Inhoud

Inleiding

Je hebt dit boek aangeschaft omdat je een helder doel voor ogen hebt: je wilt weten hoe je met de programmeertaal Swift zelf iOS-apps kunt bouwen. Dit boek gaat je daarbij helpen. Sterker nog: als je dit boek hebt doorgewerkt kun je je eigen apps schrijven. Die belofte doe ik je.

Er zijn een paar dingen die je kunt doen om het meeste uit dit boek te halen.

- Allereerst: werk het boek door in de volgorde waarin het is geschreven. Weersta de verleiding om door te bladeren en stukken over te slaan. Alle zinnen in de volgende hoofdstukken zijn geschreven met maar één doel: je zo goed mogelijk te leren hoe Swift werkt en je zo snel mogelijk resultaat te bieden. De volgorde van onderwerpen is heel bewust gekozen: alles wordt op precies het juiste moment behandeld.
- Probeer alle voorbeelden zelf uit! Dit is geen leesboek, maar een doe-boek. Zorg dat je de laatste versie van Xcode hebt geïnstalleerd en ga aan het werk. Weersta de verleiding om dit boek alleen maar door te lezen; dat leidt niet tot het doel dat je wilt bereiken.
- Typ alle programmacode zelf over. Hoewel de bronbestanden kunnen worden gedownload (**www.iosacademie.nl/swift-boek-bestanden**) leer je het meeste als je zelf aan het werk bent. Ook de fouten die je maakt als je de programma's overtypt helpen je om een betere appontwikkelaar te worden.
- Neem de tijd! Verwacht niet dat je binnen een week een professionele app bouwt. Als je dit boek grondig doorwerkt heb je waarschijnlijk een maand nodig, maar dan ben je ook goed op weg.
- Als je vragen hebt waar je geen antwoord op weet, ga naar het forum van de iOS Academie (**www.iosacademie.nl**) en stel ze daar. Je krijgt antwoord van mij of van andere lezers. Het forum is geheel gratis.

Tot zover de 'huisregels'. Tijd om aan de slag te gaan!

Hoofdstuk 1

Kennismaken met Xcode

Om apps te bouwen voor de iPhone, iPad of Mac gebruik je Xcode. Xcode kun je gratis downloaden uit de Mac App Store. In dit boek gaan we ervan uit dat je de laatste versie van Xcode op je Mac hebt geïnstalleerd.

Xcode is een buitengewoon uitgebreide programmeeromgeving. Hoewel de ontwikkelaars er alles aan doen om Xcode zo gebruiksvriendelijk en intuïtief mogelijk te maken, zal je waarschijnlijk wat tijd nodig hebben om vertrouwd te raken met de diverse onderdelen van de gebruikersinterface. De leukste manier om dit te doen is door tegelijkertijd je eerste app te bouwen. En dat is precies wat we in dit hoofdstuk gaan doen.

Je leert in dit hoofdstuk:

Het werkvenster van Xcode begrijpen.

Hoe je een eerste app bouwt.

Wat views zijn.

Hoe je met layouts werkt.

Aan de slag met Xcode

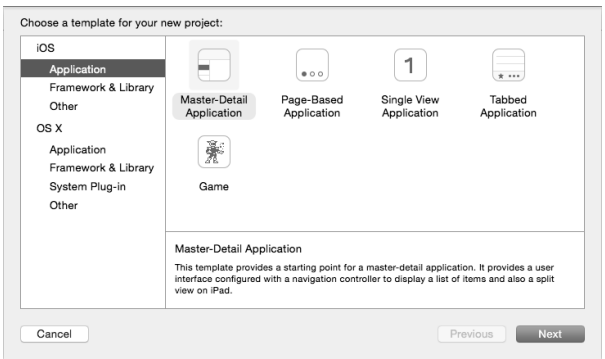
Het eerste wat je ziet als je Xcode start, is het welkomstvenster.

Afbeelding 1.1 *Het welkomstvenster.*

Als je eenmaal begonnen bent met het bouwen van apps, zal het gedeelte aan de rechterkant (waar nu **No Recent Projects** staat) een overzicht bevatten van de apps waaraan je recent hebt gewerkt.

Links in het venster zie je drie mogelijkheden. Omdat we in deze kennismaking onze eerste app gaan bouwen, kies je de tweede optie: **Create a new Xcode project**.

