



INTERNATIONAL PHP CONFERENCE BERLIN 2021 - @ROWDYRABOUW

UNLEASH YOUR
WEB SKILLS
ON NATURE!



SING
RIDE
WALK

025



WEB DEVELOPER

MOOD GOASTER

**WEB DEVELOPER
IN NATURAL HABITAT**



WEB DEVELOPER

- **HTML**
- **CSS**
- **JavaScript**
- **Sass**
- **Node Package Manager**

**WEB DEVELOPER
ON NATIVE IOS / ANDROID**



WEB DEVELOPER & NATIVE

- **App stores**
- **Provisioning files**
- **Java or Kotlin for Android**
- **Objective-C or Swift for iOS**

WEB DEVELOPER WITH NATIVESCRIPT



WEB DEVELOPER & NATIVESCRIPT

- App stores
- Provisioning files
- NativeScript
- HTML, CSS, JavaScript
- Sass
- Node Package Manager

WEB DEVELOPER
LOVES
NATIVE JAVASCRIPT





AVIATOR FIGHTER

ROWDY RABOUW

- » **@rowdyrabouw**
- » **Gouda, The Netherlands**
- » **Freelance web and app developer**
- » **Lead developer Nationale-Nederlanden Pension App**
- » **Developer Expert for Nativescript**
- » **I ❤ superhero movies**







MOBILE APP FRAMEWORK DECISION GUIDE



MOBILE APP FRAMEWORK DECISION GUIDE

Do you have too much money and time?

Yes

No

MOBILE APP FRAMEWORK DECISION GUIDE

Do you have too much money and time?

Yes

No

Native iOS and Android

- Twice the work
- Java or Kotlin for Android
- Objective-C or Swift for iOS

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want/need to use web technologies?

No

Yes

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want/need to use web technologies?

No

Yes

Xamarin

- .NET or C#
- Cross compiling
- Bindings to access native APIs

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want/need a native User Interface and native performance?

No

Yes

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want/need a native User Interface and native performance?

No

Yes

Apache Cordova / Phonegap / Capacitor with Ionic

- WebView
- DOM to manipulate
- HTML **styled** like native

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want to use modern JavaScript?

No

Yes

MOBILE APP FRAMEWORK DECISION GUIDE

Do you want to use modern JavaScript?

No

Yes

Titanium

- No ES6/ES2015 support
- Can't use NPM
- Old MVC framework (Alloy)

MOBILE APP FRAMEWORK DECISION GUIDE

Do you know and like Dart?

Yes

No

MOBILE APP FRAMEWORK DECISION GUIDE

Do you know and like Dart?

Yes

No

Flutter

- Cross compiling

MOBILE APP FRAMEWORK DECISION GUIDE

Do you know and like React?

Yes

No

MOBILE APP FRAMEWORK DECISION GUIDE

Do you know and like React?

Yes

No

React Native

- React
- Bridge to access native APIs

MOBILE APP FRAMEWORK DECISION GUIDE

Do you know and like React?

Yes

No

React Native

- React
- Bridge to access native APIs
- **Hold my beer!**





NATIVE NATIVESCRIPT

WHAT IS NATIVESCRIPT?

- Open source framework for building truly native mobile apps
- JavaScript, markup (XML/HTML) and CSS
- Native code inside your JavaScript if you want and dare
- Cross Platform: one codebase for iOS and Android
- Originally developed by software company Progress

DOCS.NATIVESCRIPT.ORG

 NativeScript Framework ▾ Developers ▾ Tools ▾ Support ▾ Enterprise ▾ Blogs 🔍 Get Started Current version 6.0 ▾

NativeScript Documentation



PLAY WITH {N}

Write NativeScript code in the browser and preview on your device in seconds



INSTALL {N}

With the NativeScript CLI you can develop, build, and debug your apps locally on macOS, Windows, and Linux



SAMPLE APPS

Get real code for common app scenarios

[For Angular](#)
[For Vue.js](#)
[For JavaScript & TypeScript](#)

Looking for...

NativeScript & Angular

[Getting Started](#)

NativeScript Core

[Getting Started](#)

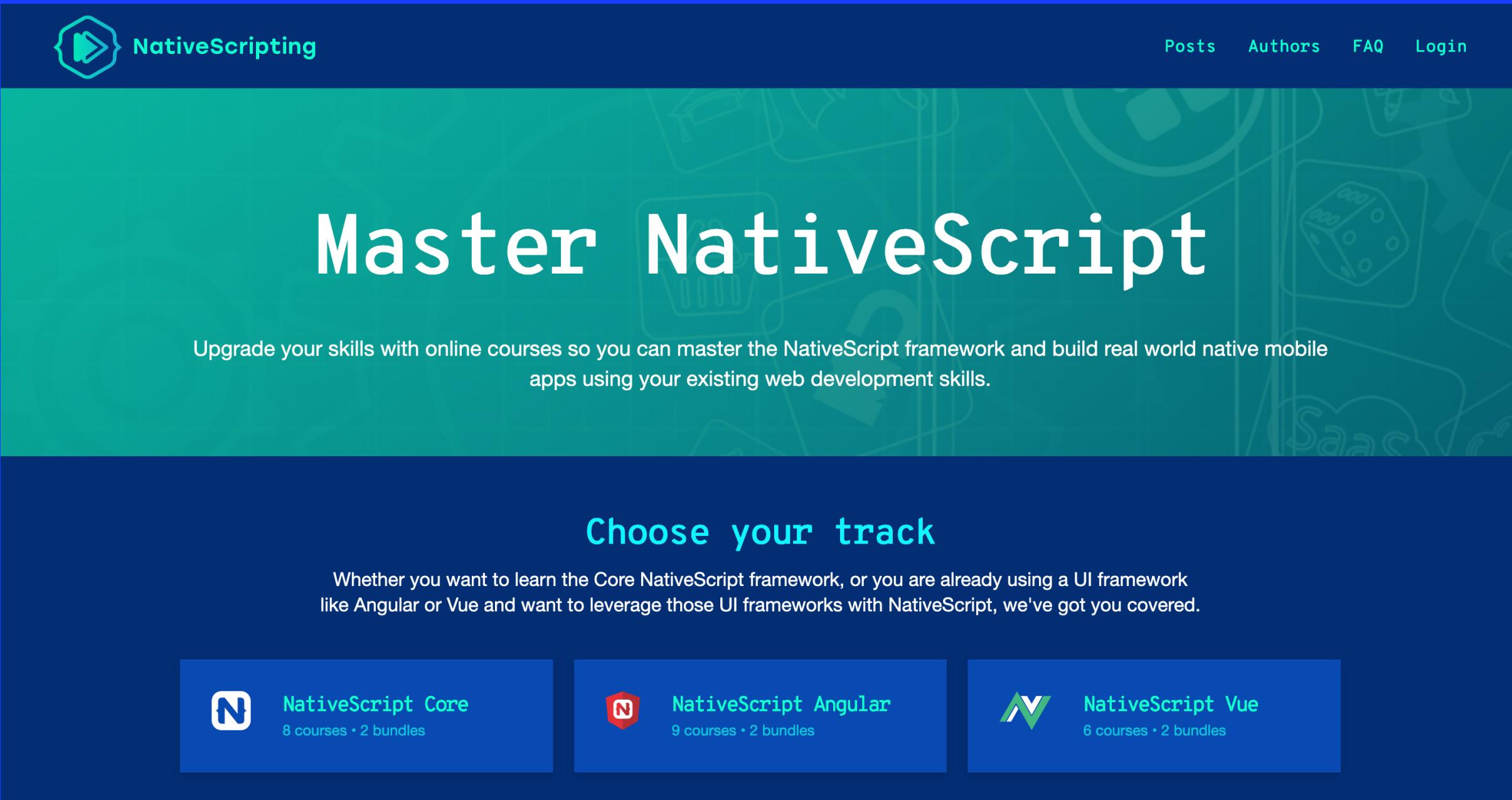
NativeScript & Vue.JS

[Getting Started](#)

NATIVESCRIPT.ORG/GET-THE-NATIVESCRIPT-BOOK



NATIVESCRIPTING.COM

The image shows the homepage of NativeScripting.com. At the top, there's a dark blue header with the "NativeScripting" logo (a play button icon inside a hexagon) and the word "NativeScripting" in white. To the right are links for "Posts", "Authors", "FAQ", and "Login". The main title "Master NativeScript" is prominently displayed in large white letters. Below it, a subtitle reads: "Upgrade your skills with online courses so you can master the NativeScript framework and build real world native mobile apps using your existing web development skills." A large "Choose your track" heading is centered below the subtitle. Three course options are listed in boxes: "NativeScript Core" (8 courses, 2 bundles), "NativeScript Angular" (9 courses, 2 bundles), and "NativeScript Vue" (6 courses, 2 bundles).

NativeScripting

Posts Authors FAQ Login

Master NativeScript

Upgrade your skills with online courses so you can master the NativeScript framework and build real world native mobile apps using your existing web development skills.

Choose your track

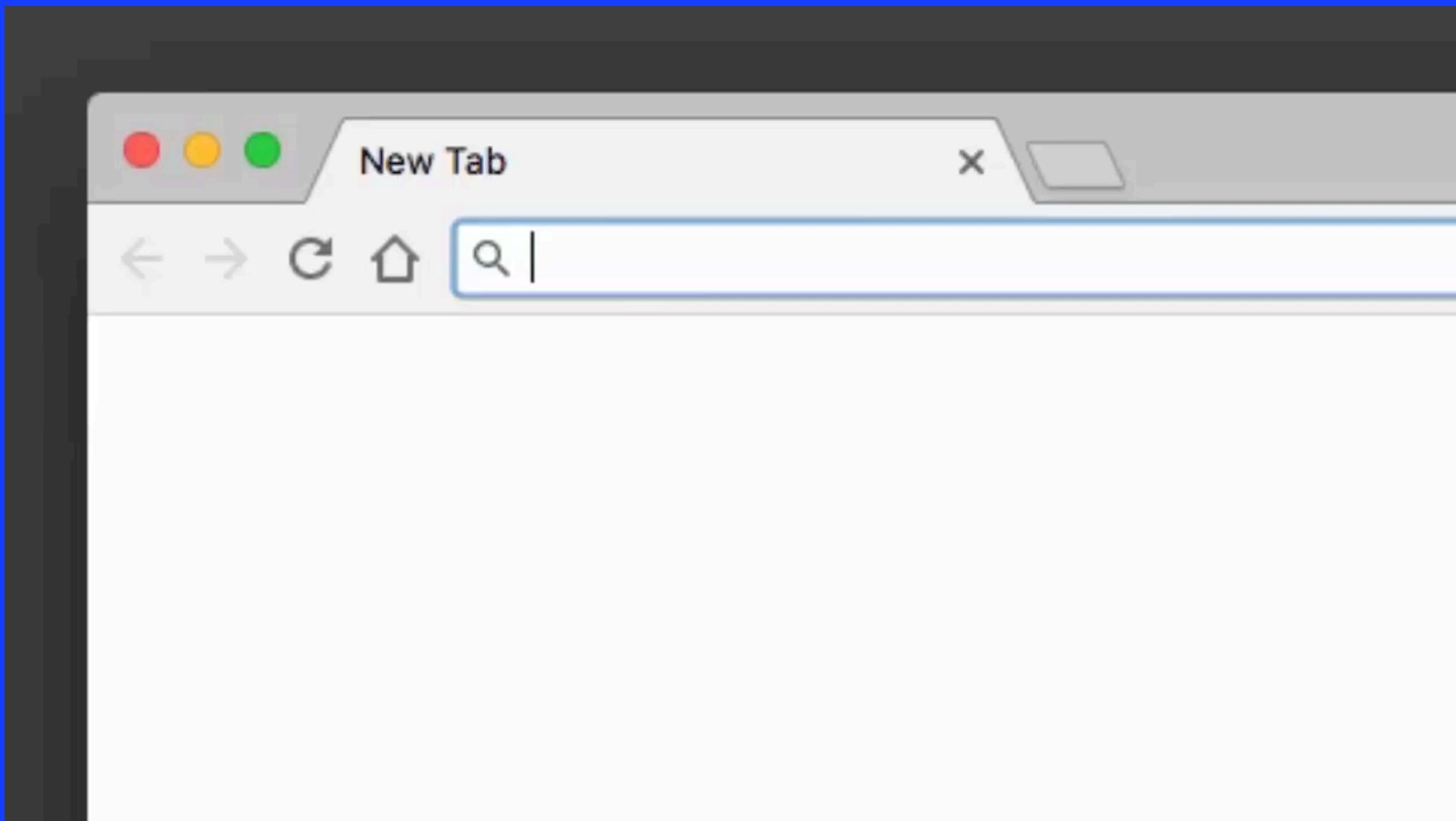
Whether you want to learn the Core NativeScript framework, or you are already using a UI framework like Angular or Vue and want to leverage those UI frameworks with NativeScript, we've got you covered.

