

Zelf apps maken lastig? Nee hoor! Zonder te programmeren kunnen je leerlingen tegenwoordig een echte app maken! Of misschien heb je zelf als docent een goed idee! Eerder maakten we al [een lijstje](#) met voorbeelden waarbij dat veel makkelijker is. En in dit artikel kijken we naar 4 nieuwe opties, die nog beter te gebruiken zijn. Zo maken je leerlingen (of jij!) echte apps, en dat zonder te programmeren!

Zonder code

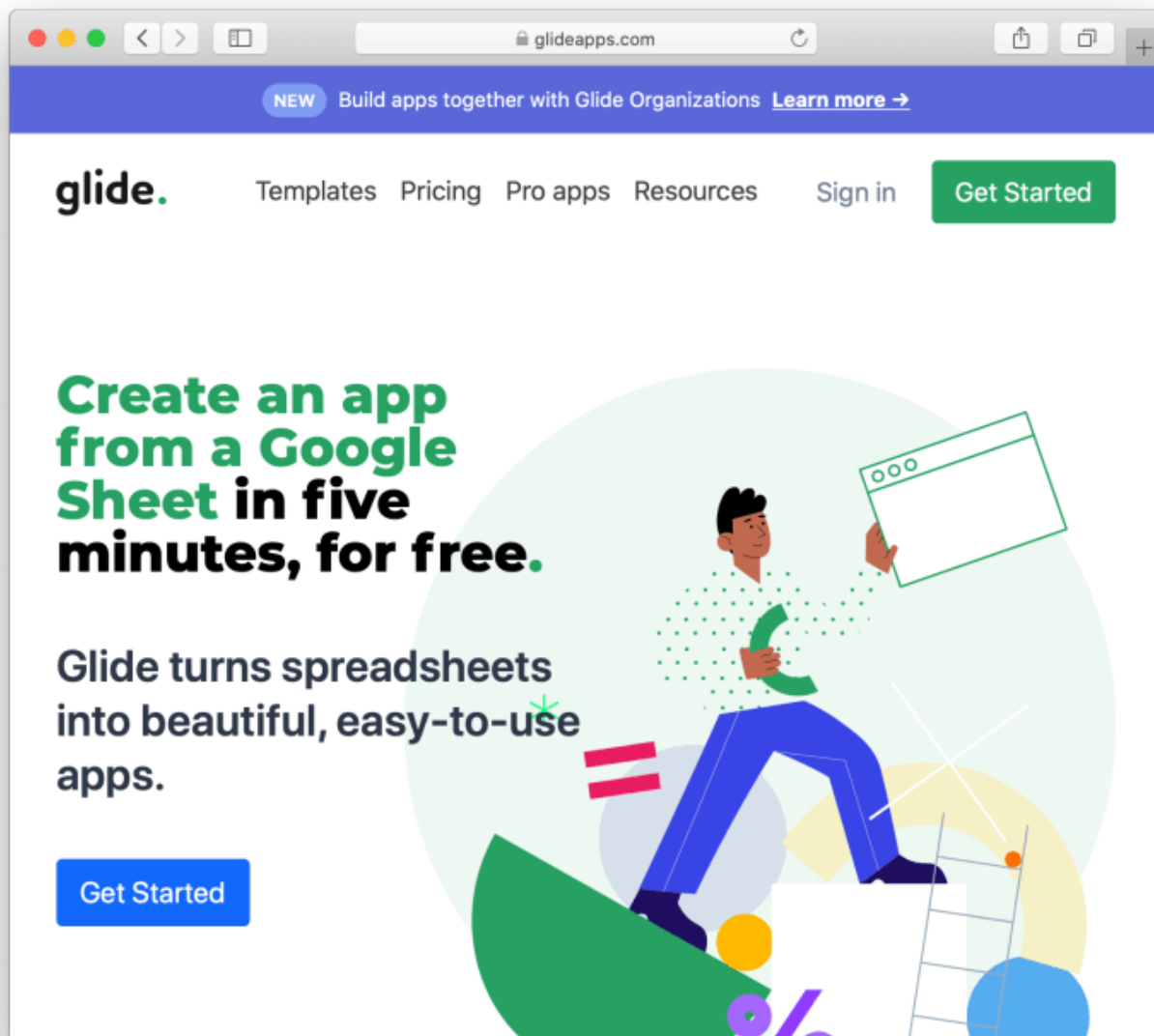
Er is een nieuwe categorie software aan het ontstaan, de 'no-code tools'. Het mooie aan deze nieuwe tools is dat er geen [programmeerkennis](#) meer nodig is. Dus geen commando's typen, maar wel met een muis een 'interface' bij elkaar klikken. Klinkt al een stuk beter, toch? Het belangrijkste is dan niet meer de programmeerkennis. Het belangrijkste is een [goed idee](#) voor een app!