In het linker gedeelte van het venster dat nu verschijnt kun je kiezen uit twee categorieën: **iOS** en **OS X**. Je kunt met Xcode namelijk zowel iOS- als OS X-apps bouwen.

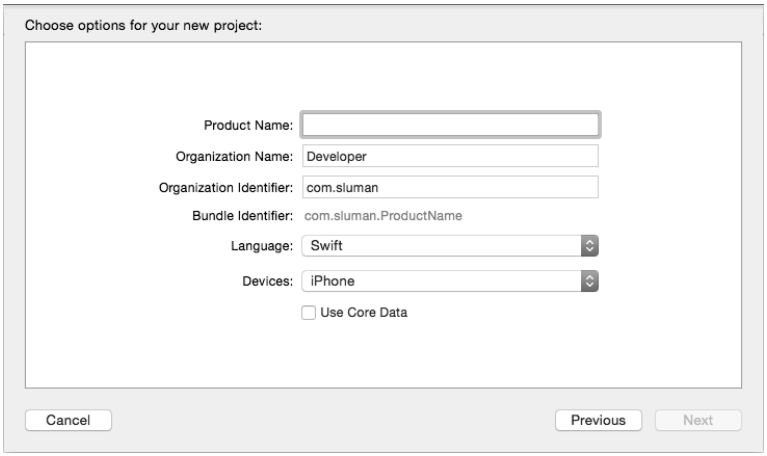


Afbeelding 1.2 *Kies een sjabloon.*

In dit hoofdstuk bouwen we een iOS-app; klik op **Application**.

Rechts in het venster zie je een aantal sjablonen (*templates*). Deze sjablonen bevatten een aantal basisbestanden voor de meest voorkomende typen apps. We komen hier later nog uitgebreid op terug. Kies nu **Single View Application**. Dit sjabloon bevat alle bestanden die nodig zijn voor de eenvoudige app die we in dit hoofdstuk gaan bouwen.

Tijd voor de volgende stap: klik op **Next**, waarna het venster uit afbeelding 1.3 verschijnt.

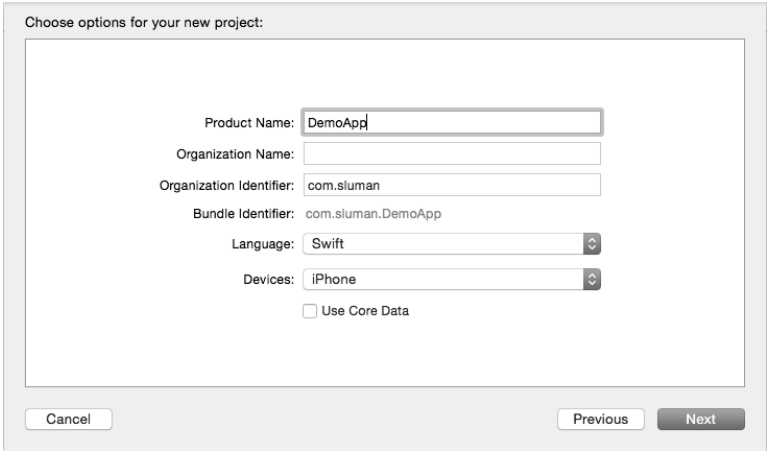


Afbeelding 1.3 *Kies de opties voor het project.*

In dit venster kiezen we een paar belangrijke opties voor onze eerste app:

- **Product Name** De naam van de app. Typ hier `DemoApp`.
- **Organization Name** Heb je een bedrijf? Typ hier dan de naam van je bedrijf. Je kunt dit veld ook leeglaten.
- **Organization Identifier** Heb je een eigen domeinnaam? Typ die hier dan achterstevoren in. Heb je geen domeinnaam? Typ dan iets willekeurigs, bijvoorbeeld nl.je-achternaam. Je moet hier iets invullen; de inhoud van de velden **Product Name** en **Organization Identifier** worden namelijk gebruikt om een unieke identificatiecode voor je app te maken: de zogeheten **Bundle Identifier**.
- **Language** Hier kun je kiezen uit Swift en Objective-C. In dit voorbeeld gebruiken we, uiteraard, Swift.