 NativeScript Core
8 courses • 2 bundles

 NativeScript Angular
9 courses • 2 bundles

 NativeScript Vue
6 courses • 2 bundles

PLAY.NATIVESCRIPT.ORG





**MARVEL
MARKUP
LIKE ON THE WEB**



MARKUP

<https://docs.nativescript.org/ui/overview>

```
<ActionBar title="Native elements" />
<StackLayout>
    <Button text="Button" tap="{{ onButtonTap }}" />
    <Switch checked="false" />
    <SegmentedBar items="{{ segmentedBarItems }}" />
    <Progress value="0" maxValue="100" />
    <Slider value="0" minValue="0" maxValue="100" />
    <DatePicker
        year="2018"
        month="1"
        day="1"
        minDate="1970-01-01"
        maxDate="2100-12-31"
    />
</StackLayout>
```

Native elements

BUTTON



ITEM 1

ITEM 2

ITEM 3



31 Dec 2017

01 Jan 2018

02 Feb 2019

Native elements

Button



Item 1

Item 2

Item 3



29	October	2015
30	November	2016
31	December	2017
1	January	2018
2	February	2019
3	March	2020
4	April	2021

TEXTFIELD

https://docs.nativescript.org/api-reference/modules/_ui_text_field_.textfield

```
<TextField />
```

```
<TextField text="" />
```

```
<TextField hint="Enter your name" />
```

TEXTFIELD: AUTOCAPITALIZATION

https://docs.nativescript.org/api-reference/modules/_ui_text_field_.textfield

```
<TextField autocapitalizationType="allCharacters" />
```

```
<TextField autocapitalizationType="sentences" />
```

```
<TextField autocapitalizationType="words" />
```

TEXTFIELD: AUTOCAPITALIZATION

https://docs.nativescript.org/api-reference/modules/_ui_text_field_.textfield

```
<TextField autocapitalizationType="allCharacters" />
```

```
<TextField autocapitalizationType="sentences" />
```

```
<TextField autocapitalizationType="words" />
```

```
<TextField autocapitalizationType="none" />
```

TEXTFIELD: AUTOCORRECT

https://docs.nativescript.org/api-reference/modules/_ui_text_field_.textfield

```
<TextField autocorrect="true" />
```

```
<TextField autocorrect="false" />
```

TEXTFIELD: KEYBOARDDTYPE

https://docs.nativescript.org/api-reference/modules/_ui_enums_.keyboarddtype

```
<TextField keyboardType="number" />
<TextField keyboardType="datetime" />
<TextField keyboardType="phone" />

<TextField keyboardType="email" />

<TextField keyboardType="url" />
```

keyboardType

datetime

number

phone

email

url

default

keyboardType

datetime

number

phone

email

url

default

TEXTFIELD: MORE ATTRIBUTES

https://docs.nativescript.org/api-reference/modules/_ui_text_field_.textfield

```
<TextField textAlignment="" />
```

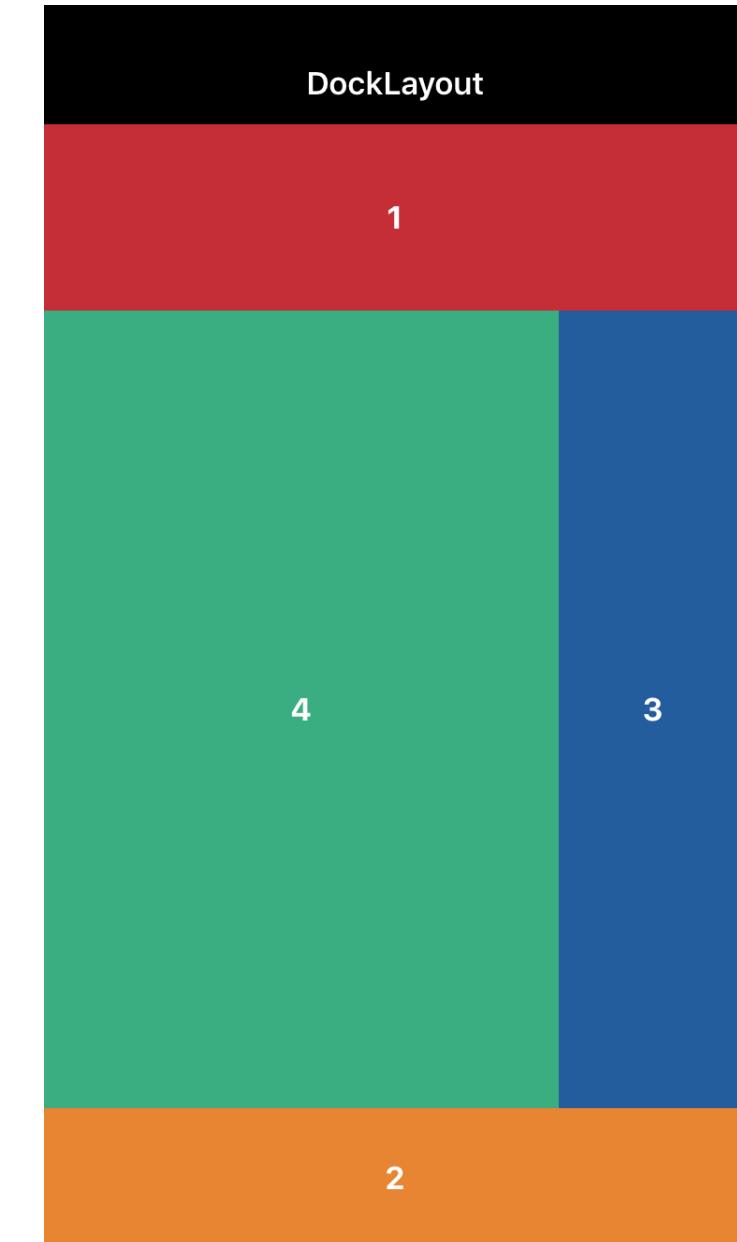
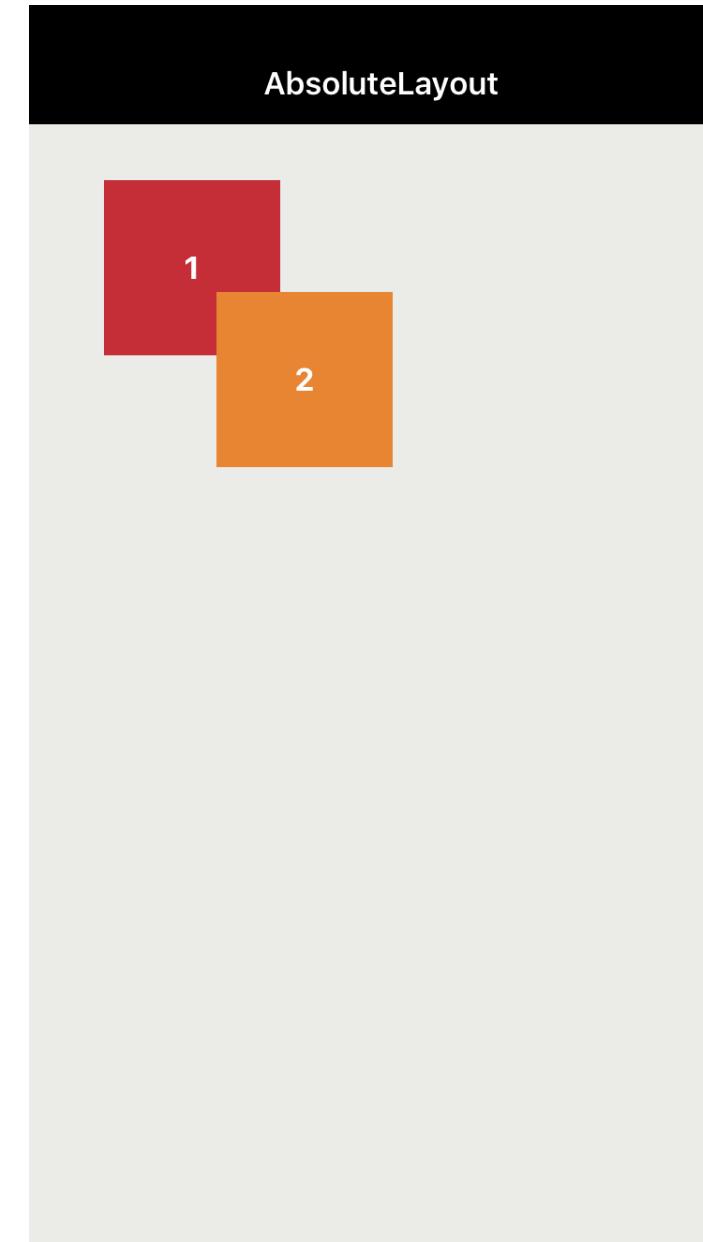
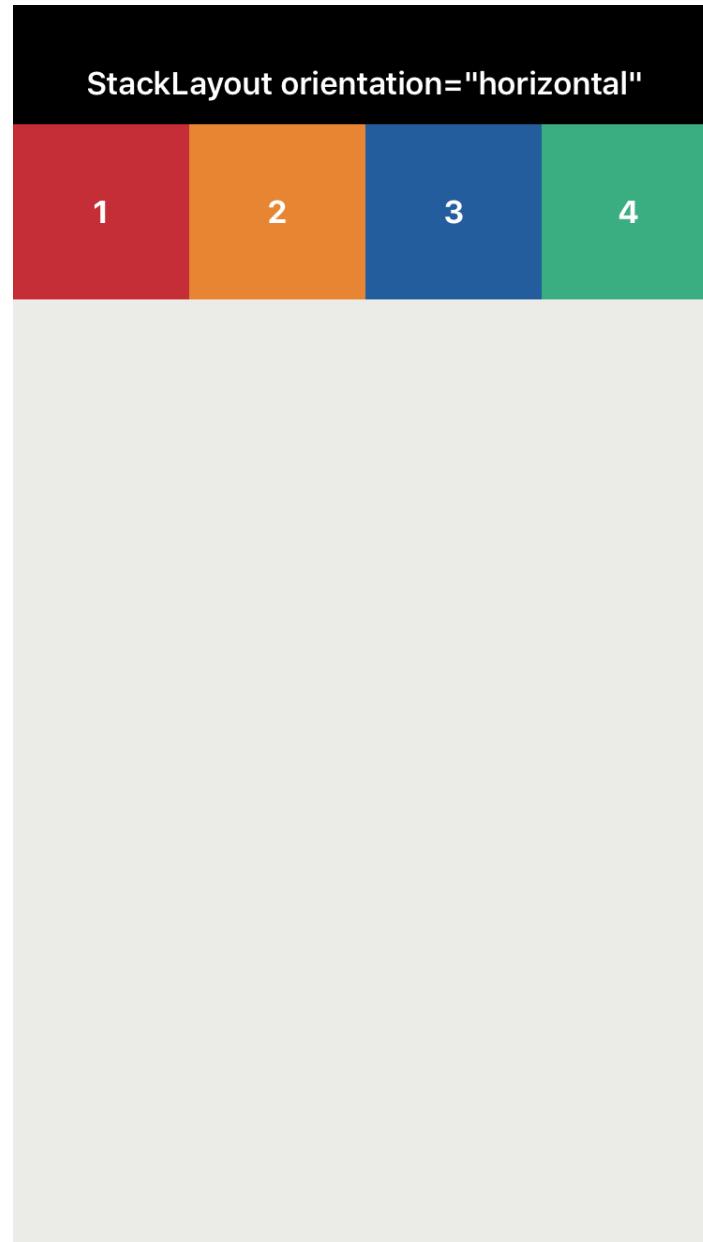
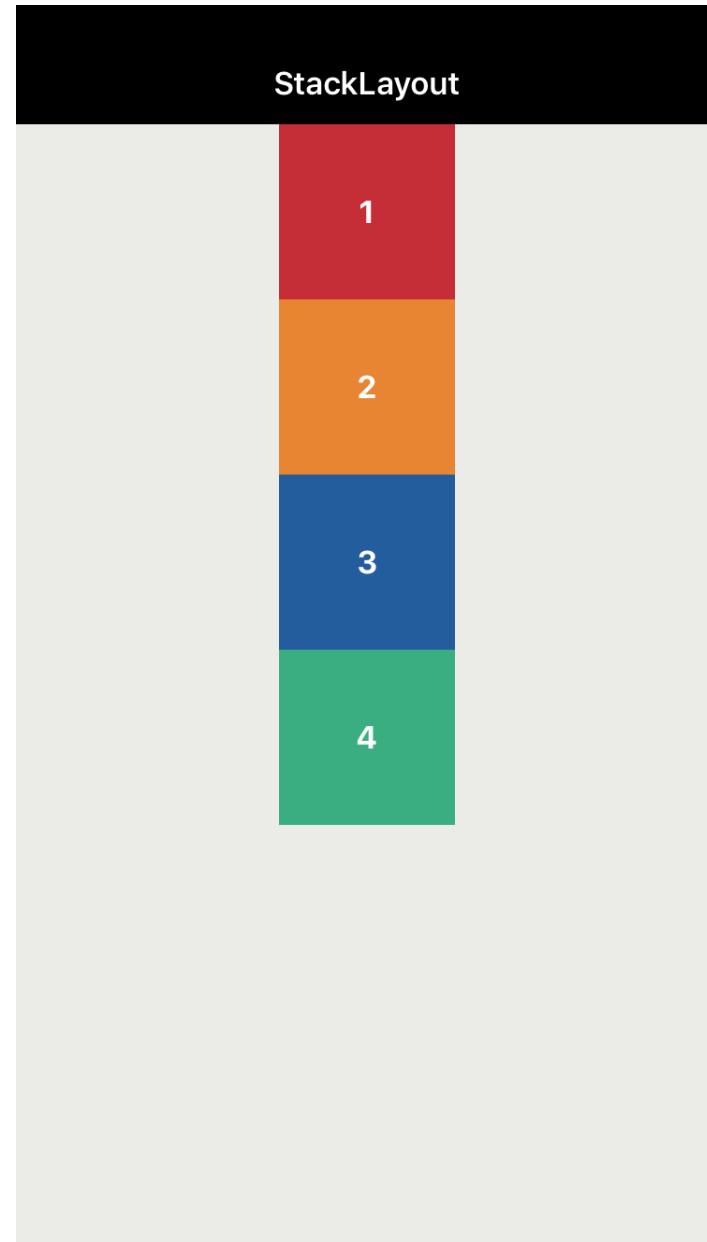
```
<TextField visibility="" />
```

```
<TextField width="" />
```

```
<TextField maxLength="" />
```

