

# **Systemutveckling**

# **Ramverk**

**25 HP**

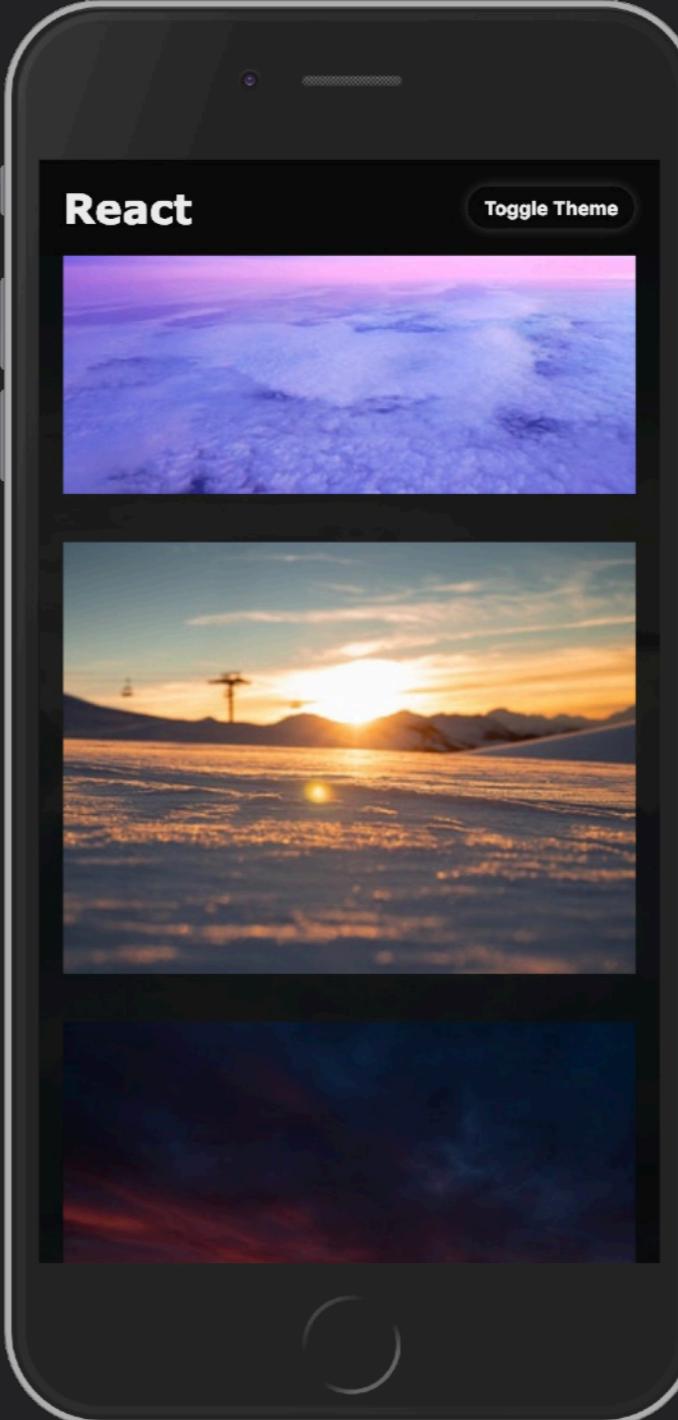
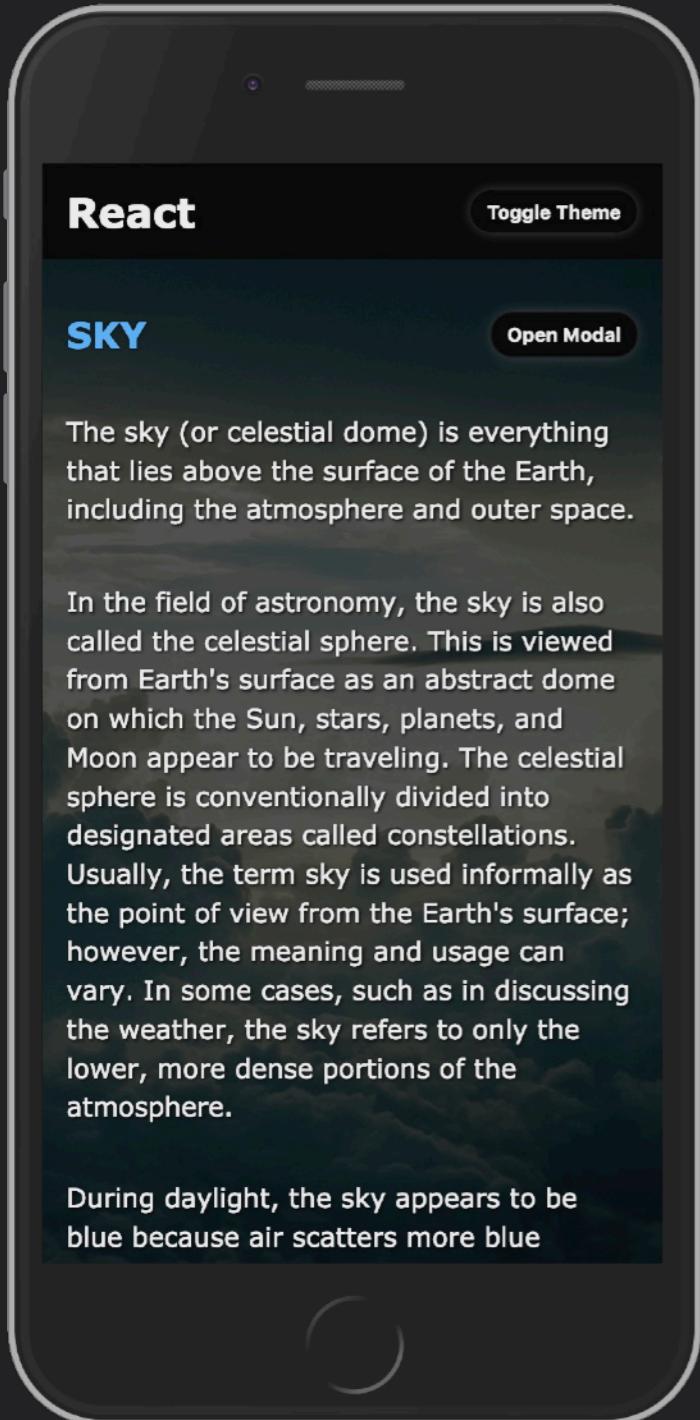
# Kursupplägget