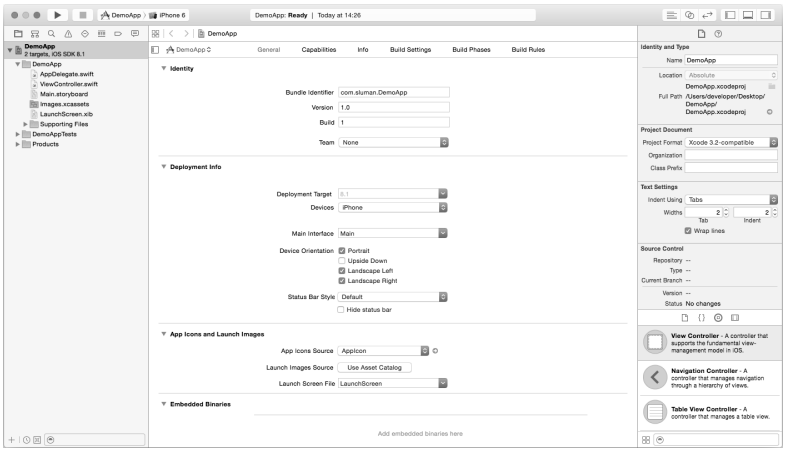
Laat de overige velden ongewijzigd; het resultaat zou moeten lijken op dat uit afbeelding 1.4.



Afbeelding 1.4 *De instellingen voor de app.*

Klik nu op **Next**. Xcode vraagt je nu waar de bestanden voor je eerste app moeten worden opgeslagen. Je kunt elke plek kiezen die je wilt, maar mijn advies is om in je thuismap een aparte map (bijvoorbeeld met de naam **Apps**) te maken waarin je al je projecten kunt bewaren.

Nadat je op **Create** hebt geklikt zal Xcode alle bij onze app behorende bestanden, op basis van het sjabloon dat we zojuist hebben gekozen, bewaren. Zodra dit is gebeurd, verschijnt het werkvenster van Xcode (zie afbeelding 1.5). Laat je niet afschrikken door de enorme hoeveelheid informatie in dit venster!



Afbeelding 1.5 *Het werkvenster van Xcode.*

Je hebt zojuist je eerste app gebouwd! Straks zie je hoe je je nieuwe app kunt starten in de **Simulator**, een bij Xcode geleverd programma waarmee je je eigen apps kunt testen, ook als je niet over een iPhone of iPad beschikt.

Het werkvenster van Xcode

Hoe gecompliceerd het werkvenster van Xcode ook lijkt, het wordt allemaal al een stuk eenvoudiger als je bedenkt dat dit werkvenster uit vijf onderdelen bestaat.

- Bovenaan zie je de werkbalk.
- In het middelste gedeelte zie je, van links naar rechts:
  - De Navigator
  - De Editor
  - De Utilities
- Onderin zie je, mits ingeschakeld, het Debug-gedeelte.

Werkbalk

De werkbalk (*toolbar*) beslaat de bovenste gedeelte van het werkvenster. Deze werkbalk bevat een aantal onderdelen. Links zie je knoppen om de app te starten en te stoppen.

Afbeelding 1.6 *Knoppen om de app te starten en te stoppen.*

Daarnaast zie je de naam van de *target*: datgene wat Xcode gaat bouwen; in ons geval is dat dus DemoApp. Daarnaast staat vermeld voor welk device onze app zal worden gebouwd: een iPhone 6.

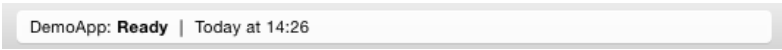
Afbeelding 1.7 *DemoApp voor iPhone 6.*

Developer-account

Hoewel iedereen met Xcode apps kan ontwikkelen, kun je die apps niet zomaar op een iPhone of iPad zetten. Daarvoor dien je je bij Apple te registreren als ontwikkelaar (Developer). Tegen betaling van 99 euro per jaar heb je dan de mogelijkheid om apps niet alleen op je eigen iPhone en/of iPad te installeren, maar ook om ze door anderen te laten testen en om ze, na goedkeuring van Apple, in de App Store te plaatsen.

Je hoeft overigens geen Apple Developer-account te hebben om met dit boek te kunnen werken. Xcode bevat een simulator waarmee je je apps op je Mac kunt testen, dus zonder dat je een echte iPhone of iPad gebruikt.