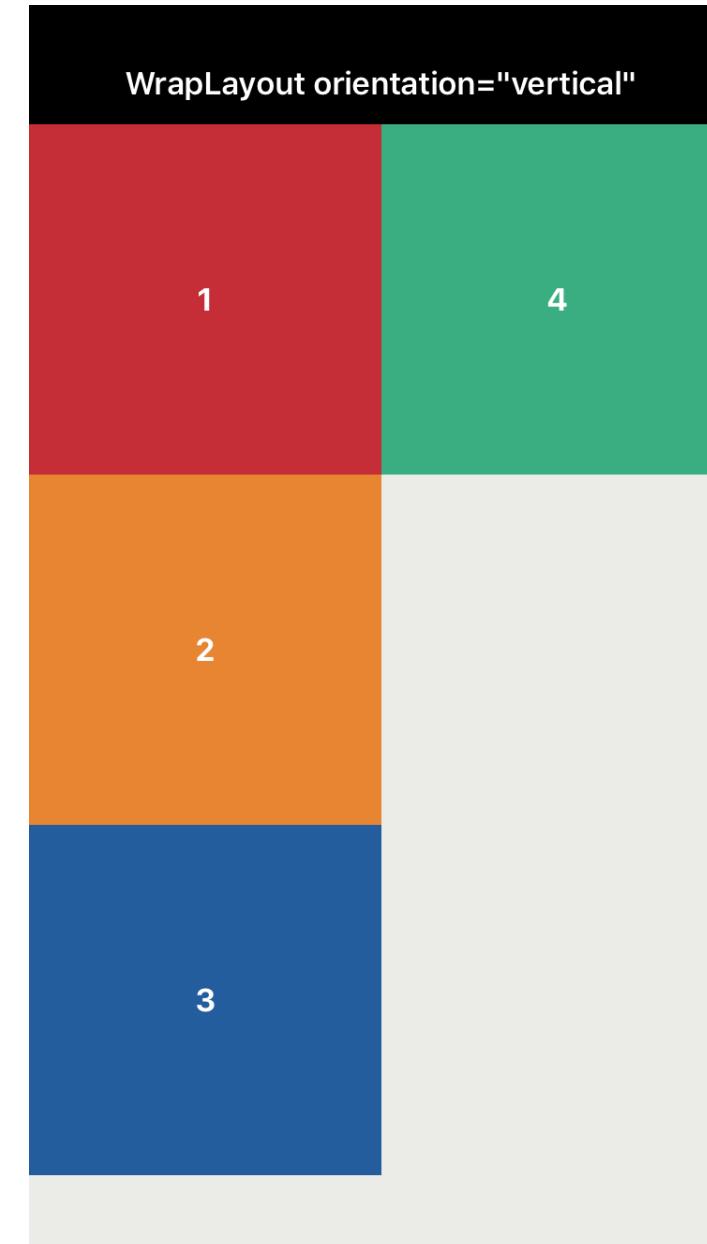
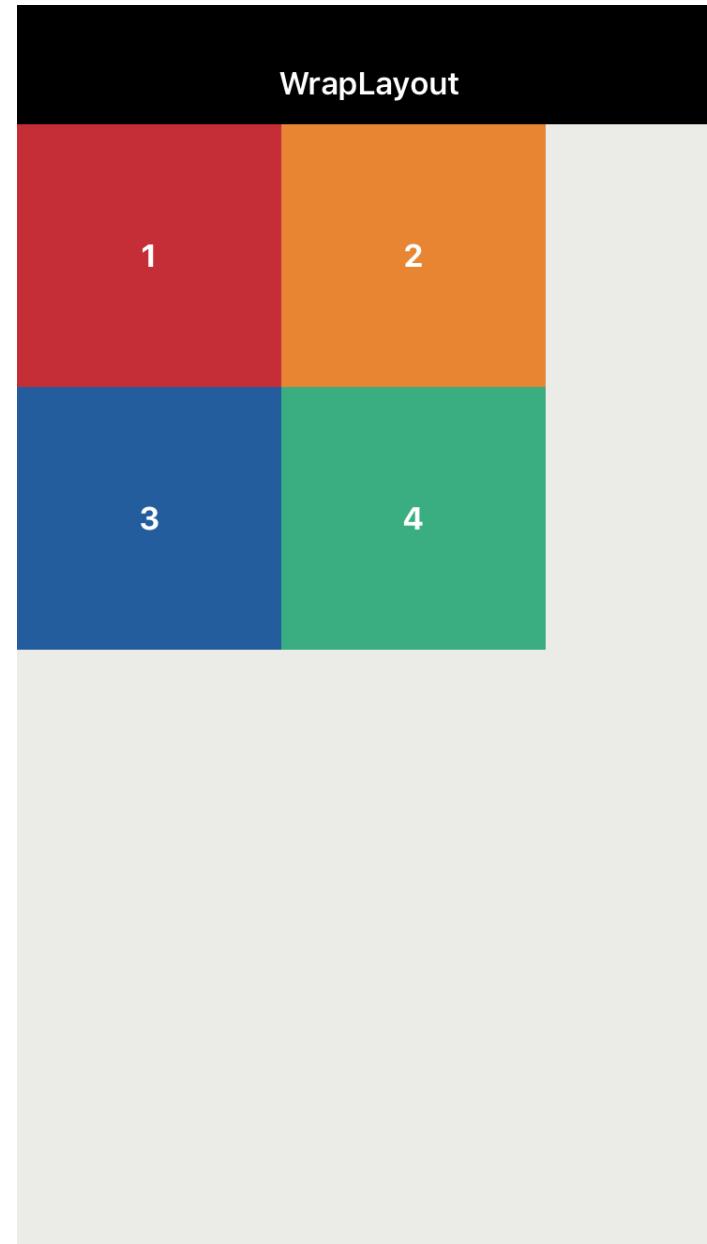
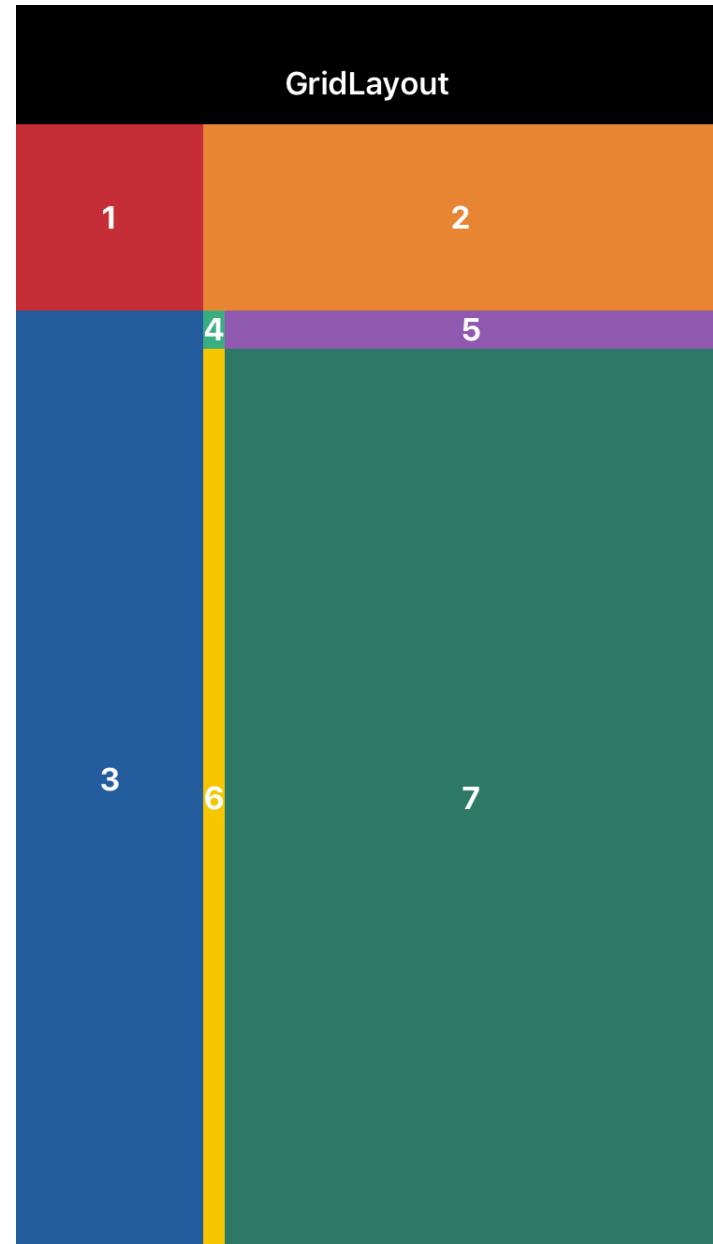
AYOUTS