Föreläsning			Innehåll (FM)	Övningar (EM)	Inlämning
1	mån 25 mars	D	Introduktion Typescript (Basic Types, Type Inferrence, Variable Declarations, Iterators and Generators)	TypeScripts Hemsida TypeScript in 5 minutes + övning	
2	ons 27 mars	V	TypeScript forts. (Functions, Classes, Interfaces, Modules)		
3	fre 29 mars	V	Introduktion React (SPA, Virtuell DOM, Kap. 1-6)	Reacts Hemsida Main Concepts inklusive övningar i CodePen Kodövning (RP)	Inlämning 1 ges ut
4	mån 1 apr.	D	React Playground (1-initial-setup + 2-layout)		
5	ons 3 apr.	V	React forts. (Kap. 7-12)		
6	fre 5 apr.	V	Create React App TS Intro	Övning "Todo App"	Handledning
7	tis 9 apr.	D	React Playground (3-navigation-with-state) React Playground (4-navigation-with-routes)	Kodövning (RP)	
8	ons 10 apr.	V	React Playground (5-code-splitting) React Playground (6-error-boundary)	Kodövning (RP)	Inlämning 1 lämnas in
9	fre 12 apr.	V	React Playground (7-portals)	Kodövning (RP)	Inlämning 2 ges ut
10	mån 15 apr.	D	React Playground (8-code-split-app-start)	Kodövning (RP)	
11	tis 16 apr.	D	React Playground (9-api-lib-axios)	Kodövning (RP)	Handledning
12	tis 23 apr.	D	React Playground (10-context)	Kodövning (RP)	
13	tors 25 apr.	V	Tentaplugg		Inlämning 2 lämnas in
14	fre 26 apr.	V	TENTAMEN		