Dat betekent niet meteen dat het voor iedereen haalbaar is om een ingewikkelde app te bouwen. Sterker: ook hier moet je beschikken over de [\(denk\)vaardigheden](#) die nodig zijn bij 'gewoon' programmeren. Verbanden leggen, algoritmisch denken, complexe taken opbreken: allerlei vaardigheden die horen bij [computational thinking](#). En bij het maken van een app oefen je precies die vaardigheden. Het maken van apps past dan ook goed bij [digitale geletterdheid](#).

Voorbeelden

Hieronder 4 verschillende tools die in basis gratis te gebruiken zijn. Het zijn stuk voor stuk programma's die echt professioneel gebruikt worden. Toch zijn ze eenvoudig genoeg om in te zetten met leerlingen. En wie weet heb jij als docent wel een top-idee voor het maken van een app!

Glide

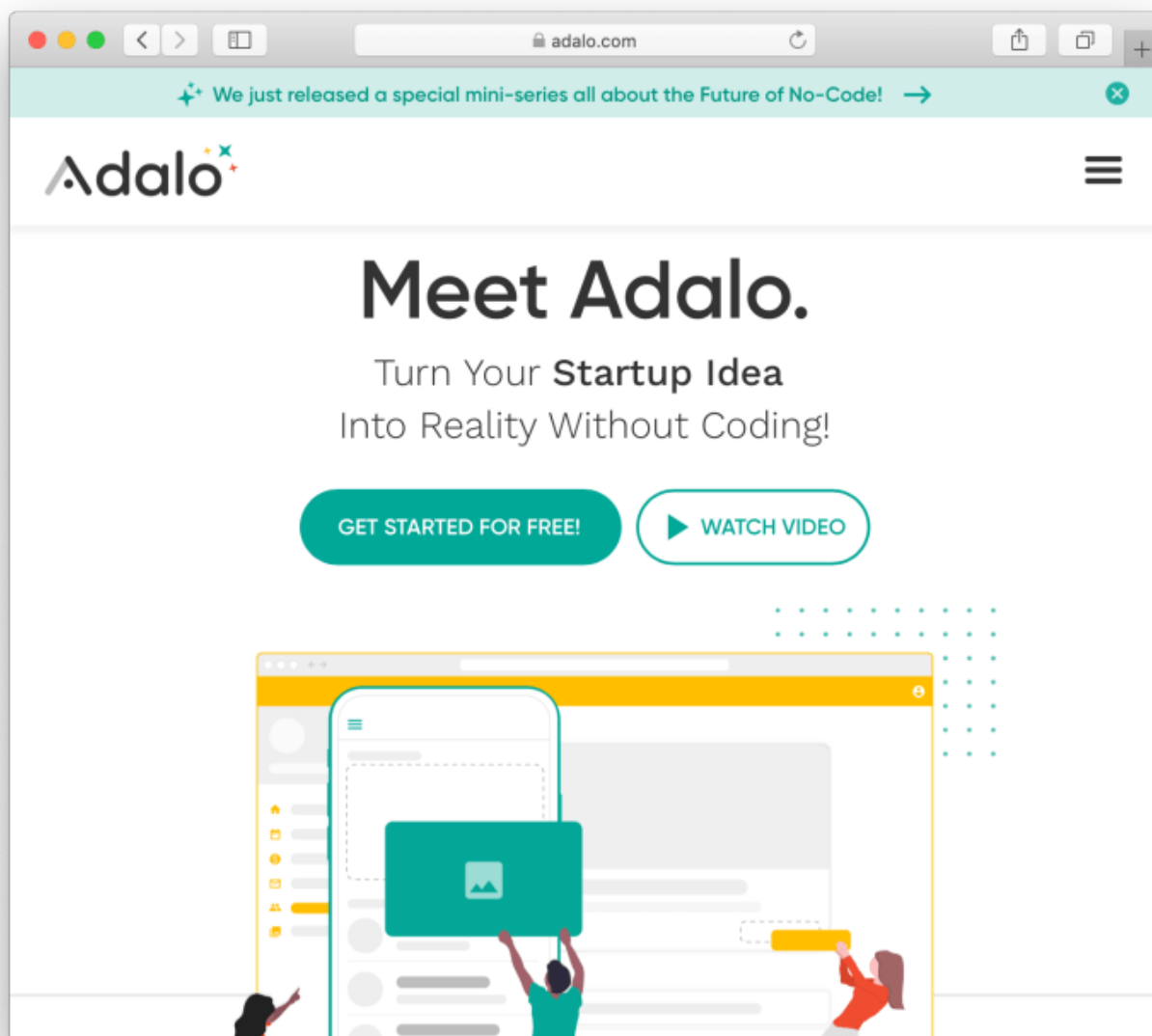


Heb je een hoop informatie waarmee je iets praktisch wil doen? Glide is een tool waarmee je vanuit een Google Sheets een app maakt. De 'voorkant' van de app klik je bij elkaar, terwijl de data in een Google Sheets bestand staat. Glide werkt heel eenvoudig, maar het is wel belangrijk om eerst de informatie op orde te hebben.