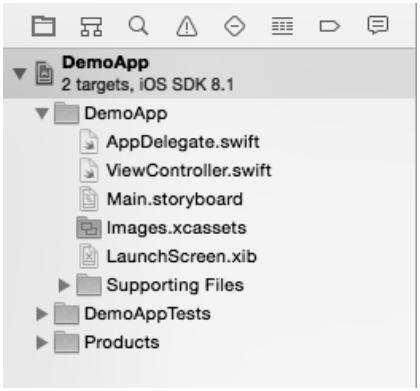
In het midden van de werkbalk staat een paneel met informatie over de status van de app (zie afbeelding 1.8). Aan de rechterkant van dit paneel is plaats voor statustekst, bijvoorbeeld als Xcode problemen heeft gevonden. Wanneer Xcode een fout heeft ontdekt, wordt dat hier aangegeven.



Afbeelding 1.8 *Het statuspaneel.*

Navigator

De Navigator (zie afbeelding 1.9), die links in het werkvenster wordt getoond, helpt je in eerste instantie om te navigeren door de diverse bestanden die bij je programma horen. Je zult hem echter ook gebruiken om op een snelle manier fouten op te sporen en om te zoeken naar bepaalde onderdelen van je app.



Afbeelding 1.9 *De Navigator.*

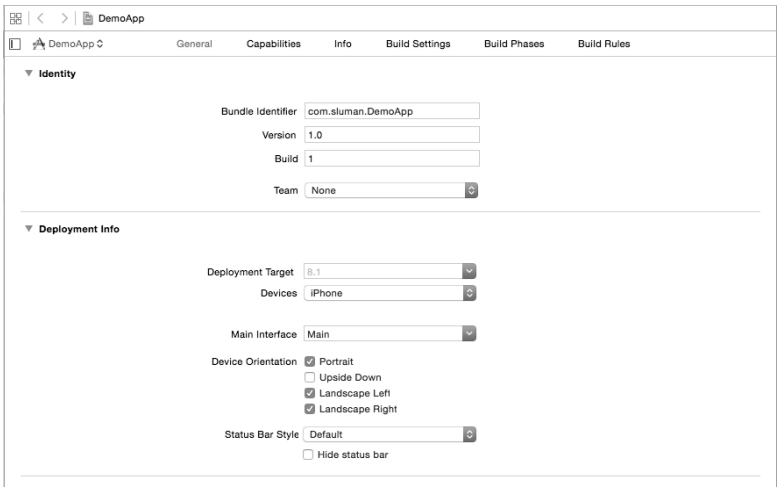
De Navigator beschikt over een eigen knoppenbalk, die je bovenaan in afbeelding 1.9 ziet. Met deze knoppenbalk kun je snel schakelen tussen de diverse navigatiemogelijkheden. Elke knop heeft z'n eigen toetscombinatie: van Cmd+1 tot en met Cmd+8. In dit hoofdstuk gebruiken we alleen de eerste knop: het overzicht van alle bestanden die bij onze app horen.

Als je meer schermruimte nodig hebt kun je de Navigator verbergen door in het groepje dat je rechts in de werkbalk ziet staan (zie afbeelding 1.10) op de meest linkse van de drie knoppen te klikken. Je kunt dit overigens ook doen door op de toetscombinatie Cmd+0 te drukken.

Afbeelding 1.10 *Besturingsknoppen voor de Navigator.*

Editor

De Editor gebruik je om onderdelen van je app aan te passen. Dat kunnen schermonderdelen (zoals knoppen, labels enzovoort) zijn, maar ook programmacode. In afbeelding 1.11 zie je een voorbeeld van de Editor; wat hier wordt getoond is afhankelijk van het onderdeel waarmee je bezig bent.



Afbeelding 1.11 *De Editor.*

Xcode beschikt over drie editors. De eerste gebruik je voornamelijk bij het schrijven van programmacode en bij het bouwen van de schermen van je app. Met de tweede kun je op een handige manier koppelingen maken tussen zichtbare onderdelen van je app (knoppen, labels enzovoort) en je programmacode.

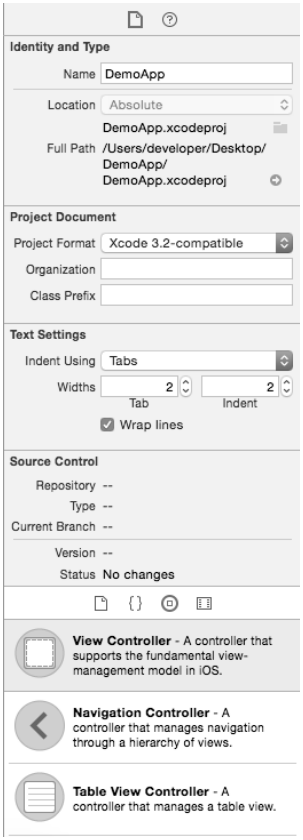
De derde editor gebruik je om verschillende versies van je programmacode met elkaar te kunnen vergelijken.