# React Playground

Projektet vi kommer jobba med framöver

# React Playground



## Branches

- master
- 1-initial-setup
- 2-layout
- 3-navigation-with-state
- 4-navigation-with-routes
- 5-code-splitting
- 6-error-boundary
- 7-portals
- 8-code-split-app-start
- 9-api-lib-axios
- 10-react-context

# React Playground

**DEMO**

# Initial Setup

Hur man konfigurerar upp ett projekt på egen hand

# Initial Setup - Flera Lösningar

Det finns inte ett rätt svar i hur man sätter upp ett projekt och det finns otroligt många saker att tänka på.

Om man snabbt vill komma igång med ett React & TypeScript projekt är `create-react-app` ett mycket bra alternativ för att enkelt komma igång.

Idag kommer vi kolla på hur man kan sätta upp ett projekt från grunden. Framöver kommer vi jobba vidare i projektet när vi lär oss React.

Lets begin!

# Initial Setup - App Starting Point

Eftersom projektet byggs med **React** och **TypeScript** innebär det att vi främst kommer använda filformatet **.tsx** (typescript jsx). Därför börjar vi med filen:

index.tsx

Det enda som görs i index.tsx filen är ett anrop till **ReactDOM.render** som renderar rot-komponenten **<App/>** in i en div i html filen.

```
ReactDOM.render(  
  <App/>,  
  document.getElementById("root")  
);
```

Webpack används sedan för att bygga om **React Applikationen** (index.tsx) till en **JavaScript Bundle** (bundle.js). Bundle filen laddas sedan in i **index.html**.

```
<script src="bundle.js"></script>
```

# Initial Setup - Folder structure

Hur ska vi struktura upp projektet?

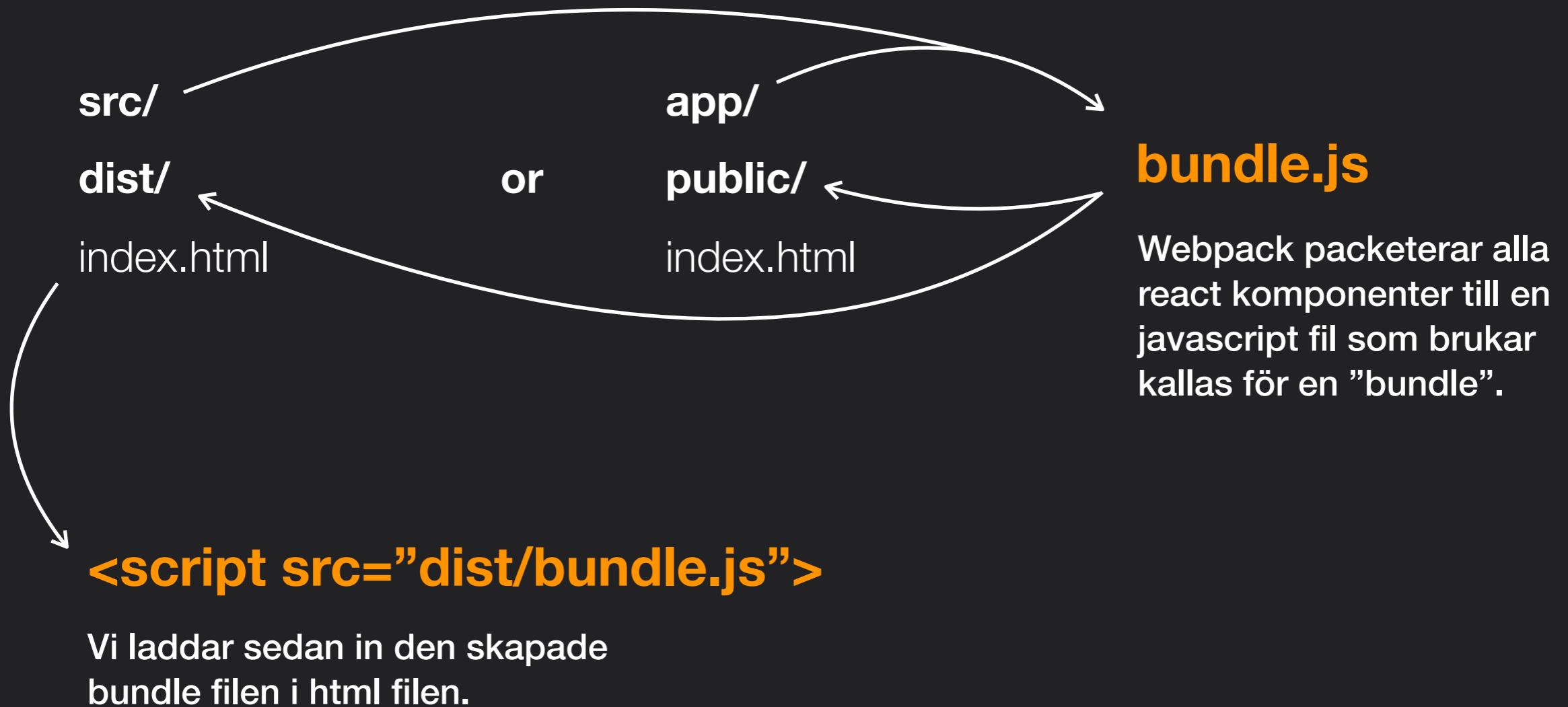
src/  
**components/**  
index.tsx  
**dist/**  
index.html

