contenuto per gestire gli elementi della nostra pagina.
- Panoramica sulle **props**, e sulla gestione delle proprietà "children", e props specifiche dei componenti.
- Inizio prima parte di stilizzazione con **CSS Modules**. Pattern che noi utilizziamo giornalmente nel nostro workflow di lavoro.
- Primo **componente complesso** (banner), introduzione di stile in modulo, invece che utilizzare lo stile **global**. Capire come organizzare lo stile per scalare (bundle con componente e stile, quando e dove).
- Introduzione al **data fetching**. Teoria di base con **getStaticProps**, **getStaticPaths** e **getServerSideProps**. Implementeremo un fetch che visualizza una list o un conteggio in pagina.
- Concetto iniziale di **dynamic routing**. Utilizzeremo una wildcard di next.js per gestire la navigazione e il contenuto della nostra pagina.

Part 3: Advanced Dynamic Content and Finishing Touches (3 ore maybe more)

Obiettivo:

Implementazione di funzione dinamiche avanzate, incluso il "dynamic-routing", data fetching automatico e alcune funzionalità client-side.

- Dynamic data fetching, utilizzando `getStaticProps` e `getServerSideProps` per andare a recuperare dalla nostra api i dati di cui abbiamo bisogno.
- Definizione in `next.config` del dominio per poter visualizzare le immagini
- Composizione della scheda personaggio e rendering condizionale con `classnames`
- Costruzione della lista dei personaggi. Creazione del componente di lista, e componente card. Concetti di passaggio di props e introduzione del componente counter e filtro.
- Route dinamica utilizzando il parametro `[id]`. Come utilizzare `getStaticPaths` e `getStaticProps` per recuperare le singole pagine di cui abbiamo bisogno.
- Esempio con pagina creata server side.
- Hydration client-side. Come vengono gestiti i dati client-side. Come utilizzare `useState` e `useEffect`. Implementazione della funzionalità del load more.
- Layout e rifiniture finali. Controllo degli stati, build e debug. Deploy su vercel o netlify.