Om een andere editor in te schakelen gebruik je de Editor-knoppen rechts bovenaan op de werkbalk van Xcode (zie afbeelding 1.12).

Afbeelding 1.12 *Knoppen van de Editor.*

Utilities

Rechts op het scherm, naast de Editor, zie je het paneel Utilities (afbeelding 1.13). Wat er bovenaan in dit paneel wordt getoond, is context-afhankelijk: het hangt nauw samen met datgene wat je, bijvoorbeeld in de Navigator of in de Editor, hebt geselecteerd. In latere stappen komen we hier nog uitgebreid op terug.



Afbeelding 1.13 *Het paneel Utilities.*

In het onderste gedeelte van het paneel Utilities zie je de bibliotheek (*library*), met daarin objecten die je aan je app kunt toevoegen.

Als je meer schermruimte nodig hebt, kun je het paneel Utilities verbergen door in de werkbalk op de meest rechtse van de in afbeelding 1.10 getoonde knoppen te klikken. Er is ook een toetscombinatie die je daarvoor kunt gebruiken: Cmd+Alt+0.

Debug-venster

Onderaan in het werkvenster wordt, in sommige gevallen, het Debug-venster getoond (zie afbeelding 1.14). Het Debug-venster komt van pas als je bezig bent om de fouten uit je app te halen. Uiteraard komen we hier nog uitgebreid op terug.

Afbeelding 1.14 *Het Debug-venster.*

Als je het Debug-venster wilt verbergen (of tonen), kan dat door op de middelste van de drie knoppen uit afbeelding 1.10 (rechts op de werkbalk) te klikken. Ook hiervoor bestaat een toetscombinatie: Cmd+Shift+Y.

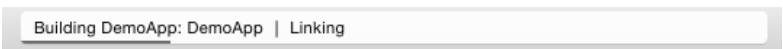
De eerste app starten

Nu we kennis hebben gemaakt met het werkvenster van Xcode, is het tijd geworden om onze eerste app te starten. Xcode heeft immers al, op basis van het sjabloon dat we aan het begin van deze stap hebben gekozen, een kant-en-klare app gemaakt. Veel doet deze app nog niet, maar we kunnen hem al starten en stoppen. En dat is precies wat we nu gaan doen.

Klik, in de werkbalk van Xcode, op de knop **Play**. Je kunt ook **Product**, **Run** in het menu kiezen of op Cmd+R drukken.

Achter elkaar gebeuren nu de volgende dingen:

1. Xcode bouwt een uitvoerbare versie van de app (zie afbeelding 1.15).

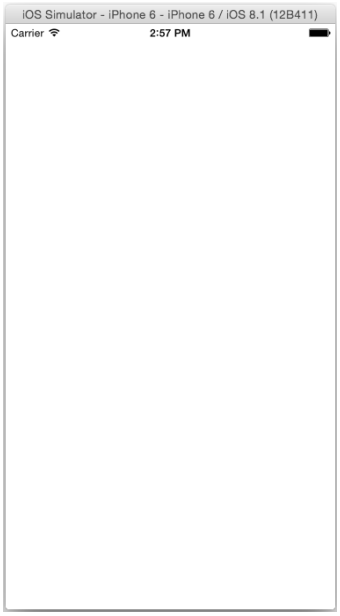


Afbeelding 1.15 *Xcode bouwt een uitvoerbare versie van de app.*

2. Xcode kopieert de app naar de Simulator, een bij Xcode behorend programma waarmee een iPhone en iPad kunnen worden nagebootst.
3. De Simulator wordt gestart. Als je Xcode (en de Simulator) voor de allereerste keer gebruikt kan het voorkomen dat je Xcode met je Admin-account toestemming moet geven om in te grijpen in het besturingssysteem van je Mac (zie afbeelding 1.16).