or

app/  
**components/**  
index.tsx  
**public/**  
index.html

# Initial Setup - Folder structure

Hur ska vi struktura upp projektet?



# Initial Setup - Projektfiler

**package.json (npm)**

**tsconfig.json (typescript)**

**webpack.config.js**

# Initial Setup - Projektfiler

**npm init**

package.json

**typescript --init**

tsconfig.json

**webpack-cli init**

webpack.config.js

**npm install --save react**

src/  
components/  
dist/  
**node\_modules/**  
index.html  
package.json  
tsconfig.json  
webpack.config.js



# Initial Setup - GIT

## Varför används filen `.gitignore`?

Vissa filer skapas när vi kör kommandon och behövs inte sparas på repot. Därför vill vi förhindra att git läser dessa filer.

Mapparna **dist** och **node\_modules** innehåller endast genererade filer och ignoreras genom att skriva

**dist/**  
**node\_modules/**

i `.gitignore` filen.



src/  
  components/  
  dist/  
  node\_modules/  
  index.html  
  package.json  
  tsconfig.json  
  webpack.config.js  
  .gitignore

# Initial Setup - Övning

`beginExercise();`

# Initial Setup - Övning

## TypeScript Handbook (React & Webpack)

<https://www.typescriptlang.org/docs/handbook/react-&-webpack.html>

**Konfigurera upp ett eget projekt**

Utgå ifrån TypeScripts tutorial.

**OBS: Börja med att sätta upp ett git repo**

# Initial Setup - Övning

## Att tänka på / gör även

Installera webpack lokalt i projektet.

För att kunna köra webpack CLI-kommandon behövs webpack-cli installeras (troligtvis).

Installera npm paketet live-server och se om ni kan få sidan till att ladda om sig när koden ändras. (webpack --watch)