<https://docs.nativescript.org/ui/layouts/layout-containers>



AYOUTS

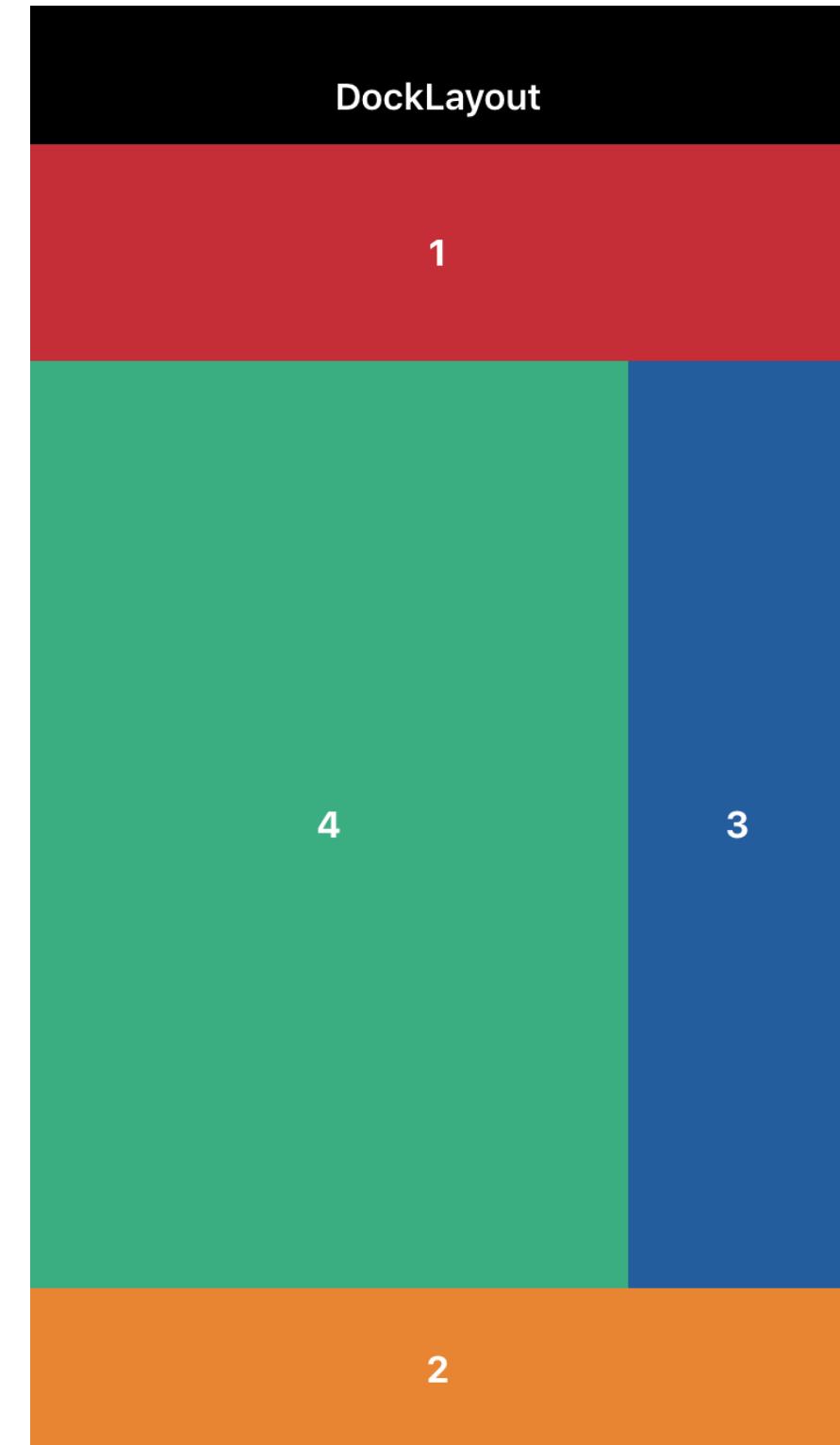
<https://docs.nativescript.org/ui/layouts/layout-containers>



DOCKLAYOUT

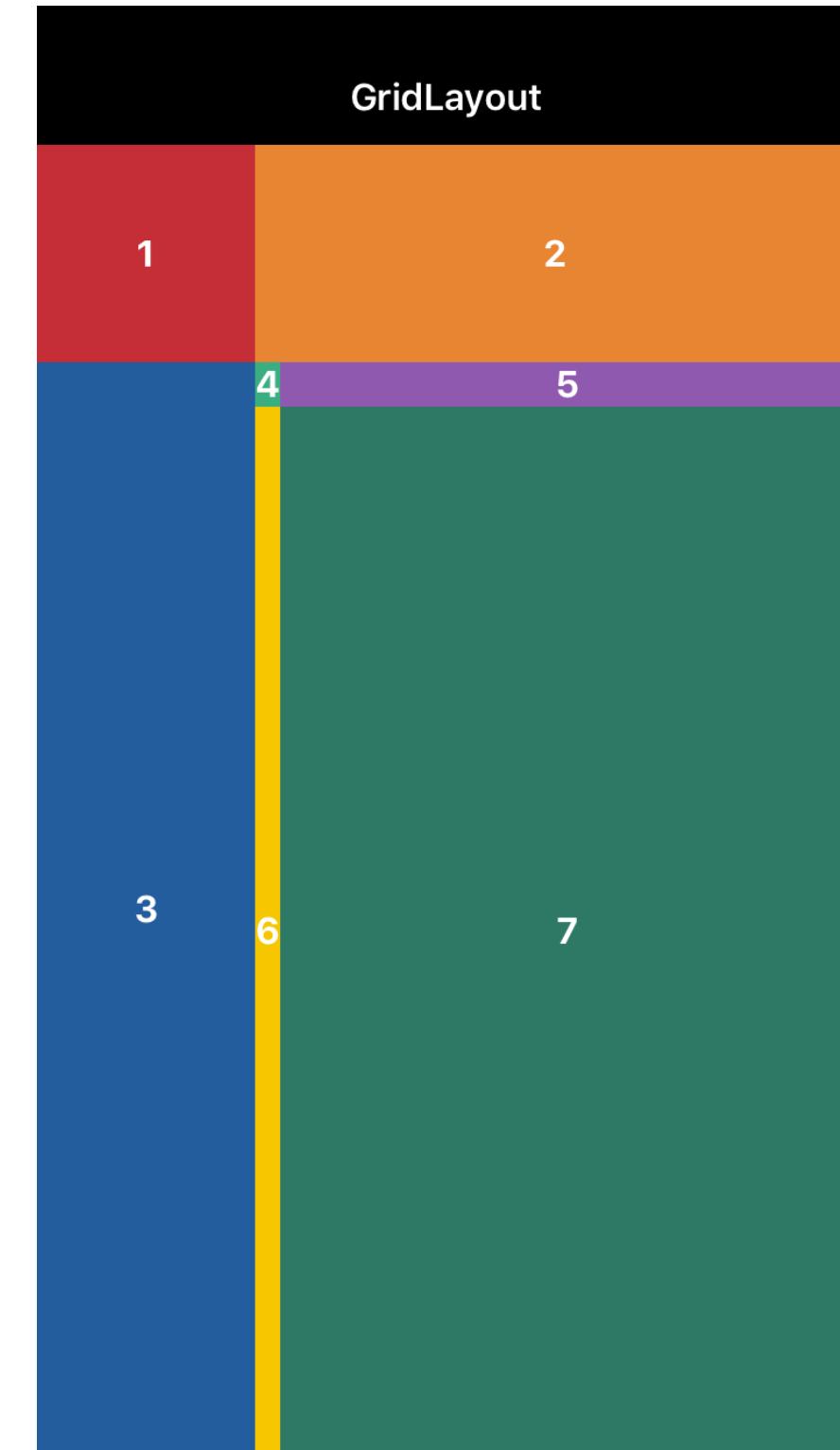
```
<DockLayout height="100%" stretchLastChild="true">
    <label text="1" dock="top" />
    <label text="2" dock="bottom" />
    <label text="3" dock="right" />
    <label text="4" dock="left" />
</DockLayout>
```

```
<!--
  1 + 2 have fixed height
  3 has fixed width
  4 will get all remaining space
-->
```



GRIDLAYOUT

```
<GridLayout  
    rows="100, auto, *"  
    columns="100, auto, *"  
    <Label text="1" row="0" col="0">  
    <Label text="2" row="0" col="1"  
        colSpan="2"/>  
    <Label text="3" row="1" col="0"  
        rowSpan="2"/>  
    <Label text="4" row="1" col="1"/>  
    <Label text="5" row="1" col="2"/>  
    <Label text="6" row="2" col="1"/>  
    <Label text="7" row="2" col="2"/>  
</GridLayout>
```





A woman with blonde hair tied back in a ponytail is shown from the chest up. She is wearing a red top and is looking down at a laptop screen. The background is a blurred office environment.

CASCADING STYLE SHEETS

CASCADING STYLE SHEETS

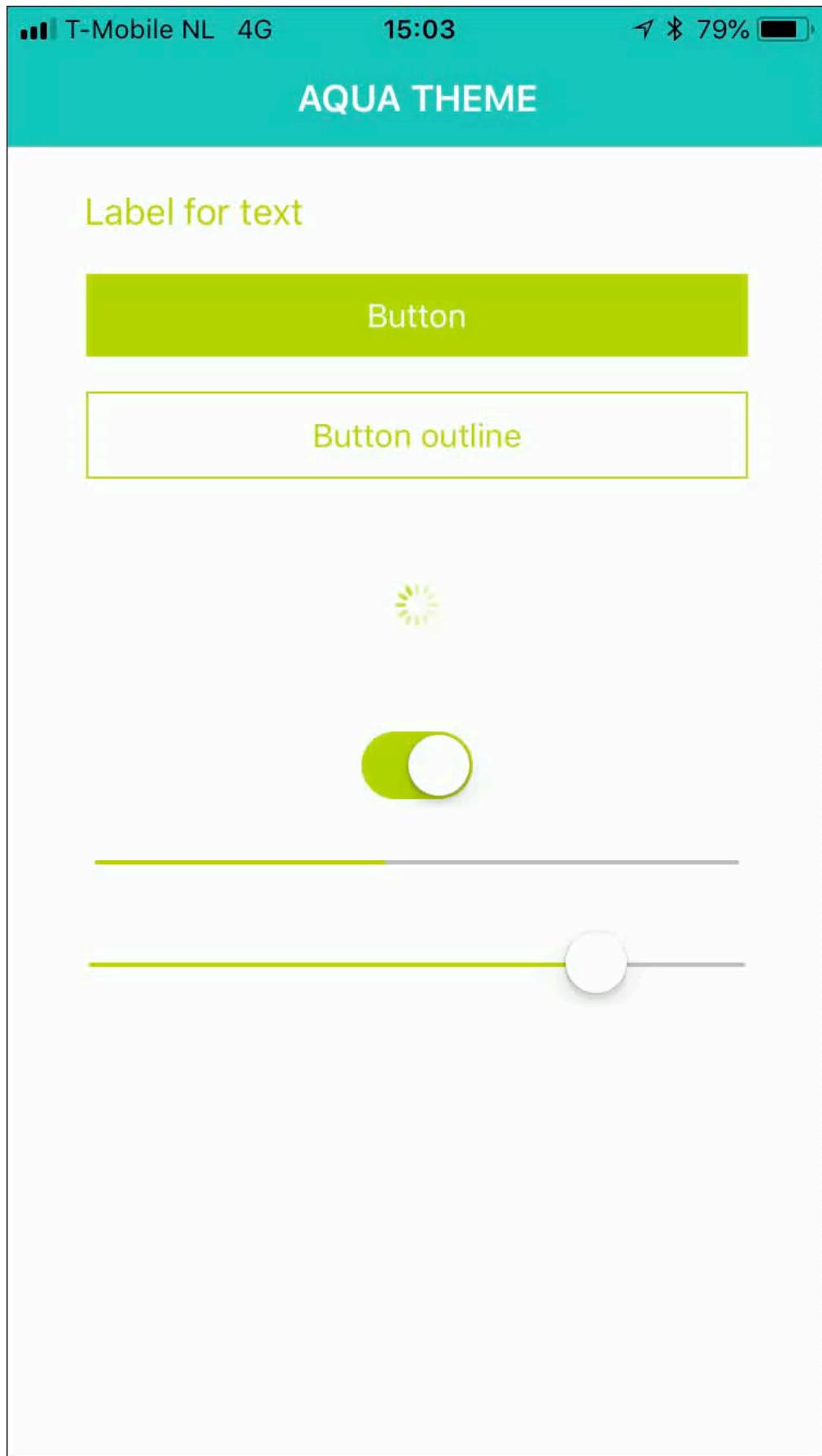
<https://docs.nativescript.org/ui/styling>