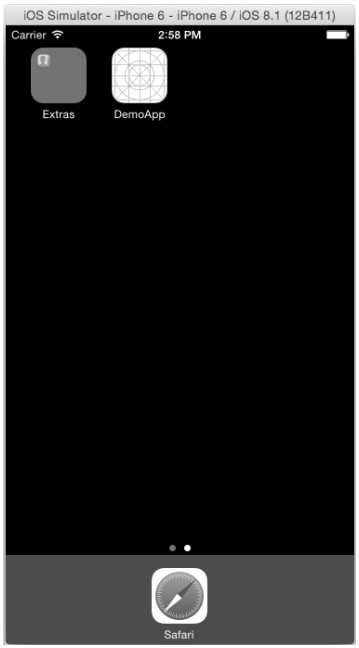
Afbeelding 1.16 *Toestemming verlenen.*

4. De Simulator start de app.



Afbeelding 1.17 *De app wordt gestart.*

Inderdaad: een wit scherm is alles wat je op dit ogenblik ziet. Maar Xcode heeft wel degelijk een app gebouwd. Dat kun je zien als je, uit het Hardware-menu van de iOS Simulator, de optie Home kiest (je kunt ook op Cmd+Shift+H drukken).



Afbeelding 1.18 *Daar staat de app.*

De app is keurig zichtbaar op het tweede Home-scherm van de Simulator, met de naam die we er in de vorige stap aan hebben gegeven: DemoApp.

Meer over de Simulator

De Simulator is een handige, eigenlijk onmisbare app om je eigen apps te kunnen testen. Hoewel de Simulator een Mac-app is, gedraagt hij zich als een echte iPhone of iPad. Je kunt alle bewegingen die je op je iPhone of iPad met je vinger doet met de muis uitvoeren. Voor de Home-knop en de Vergrendelknop zijn zowel menuopties als toetscombinaties beschikbaar.

Met de muis ergens naar wijzen en erop klikken heeft hetzelfde effect als wanneer je op een echte iPhone/iPad met je vinger iets aanraakt. Probeer dat maar eens door op het DemoApp-pictogram te klikken.

Op de Home-knop druk je met de toetscombinatie Cmd+Shift+H.

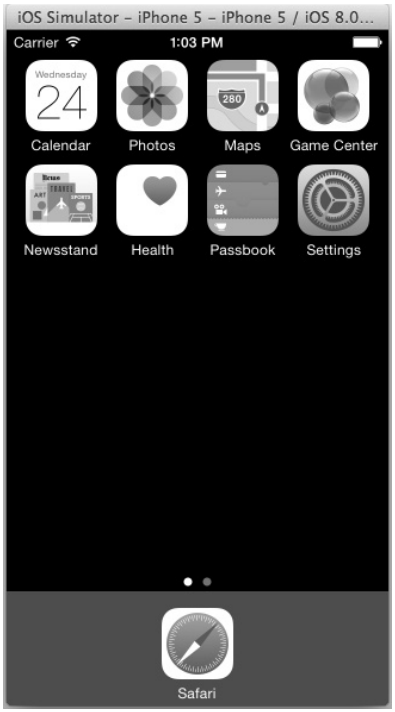
Slepen doe je door de linkermuisknop ingedrukt te houden en de muis te bewegen. Zoomen doe je door, terwijl je Alt ingedrukt houdt, de muis te bewegen. Er verschijnen dan twee grijze cirkels op het scherm van de Simulator, die laten zien in hoeverre je in- of uitzoomt (zie afbeelding 1.19).

Afbeelding 1.19 *Zoomniveau.*

Je kunt de Simulator ook 'kantelen', waarmee je doet alsof je je iPhone of iPad naar links of rechts draait. Hiervoor kun je de opties uit het Hardware-menu gebruiken: **Rotate Left** en **Rotate Right**. Er zijn ook twee toetscombinaties voor beschikbaar: Cmd+Pijl-links en Cmd+Pijl-rechts.

De Simulator als ander apparaat gebruiken

In de voorgaande afbeeldingen werd de Simulator gebruikt om een Engelstalige iPhone 6 te simuleren. Je kunt echter ook voor een ander apparaat kiezen, bijvoorbeeld een iPhone 5. Dit doe je met de optie **Device** in het menu **Hardware**. In afbeelding 1.20 zie je hoe het Home-scherm van de iPhone 5 wordt gesimuleerd door de optie **iPhone 5** te kiezen.



Afbeelding 1.20 *Simulatie van de iPhone 5.*

Taal instellen

Het instellen van de taal doe je, net als op een echte iPhone, via de app Instellingen (in het Engels **Settings**). Standaard staat de iPhone Simulator op Engels ingesteld, maar niets houdt je tegen om er een Nederlandse versie van te maken.