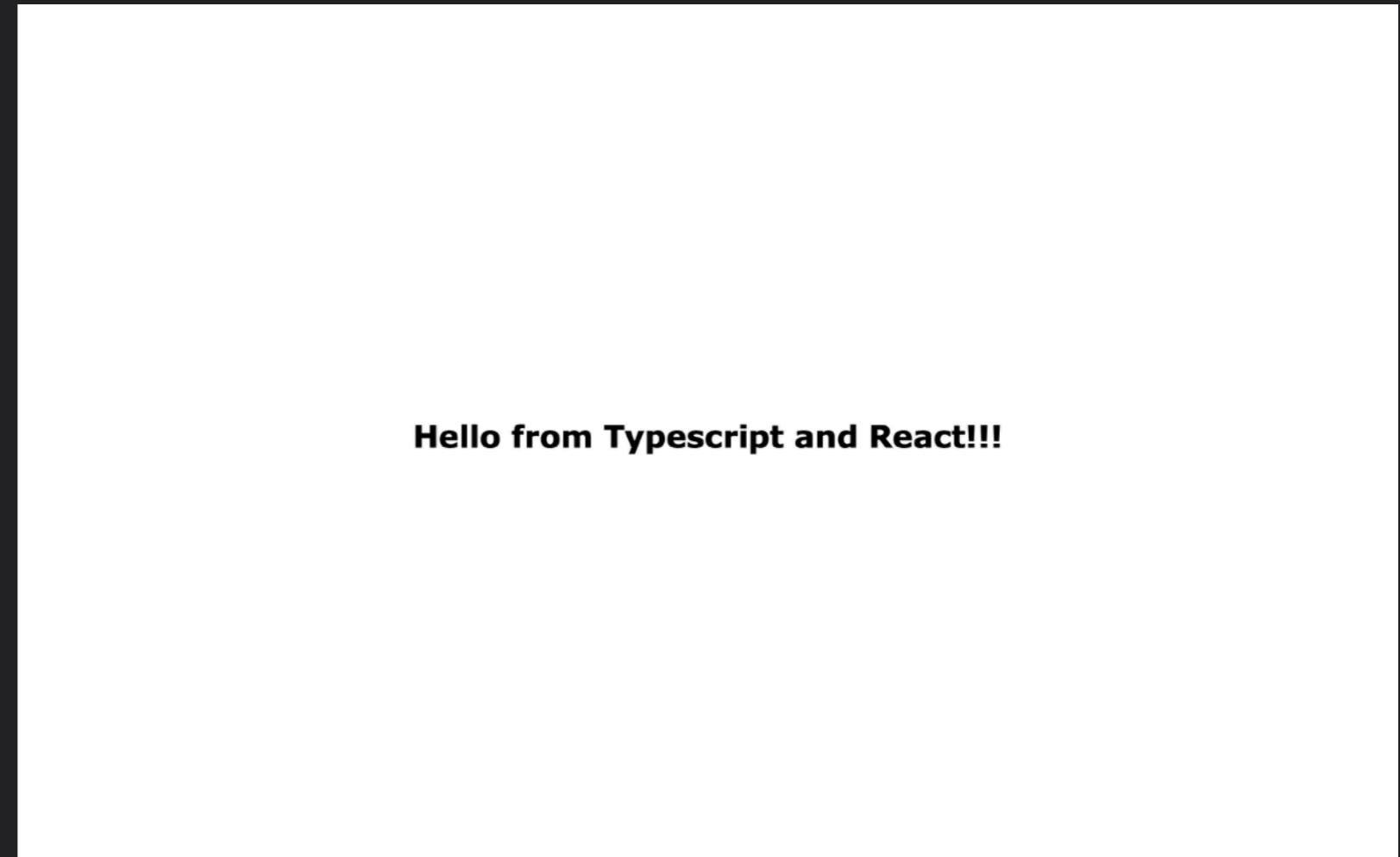
Centrera introduktionstexten horisontellt och vertikalt på er sida (använd flexbox).

# Initial Setup - Övning

Mobile



Desktop



# Initial Setup - Imports & Exports

## tsconfig.json

```
"allowSyntheticDefaultImports": true,  
"esModuleInterop": true
```

## Låter oss importera så här

```
import React, { CSSProperties } from 'react';
```

## Default exports

```
export default function App() {
```

```
import App from './components/app';
```

# Initial Setup - Live update

## package.json

```
"scripts": {  
  "app": "npm-run-all --parallel build-watch live-server",  
  "build-watch": "webpack-cli --watch",  
  "live-server": "live-server"  
},
```

Med hjälp av npm paketet **npm-run-all** kan vi köra flera program parallelt i samma terminalfönster.

Installera paketet, konfigurera upp package.json filen enligt bilden ovan och kör kommandot **npm run app**.

# Layout

Den grundläggande layouten för projektet

# Layout - Fullscreen app

Det finns många sätt att strukturera upp sin sida. Vi kommer att utgå ifrån en fullscreen layout, sidan skall alltså inte gå att skrolla. Målet är att få till en känsla av att det är en app man använder och inte en vanlig hemsida.