- a large subset of CSS properties is supported
- device-independent pixels
- application-wide, page-specific or inline
- platform-specific possible
- animations
- SASS

{N} CORE THEMES

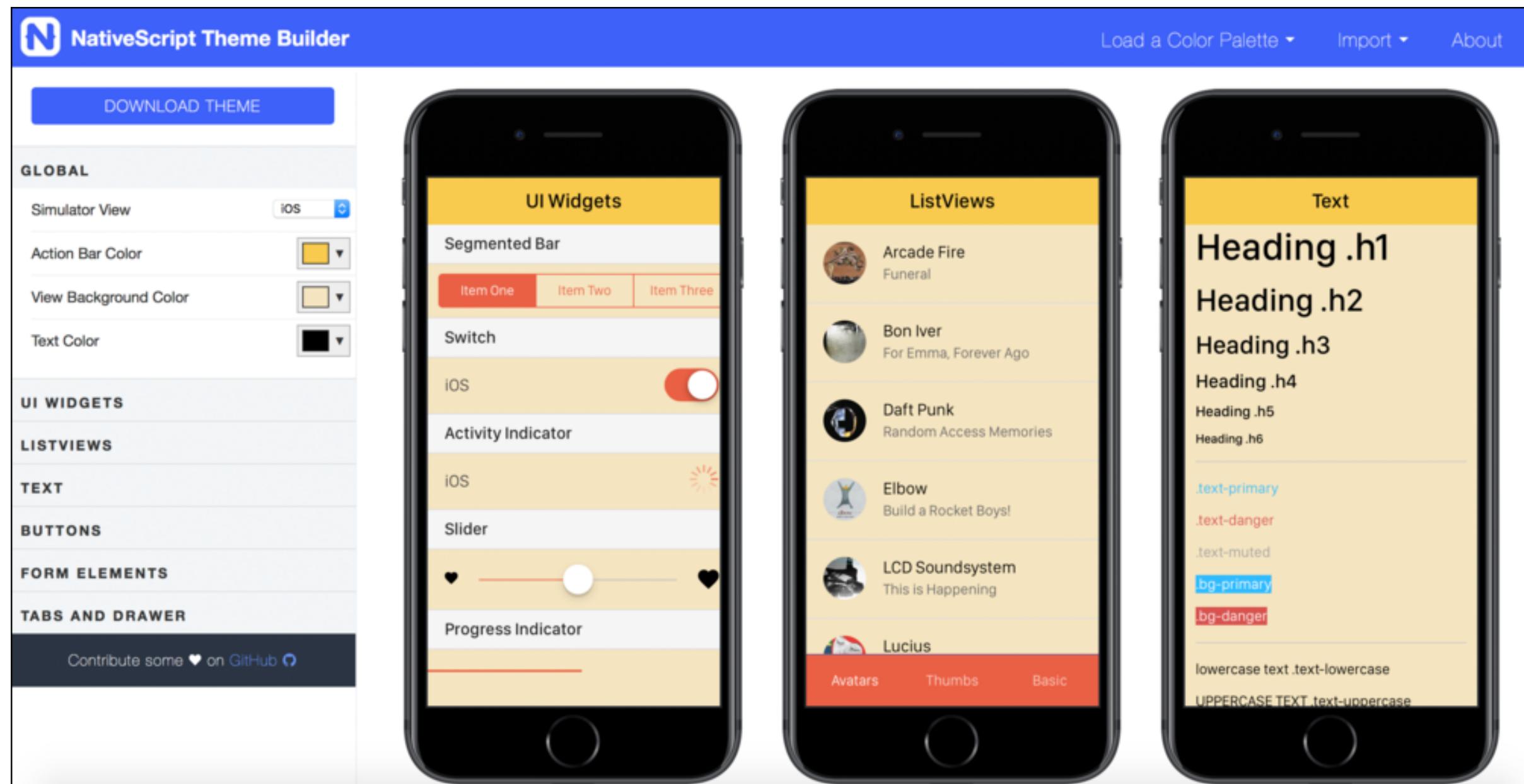
<https://docs.nativescript.org/ui/theme>

- **ready to use color schemes**
- **tailored for iOS and Android**



NATIVESCRIPT THEME BUILDER

<https://www.nativescriptthemebuilder.com>



TABVIEW

```
<TabView height="100%">
  <StackLayout *tabItem="{title: 'Rocket Raccoon'}" class="full rocket" />
  <StackLayout *tabItem="{title: 'Harley Quinn'}" class="full harley" />
  <StackLayout *tabItem="{title: 'Hulk'}" class="full hulk" />
</TabView>
```

TABVIEW

```
.full {  
    background-size: cover;  
    background-position: center;  
    background-repeat: no-repeat;  
}  
  
.rocket {  
    background-image: url("~/images/rocket-raccoon.jpg");  
}  
  
.harley {  
    background-image: url("~/images/harley-quinn.jpg");  
}  
  
.hulk {  
    background-image: url("~/images/hulk.jpg");  
}
```