De Simulator stoppen

Om de Simulator te stoppen ga je terug naar Xcode en druk je op de werkbalk op de knop **Stoppen**. Je kunt ook op Cmd+punt drukken of **Product**, **Stop** kiezen. Niet alleen stop je hiermee de Simulator, ook keer je terug naar de Editor van Xcode, zodat je verder kunt werken aan je app.

En dat is precies wat we hierna gaan doen. Onze app doet het, maar er is nog niet veel te zien; een leeg, wit scherm. In het volgende gedeelte laten we zien hoe je dat scherm met behulp van Xcode kunt aanpassen.

De eerste aanpassingen aan de app

Inmiddels heb je gezien hoe je een app bouwt en hoe je deze kunt starten in de Simulator. De app die gebouwd is was gebaseerd op een van de eenvoudigste sjablonen van Xcode: een **Single View Application** (een app met één *view*: één scherm).

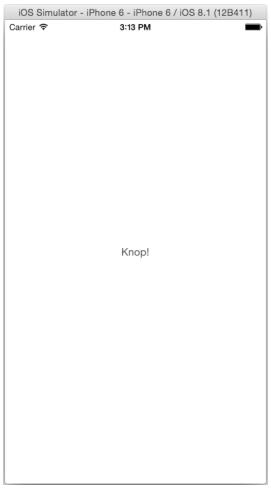
Nu gaan we het uiterlijk van onze app aanpassen. De achtergrondkleur van de app maken we geel en we zetten er een label met rode tekst op.

Views

Een van de termen die je vanaf nu steeds vaker zult tegenkomen, is *view*. Een app is opgebouwd uit een of meer views. Een view is een rechthoek van willekeurige afmetingen waarin informatie kan worden getoond. Het witte scherm dat we hebben gezien toen we onze app voor het eerst hebben gestart is ook zo'n view: eentje die het gehele scherm van de iPhone beslaat en waarvan de achtergrondkleur wit is.

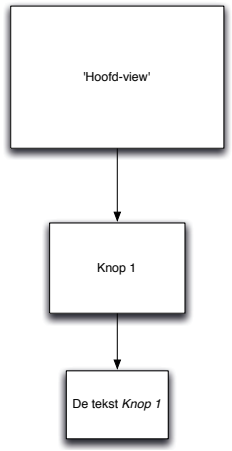
Zoals gezegd: alles wat je op het scherm van een iPhone/iPad ziet bestaat uit een of meer views. De *controls* (knoppen, schuifregelaars, spinners enzovoort) die je ziet zijn dus ook views, die op hun beurt weer uit een aantal *subviews* bestaan. Een view kan dus een of meer subviews hebben, maar een view heeft altijd maar één *superview* (de view waar hij van afstamt).

Een (wat vereenvoudigd) voorbeeld zie je in afbeelding 1.21: een iPhone-app met slechts één knop.



Afbeelding 1.21 *Een iPhone-app met slechts één knop*

De knop is een subview van de hoofdview (in het Engels *main view* genoemd) van onze app: de view met de witte achtergrondkleur. De tekst op de knop is een subview van de knop zelf. Er is dus sprake van drie views: de tekst 'Knop!' is in dit voorbeeld een subview van de knop zelf. De knop zelf is dus de superview van de tekst 'Knop!'. De knop zelf is een subview van de hoofdview van onze app; deze hoofdview is dus de superview van de knop. In afbeelding 1.22 zie je daarvan een schematische weergave.

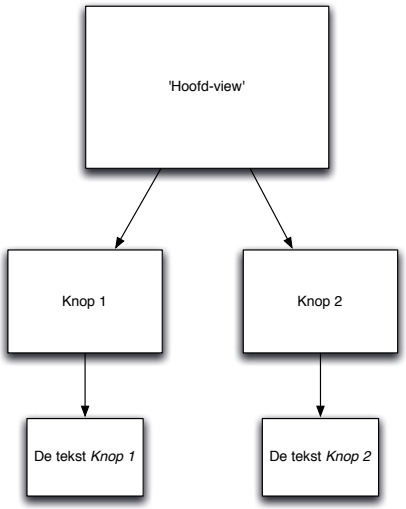


Afbeelding 1.22 *Views schematisch weergegeven.*

Nog een voorbeeld zie je in afbeelding 1.23: een app met twee knoppen, **Knop 1** en **Knop 2**.