Det finns några olika sätt att åtstadkomma en fullscreen layout, här är ett förslag:

```
html, body, body > div {  
    height: 100%;  
    margin: 0;  
}
```

# Layout - Komponenter

## En simpel funktionskomponent

```
import React from 'react';
import Layout from './layout';

/**
 * Application top level component. This is a good place for future
 * top level components – ErrorBoundary, Provider & Suspense, etc.
 */
export default function App() {
  return <Layout/>;
}
```

# Layout - CSS i JSX

```
import React, { CSSProperties } from 'react';

/** React function component */
export default function Header() {

  return (
    <div style={header}>
      <h1 style={headerItem}>React Playground</h1>
    </div>
  );
}

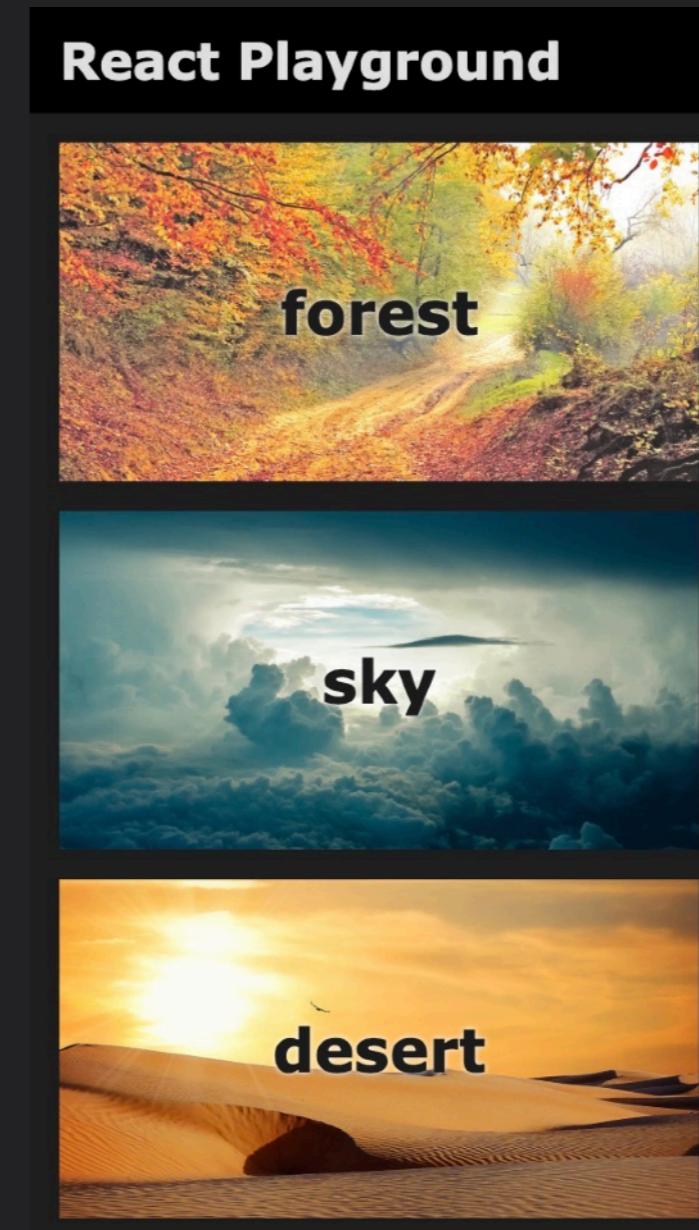
const header: CSSProperties = {
  height: '4em',
  background: 'black',
  color: '#E1E1E1',
  display: 'flex',
  alignItems: 'stretch',
  padding: '0 1em'
};

const headerItem: CSSProperties = {
  fontSize: '1.7em',
  margin: 0,
  padding: 0,
  display: 'flex',
  alignItems: 'center',
  cursor: 'pointer'
};
```

# Layout - Övning

Skapa appens grundstruktur & framsida

```
<App>  
  <Layout>  
    <Navbar/>  
    <Content>  
      <SectionItem/>  
      <SectionItem/>  
      <SectionItem/>  
    </Content>  
  </Layout>  
</App>
```