TabView

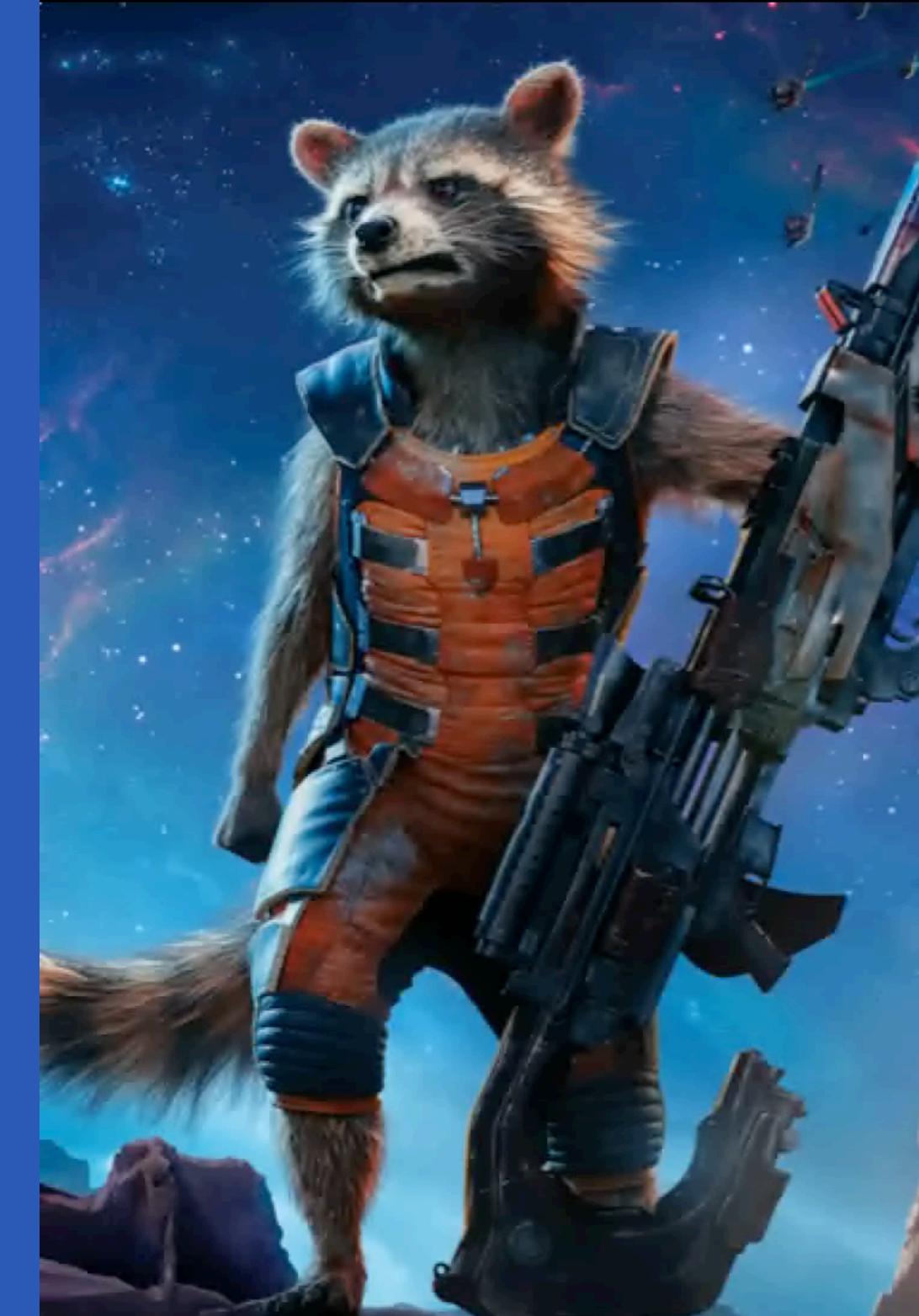
ROCKET
RACCOON

HARLEY QUINN

HULK



TabView



Rocket Raccoon

Harley Quinn

Hulk



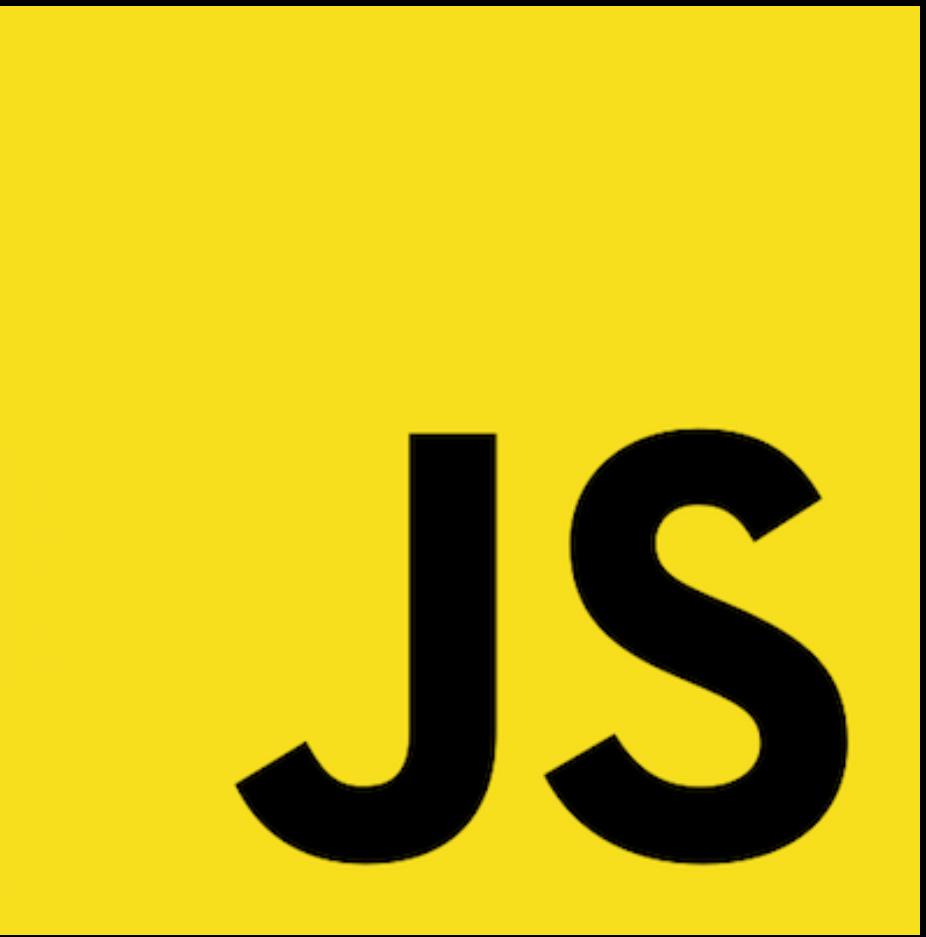
4 FANTASTIC CHOICES





6 BIG HIT PROPS





JAVASCRIPT



TYPESCRIPT



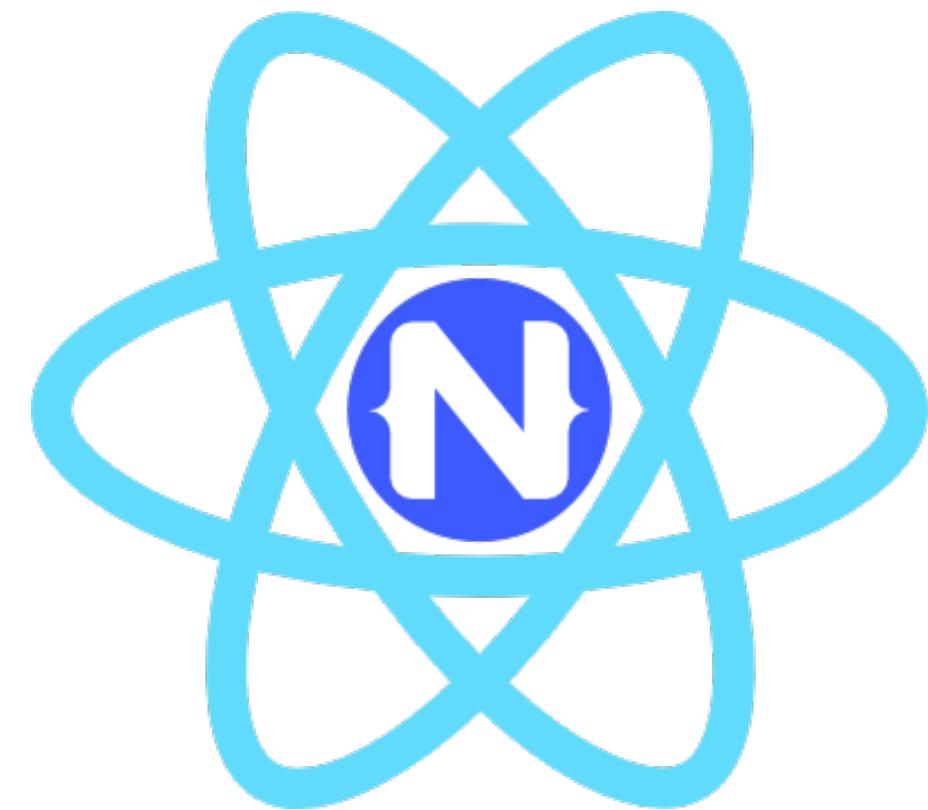
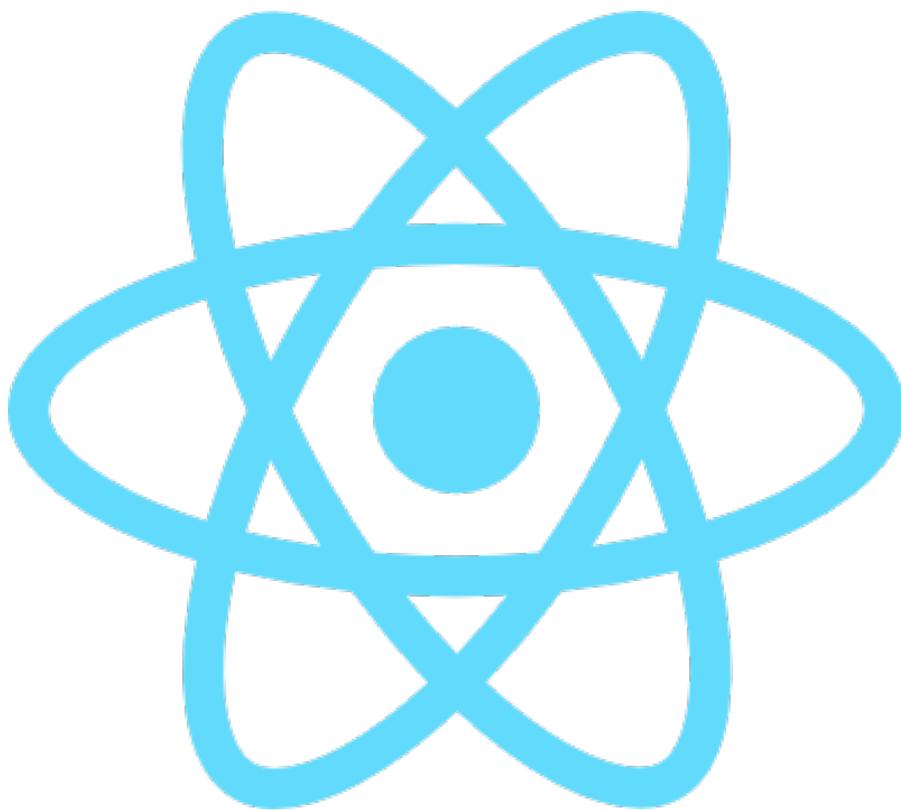
ANGULAR



VUE.js
IGOR RANDJELOVIC (NATIVESCRIPT-VUE)



SVELTE
DAVID PERSHOUSE (SVELTE NATIVE)



REACT

JAMIE BIRCH (REACT NATIVE SCRIPt)



Damaged



**NATIVE-CODE
OBJECTIVE-C**

{N} ANGULAR DIRECTORY STRUCTURE

app/components/slider/slider.component.html

slider.component.css

slider.component.ts

slider-routing.module.ts

slider-module.ts

SLIDER.COMPONENT.HTML

```
<ActionBar>
  <NavigationButton visibility="collapsed"></NavigationButton>
</ActionBar>
<StackLayout>
  <Slider></Slider>
</StackLayout>
```

SLIDER.COMPONENT.CSS

```
ActionBar {  
    background-color: #ffffff;  
}
```

```
StackLayout {  
    padding: 50;  
}
```

SLIDER.COMPONENT.TS

```
import { Component } from "@angular/core";  
  
@Component({  
  selector: "app-slider",  
  moduleId: module.id,  
  templateUrl: "slider.component.html",  
  styleUrls: ["slider.component.css"],  
})  
  
export class SliderComponent {  
  constructor() {}  
}
```

SLIDER-ROUTING.MODULE.TS

```
import { NgModule } from "@angular/core";
import { Routes } from "@angular/router";
import { NativeScriptRouterModule } from "@nativescript/angular";

import { SliderComponent } from "./slider.component";

const routes: Routes = [{ path: "", component: SliderComponent }];

@NgModule({
  imports: [NativeScriptRouterModule.forChild(routes)],
  exports: [NativeScriptRouterModule],
})

export class SliderRoutingModule {}
```

SLIDER MODULE.TS

```
import { NgModule, NO_ERRORS_SCHEMA } from "@angular/core";
import { NativeScriptCommonModule } from "@nativescript/angular/common";

import { SliderRoutingModule } from "./slider-routing.module";
import { SliderComponent } from "./slider.component";

@NgModule({
    imports: [NativeScriptCommonModule, SliderRoutingModule],
    declarations: [SliderComponent],
    schemas: [NO_ERRORS_SCHEMA],
})

export class SliderModule {}
```



SLIDER.COMPONENT.HTML

```
<ActionBar>
  <NavigationButton visibility="collapsed"></NavigationButton>
</ActionBar>
<StackLayout>
  <Slider slider-icon></Slider>
</StackLayout>
```

- **attribute directive**
- **changes the appearance or behavior of an element**

SLIDER.DIRECTIVE.TS

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "@nativescript/core";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      const uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

SLIDER.DIRECTIVE.TS

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "@nativescript/core";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      const uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

SLIDER.DIRECTIVE.TS

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "@nativescript/core";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      const uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

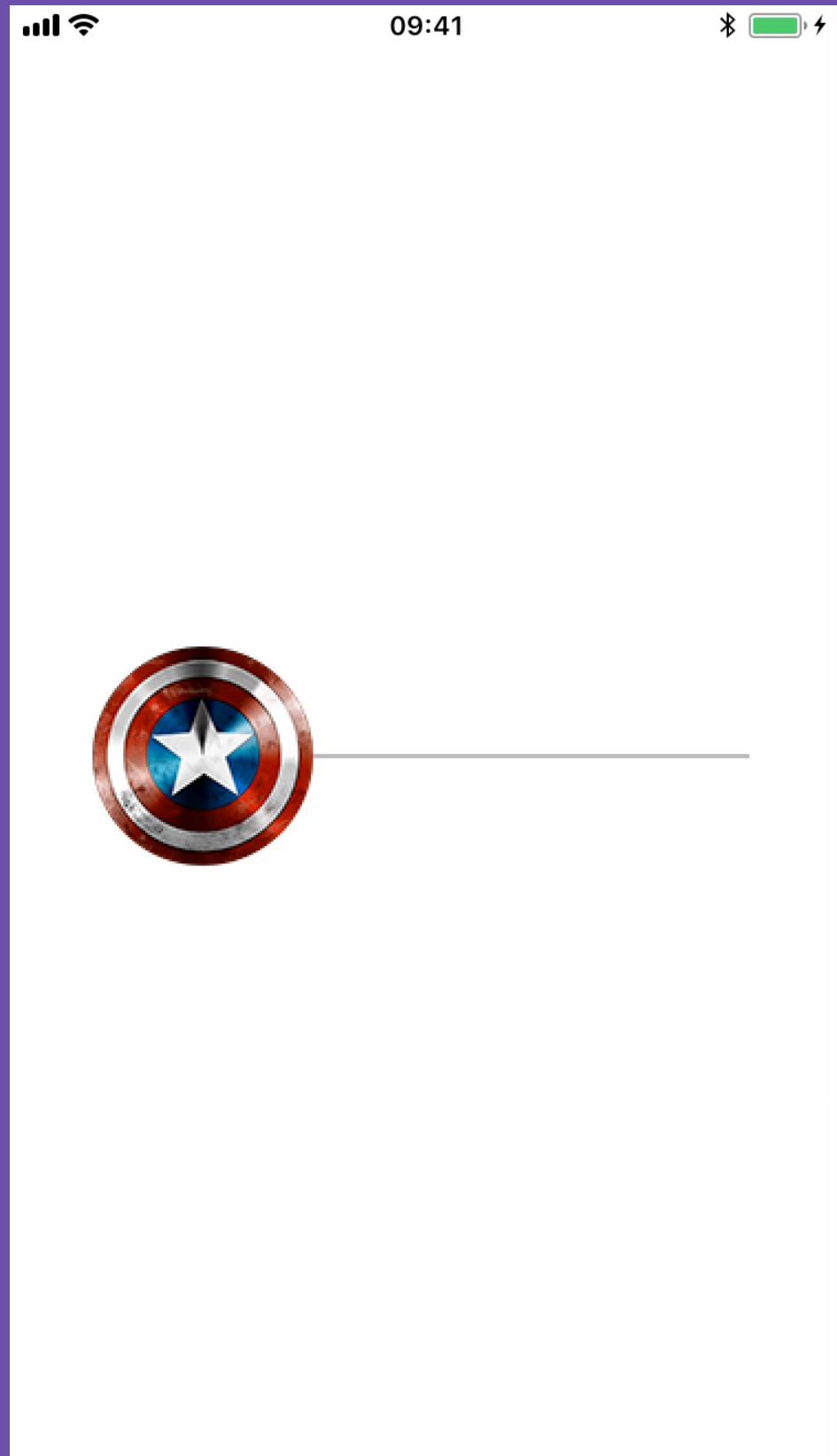
SLIDER MODULE.TS

```
import { NgModule, NO_ERRORS_SCHEMA } from "@angular/core";
import { NativeScriptCommonModule } from "@nativescript/angular/common";

import { SliderIconDirective } from "./slider.directive";

import { SliderRoutingModule } from "./slider-routing.module";
import { SliderComponent } from "./slider.component";

@NgModule({
  imports: [NativeScriptCommonModule, SliderRoutingModule],
  declarations: [SliderComponent, SliderIconDirective],
  schemas: [NO_ERRORS_SCHEMA]
})
export class SliderModule {}
```