Afbeelding 1.23 *Een app met twee knoppen.*

De hoofdview heeft nu twee subviews: twee knoppen. Beide knoppen hebben elk ook weer een subview: de tekst die erop staat. Schematisch ziet de view-hiërarchie er nu dus uit als in afbeelding 1.24.



Afbeelding 1.24 *Schematische weergave van de views in de app met twee knoppen.*

- De hoofdview heeft twee subviews: **Knop 1** en **Knop 2**.
- **Knop 1** heeft één subview: de tekst **Knop 1**. Ook **Knop 2** heeft één subview: de tekst **Knop 2**.
- De tekst **Knop 1** heeft geen subviews. Wel heeft deze tekst een superview: **Knop 1** zelf.
- De tekst **Knop 2** heeft ook geen subviews, maar wel een superview: **Knop 2** zelf.
- De knoppen **Knop 1** en **Knop 2** hebben beide dezelfde superview: de hoofdview.

Zorg ervoor dat je de begrippen subview (een view die deel uitmaakt van een andere view) en superview (de view waarvan de huidige view deel uitmaakt) goed begrijpt: je komt ze namelijk nog tientallen malen tegen.

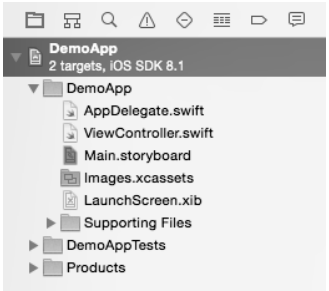
De app aanpassen

Zoals gezegd gaan we de app op de volgende manieren aanpassen:

1. We veranderen de achtergrondkleur van de hoofdview (*main view*) in geel.
2. We zetten er een label op met een rode tekst.

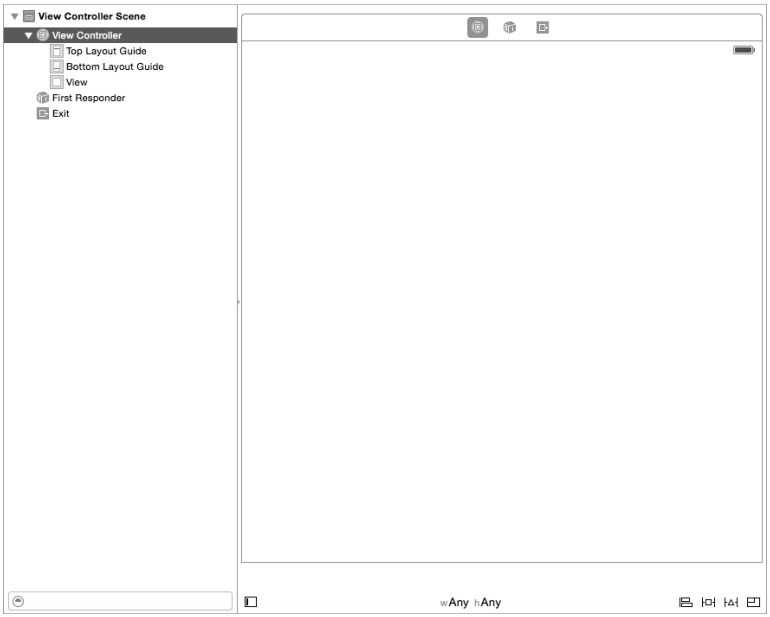
Ga naar Xcode. Als de DemoApp nog actief is, stop hem dan door op **Stop** te klikken (of met behulp van de toetscombinatie Cmd+punt).

We maken nu kennis met het allereerste bestand van onze app: het *storyboard*. Zorg dat de Navigator zichtbaar is en kies de Project Navigator door op het meest linkse pictogram in de Navigator-werkbalk te klikken (of gebruik de toetscombinatie Cmd+1). De Navigator ziet er nu uit zoals afbeelding 1.25. Vouw, indien nodig, de DemoApp-groep open.



Afbeelding 1.25 *De navigator.*

In de DemoApp-groep zie je vijf bestanden staan. Het gaat nu om het derde bestand, met de naam **Main.storyboard**. Klik op dat bestand, waarna de inhoud van de editor verandert (afbeelding 1.26).



Afbeelding 1.26 *Weergave in de editor.*