# Layout - Övning

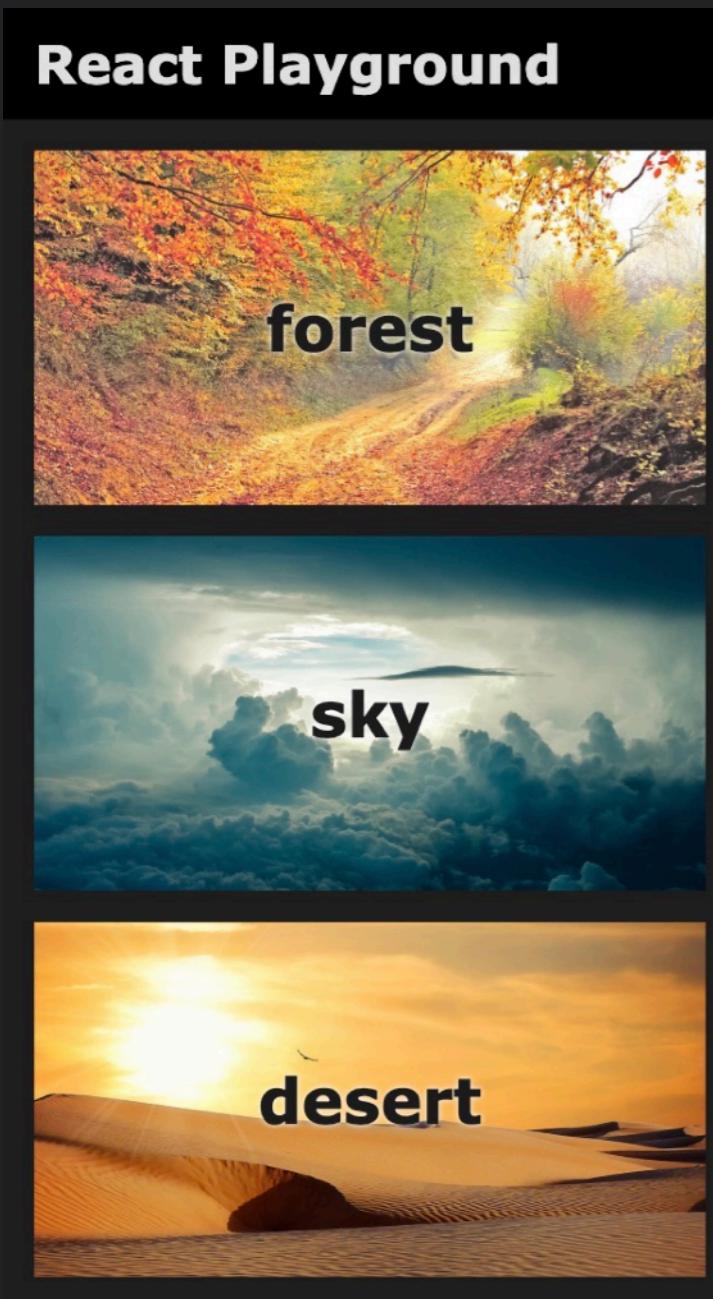
## Skapa appens grundstruktur & framsida

```
<App>                                App renderar Layout  
  <Layout>                            Layout renderar Navbar & Content  
    <Navbar/>                        Navbar renderar en Titel-text  
    <Content>                         Content renderar 3st SectionItem's  
      <SectionItem/>                  SectionItem renderar en Bild & Text  
      <SectionItem/>  
      <SectionItem/>  
    </Content>  
  </Layout>  
</App>
```