SLIDER.COMPONENT.HTML

```
<ActionBar>
  <NavigationButton visibility="collapsed"></NavigationButton>
</ActionBar>
<StackLayout>
  <AbsoluteLayout>
    <StackLayout #background class="captain" top="0" left="0"></StackLayout>
    <FlexboxLayout class="flexcontainer" top="0" left="0">
      <Slider slider-icon (valueChange)="onSliderChange($event)"></Slider>
    </FlexboxLayout>
  </AbsoluteLayout>
</StackLayout>
```

SLIDER.COMPONENT.CSS

```
.captain {  
    background-image: url("~/assets/images/captain-america.jpg");  
    background-repeat: no-repeat;  
    background-position: center;  
    background-size: cover;  
    height: 100%;  
    width: 100%;  
    opacity: 0;  
}
```

SLIDER.COMPONENT.CSS

```
.flexcontainer {  
    justify-content: center;  
    align-items: center;  
    height: 100%;  
    width: 100%;  
}  
  
Slider {  
    width: 80%;  
    background-color: #bc1a0f;  
}
```

SLIDER.COMPONENT.TS

```
import { Component, OnInit, ViewChild, ElementRef } from "@angular/core";
import { Page } from "@nativescript/core/ui/page";
import { Slider } from "@nativescript/core/ui/slider";
import { StackLayout } from "@nativescript/core/ui/layouts/stack-layout";
import { TNSPlayer } from "nativescript-audio";

@Component({
    selector: "app-slider",
    moduleId: module.id,
    templateUrl: "slider.component.html",
    styleUrls: ["slider.component.css"]
})
```

SLIDER.COMPONENT.TS

```
export class SliderComponent implements OnInit {
  @ViewChild("background") background: ElementRef;
  private viewStack: StackLayout;
  private player: TNSPlayer;

  constructor(private page: Page) {}

  ngOnInit() {
    this.page.actionBarHidden = true;
    this.viewStack = this.background.nativeElement;
    this.player = new TNSPlayer();
    this.player.initFromFile({
      audioFile: "~/assets/audio/captain.mp3",
      loop: false
    });
  }
}
```

SLIDER.COMPONENT.TS

```
onSliderValueChange(args) {
  const slider = <Slider>args.object;
  // opacity and volume range is 0 - 1
  const sliderValue = slider.value / 100;
  this.viewStack.opacity = sliderValue;
  if (Math.round(slider.value) > 0) {
    this.player.play();
    this.player.volume = sliderValue;
  } else {
    this.player.seekTo(0);
    this.player.pause();
  }
}
```





**PACKAGES
&
LIBRARIES**

NODE PACKAGE MANAGER

- commonly known as npm
- ready to use JavaScript modules
- about 650.000 packages of free, reusable code

ANDROID ARSENAL

- **libraries for Android (Java / Kotlin)**

COCOAPODS

- **libraries for iOS (Objective-C / Swift)**

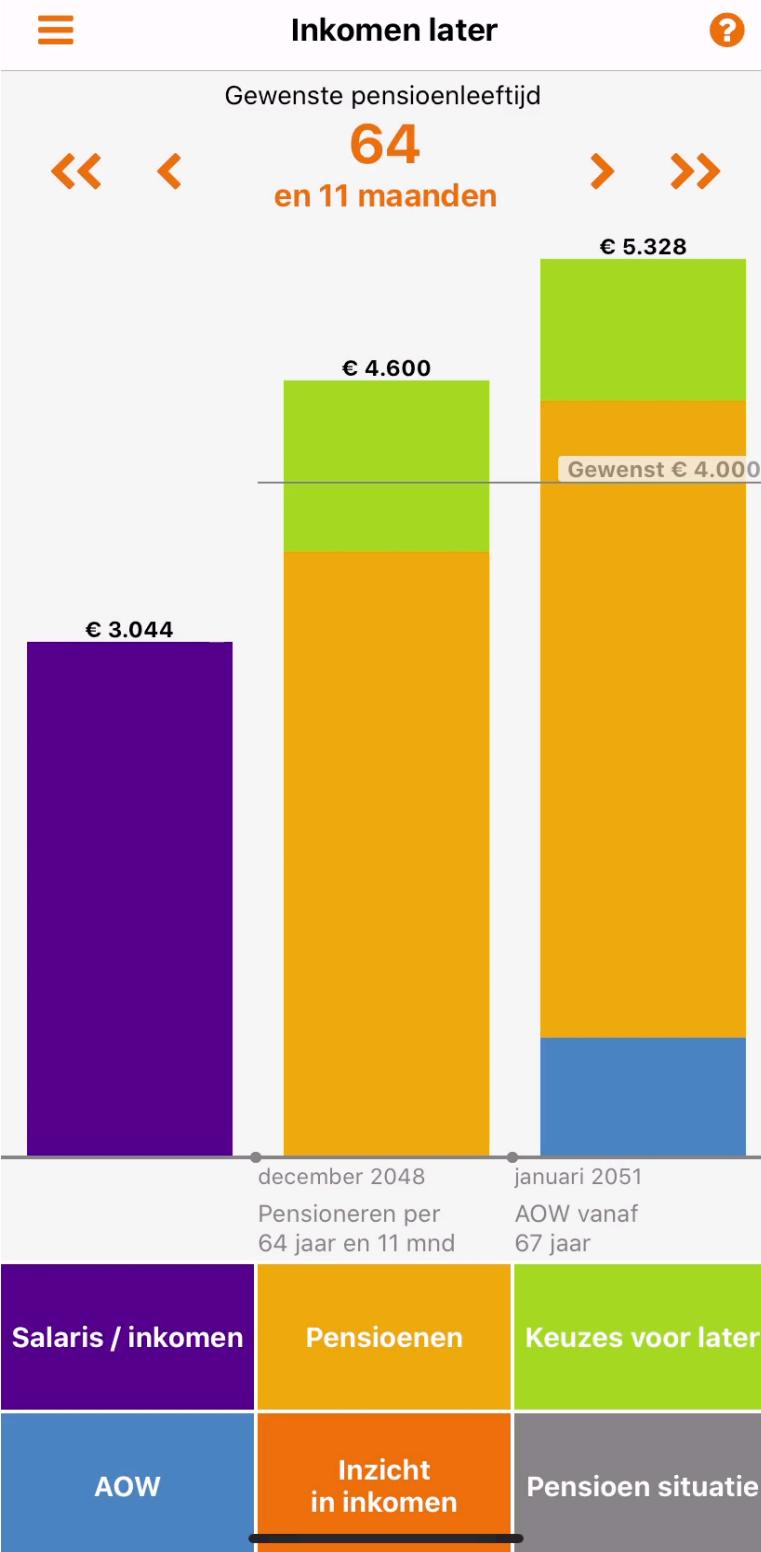




NOX-TRANSLATE

MULTILINGUAL: NGX-TRANSLATE

- **internationalization library for Angular**
- **define translations in different languages**
- **switch between them easily**
- **no hardcoded text/labels, all in one place**
- **start directly, even with one language**

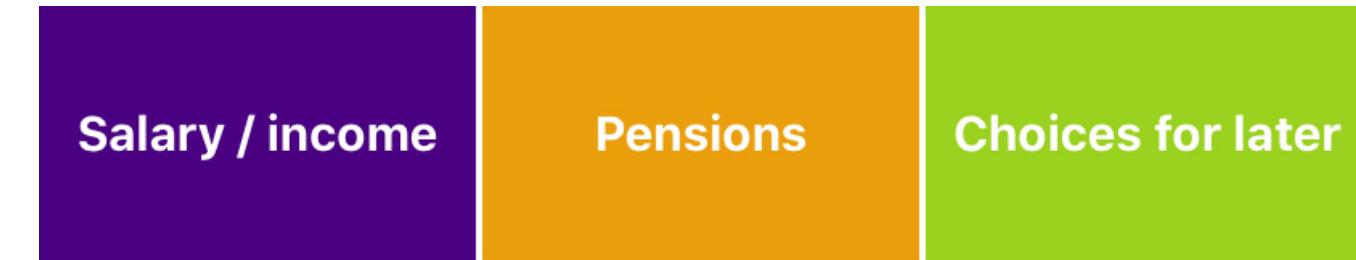


MULTILINGUAL: NGX-TRANSLATE



```
{  
  "CHART": {  
    "BUTTONS": {  
      "SALARY": "Salaris / inkomen",  
      "PENSIONS": "Pensioenen",  
      "CHOICES": "Keuzes voor later"  
    }  
  }  
}
```

MULTILINGUAL: NGX-TRANSLATE



```
{  
  "CHART": {  
    "BUTTONS": {  
      "SALARY": "Salary / income",  
      "PENSIONS": "Pensions",  
      "CHOICES": "Choices for later"  
    }  
  }  
}
```

MULTILINGUAL: NGX-TRANSLATE



```
<button [text]="'CHART.BUTTONS.SALARY' | translate"></button>
<button [text]="'CHART.BUTTONS.PENSIONS' | translate"></button>
<button [text]="'CHART.BUTTONS.CHOICES' | translate"></button>
```

[] = one way data binding in Angular

| = display-value transformations



NATIVESCRIPT PIUGINS



WHAT ARE {N} PLUGINS?

When the NativeScript core modules do not provide the native device or platform capability that you need, you can use plugins.

- usually for both iOS and Android
- JavaScript interface to native platform code



VERSIO
N
PUGI

VERSION PLUGIN

- » **package.json**
- » **version.d.ts**
- » **version.android.ts**
- » **version.ios.ts**

PACKAGE.JSON

```
{  
  "name": "nativescript-version",  
  "version": "1.0.0",  
  "main": "version",  
  "typings": "version.d.ts",  
  "nativescript": {  
    "platforms": {  
      "android": "7.0.0",  
      "ios": "7.0.0"  
    }  
  }  
  ...  
}
```

VERSION.DTS

```
export declare class Version {  
    constructor();  
  
    getVersion(): string;  
}
```

VERSION.ANDROID.TS

```
import { Application } from "@nativescript/core";

declare const android: any;

export class Version {
    constructor() {}

    getVersion(): string {
        const PackageManager = android.content.pm.PackageManager;
        const pkg = Application.android.context
            .getPackageManager()
            .getPackageInfo(
                Application.android.context.getPackageName(),
                PackageManager.GET_META_DATA
            );
        return pkg.versionName;
    }
}
```

VERSION.ANDROID.TS

```
import { Application } from "@nativescript/core";

declare const android: any;

export class Version {
    constructor() {}

    getVersion(): string {
        const PackageManager = android.content.pm.PackageManager;
        const pkg = Application.android.context
            .getPackageManager()
            .getPackageInfo(
                Application.android.context.getPackageName(),
                PackageManager.GET_META_DATA
            );
        return pkg.versionCode;
    }
}
```

ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1.0 demo">
    ...
</manifest>
```

VERSION.IOS.TS

```
declare const NSBundle: any;

export class Version {
    constructor() {}

    getVersion(): string {
        const version = NSBundle mainBundle.objectForInfoDictionaryKey(
            "CFBundleShortVersionString"
        );
        return version;
    }
}
```

VERSION.IOS.TS

```
declare const NSBundle: any;

export class Version {
    constructor() {}

    getVersion(): string {
        const version = NSBundle mainBundle.objectForInfoDictionaryKey(
            "CFBundleShortVersionString"
        );
        return version;
    }
}
```

INFO.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>CFBundleShortVersionString</key>
    <string>1.0 demo</string>
    ...
  </dict>
</plist>
```