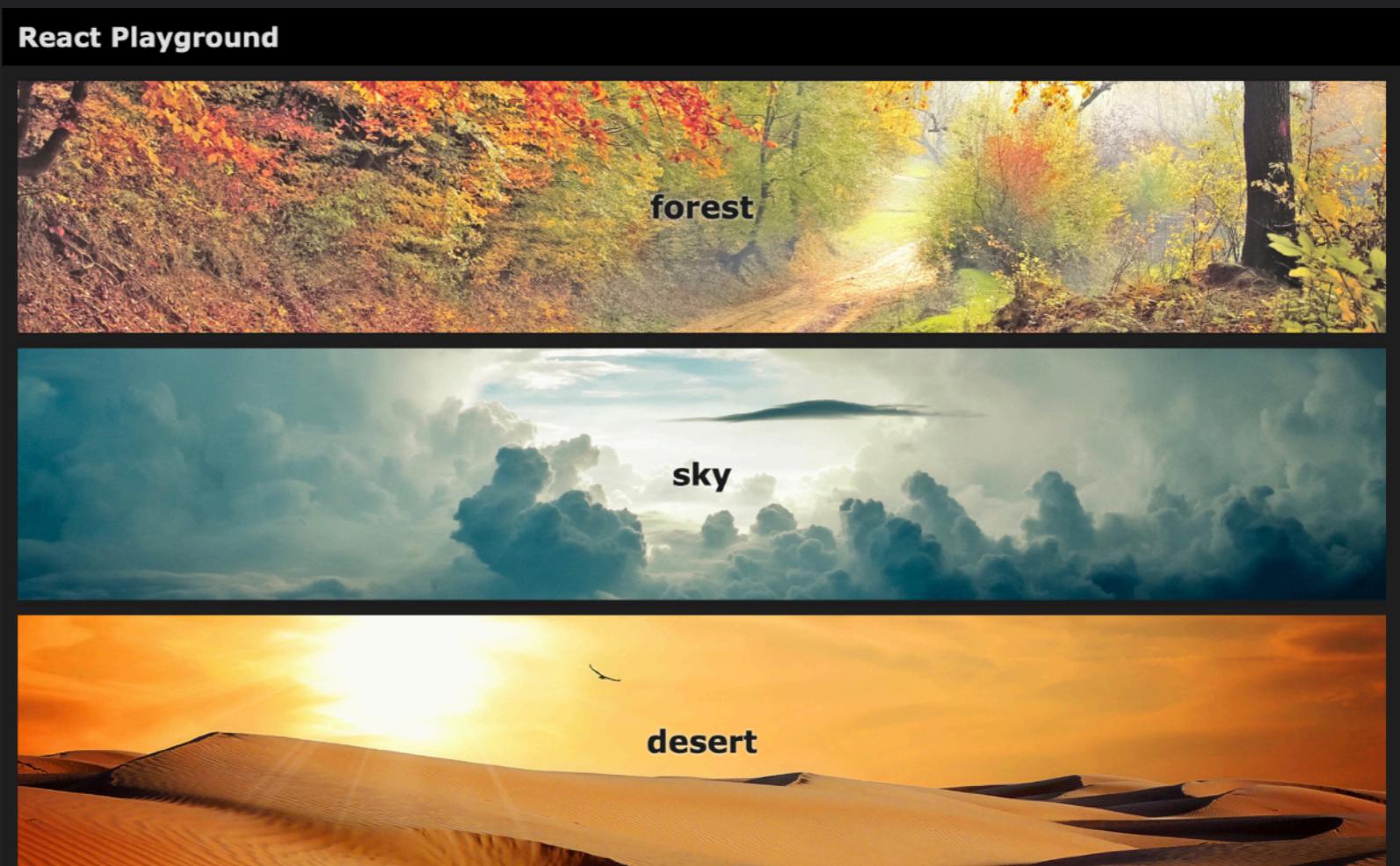
**Kom ihåg att skriva CSS'en i era .jsx filer och använd display: flex för positionering!**

# Layout - Övning

Mobile



Desktop



# Läsanvisningar

## TypeScript Handbook

<https://www.typescriptlang.org/docs/handbook>

## React Docs

<https://reactjs.org/docs/getting-started.html>

## Webpack Documentation

<https://webpack.js.org/concepts/>

# Nästa lektion

**React fortsättning**

Main Concepts - kap. 7-12

# Inlämningsuppgift

**Ta med ett ramverk till klassen**

Lämnas ut i eftermiddag på Zenit och Slack

# Tack