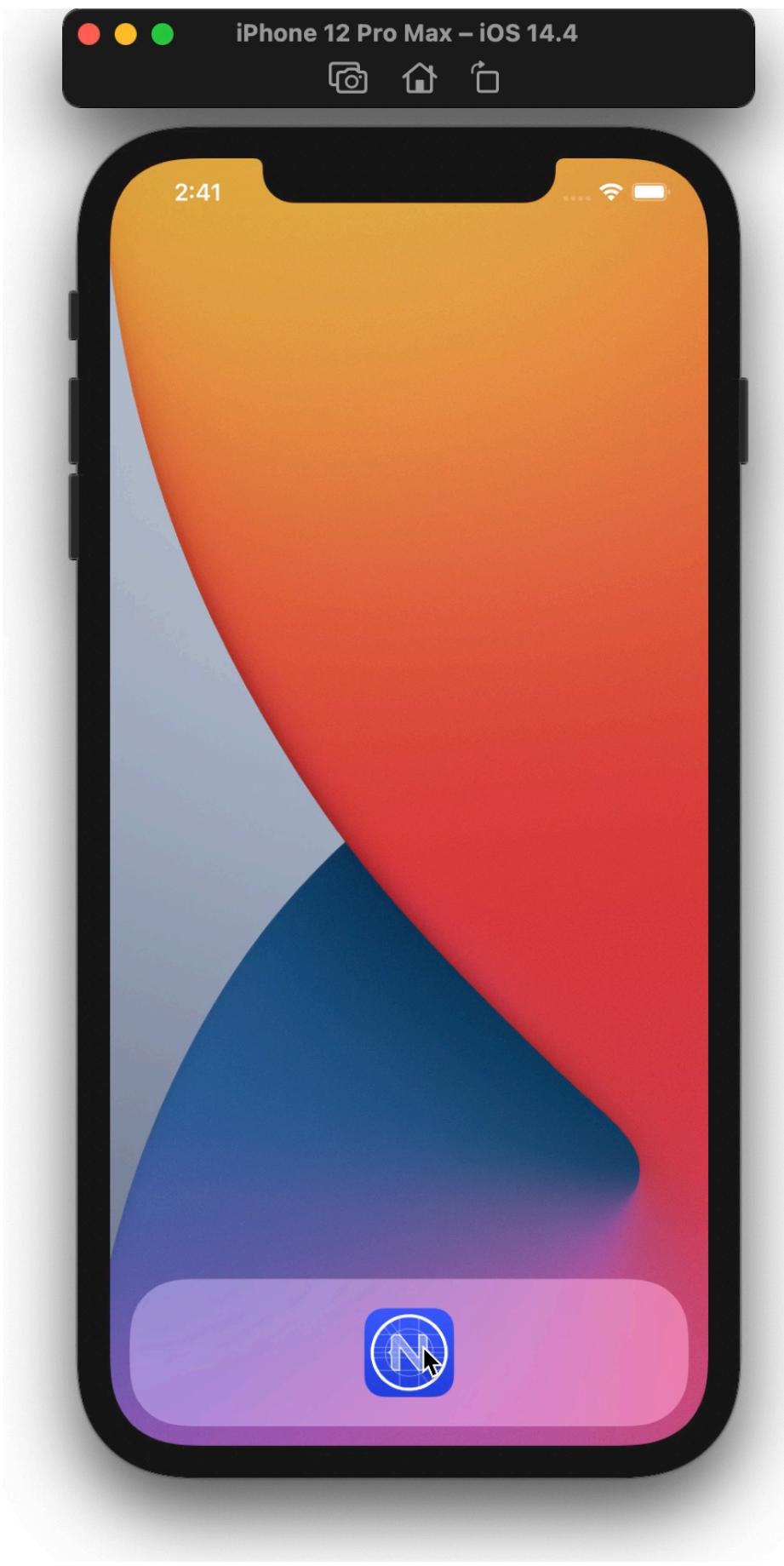
INSTALL

```
# install from NPM  
ns plugin add nativescript-version
```

```
# install from local  
ns plugin add ./nativescript-version
```

USAGE

```
import { Version } from "nativescript-version";  
  
const version = new Version().getVersion();
```



NATIVESCRIPT MARKETPLACE

 NativeScript | Marketplace

Authors | Privacy | NativeScript.org

Plugins Templates Samples

SUBMIT PLUGIN

Search for plugins

Recently Added

 **Chasexc NativeScript Fingerprint Auth**
by chasexc | Version 7.0.2
A fingerprint authentication plugin for use in NativeScript apps


 **Fitness**
by donswayo | Version 1.4.0
Google Fit and Apple HealthKit.


 **Set Version**
by farfromrefuge | Version 0.1.5

Recently Updated

 **OAuth 2**
by Alexander Ziskind | Version 2.3.0
OAuth 2 generic authorization plugin for NativeScript that doesn't install third party native libraries


 **Nota NativeScript Webview Ext**
by Morten Sjøgren | Version 6.2.1
Extended WebView for NativeScript which adds 'x-local' scheme for local-files, events between WebView and native-layer, javascript execution, injecting CSS and JS-files.


NATIVESCRIPT MARKETPLACE

The screenshot shows the NativeScript Marketplace interface. At the top, there's a blue header bar with the NativeScript logo and the text "NativeScript | Marketplace". On the right side of the header are links for "Authors", "Privacy", and "NativeScript.org". Below the header, there are three navigation tabs: "ALL", "GENERAL" (which is underlined, indicating it's the active category), and "HEALTHCARE".

The main content area displays four template cards:

- Blank**: A card for a blank template. It shows a screenshot of a white screen labeled "Home". Below the screenshot, the text reads "by NativeScript Team | Version 6.2.4" followed by four small icons for Angular (A), Vue (V), TypeScript (TS), and JavaScript (JS). A note below says "Blank template for NativeScript apps using Angular".
- Drawer Navigation**: A card for a side navigation template. It shows a screenshot of a mobile app with a blue header bar and a sidebar containing "User Name", "username@mail.com", "Home", "Browse", "Search", "Featured", and "Settings". Below the screenshot, the text reads "by NativeScript Team | Version 6.2.4" followed by four small icons for Angular (A), Vue (V), TypeScript (TS), and JavaScript (JS). A note below says "Side navigation template".
- Tab Navigation**: A card for a tabbed interface template. It shows a screenshot of a mobile app with a blue header bar and a tab bar at the bottom with three items: "Home", "Browse", and "Search". The main content area lists 11 items from "Item 1" to "Item 11". Below the screenshot, the text reads "by NativeScript Team | Version 6.2.4" followed by four small icons for Angular (A), Vue (V), TypeScript (TS), and JavaScript (JS). A note below says "Tabbed interface template".
- Browse**: A card for a browse template. It shows two screenshots of a mobile app with a blue header bar and a "Browse" button. The first screenshot is labeled "5:51 PM" and the second is labeled "5:51 PM". Below the screenshots, the text reads "by NativeScript Team | Version 6.2.4" followed by four small icons for Angular (A), Vue (V), TypeScript (TS), and JavaScript (JS).

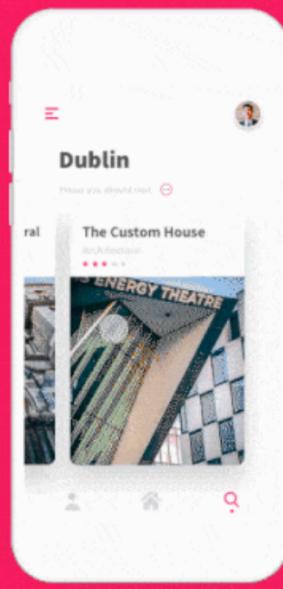
NATIVESCRIPT MARKETPLACE

 NativeScript | Marketplace

Authors | Privacy | NativeScript.org

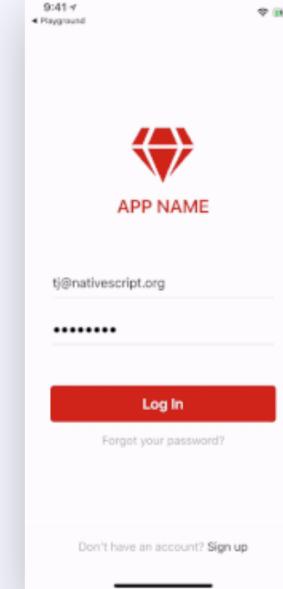
FRAMEWORKS
[All](#) Angular Vue Core React

CATEGORIES
[All](#) Layouts & Pages Forms & Data Interaction Animations Media



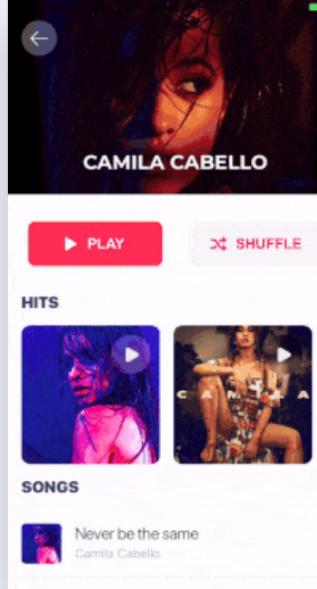
Animated Search
by Clément Roche 

Rich animations power this city guide sample app, including a card-based UI.



Building a Good-Looking Login Form
by Multiple authors   

A good looking login/registration experience is a must have for any app. Here is a simple yet elegant looking login form example.



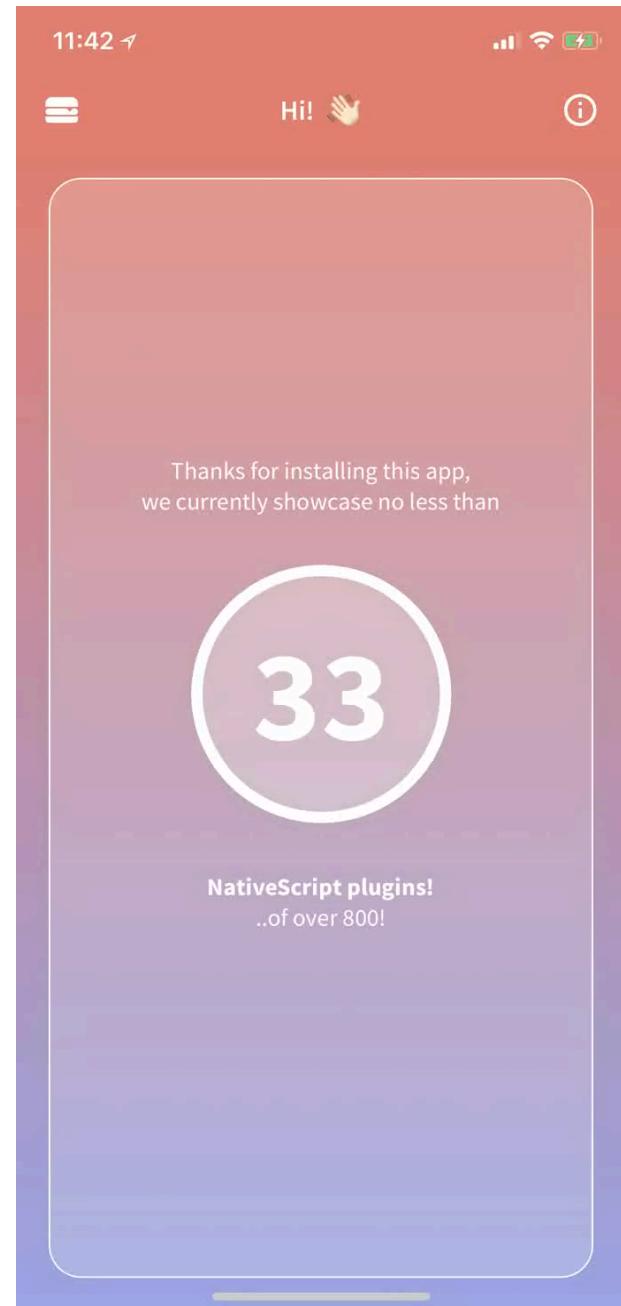
Music Streaming App
by Yassine Zanina  

This sample app contains an amazing design for to help bootstrap the next great music streaming app.

GNB PIUGGINS DEMO



GITHUB.COM/EDDYVERBRUGGEN/NATIVESCRIPT-PLUGINSHOWCASE





THANK YOU!

A close-up, high-contrast image of the symbiote known as Venom. His face is filled with sharp, fang-like teeth and a wide, toothy grin. His skin is a dark, iridescent greenish-blue with visible veins and textures. He has two long, thin, light-colored antennae-like structures extending from his forehead. The lighting is dramatic, with bright highlights on his teeth and the edges of his face against a dark, blurred background.

ROWDY.CODES/IPC