Met Glide maak je geen app voor in de app-store, maar wel een 'web-app'. Dit is een soort website die je als app op je telefoon kunt zetten, zonder dat het via de app-store geïnstalleerd moet worden.

[\[Glide: apps maken vanuit een Google Sheet](#)

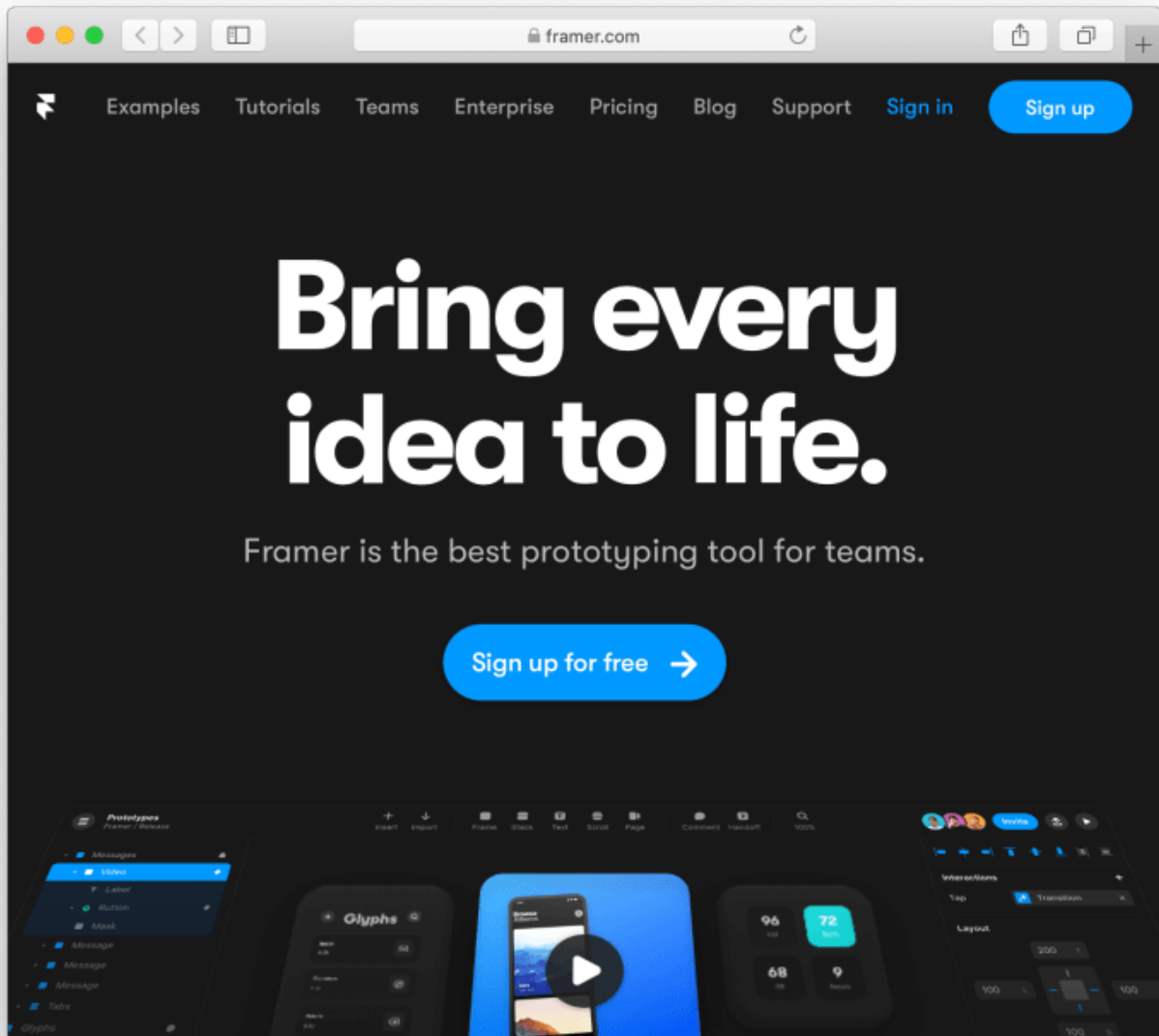
Adalo



Wil je een echte app maken en die uitbrengen in de app-store? Dat kan met Adalo. Ook hier kun je een data-base gebruiken, maar je kunt ook verschillende elementen samenvoegen en zo een app ontwerpen. Adalo is iets lastiger in het begin, en kent ook beperkingen, maar het werkt daarna praktisch en snel. Je hebt snel resultaat en het geheel ziet er ook netjes uit.

[Adalo: Apps maken zonder code](#)

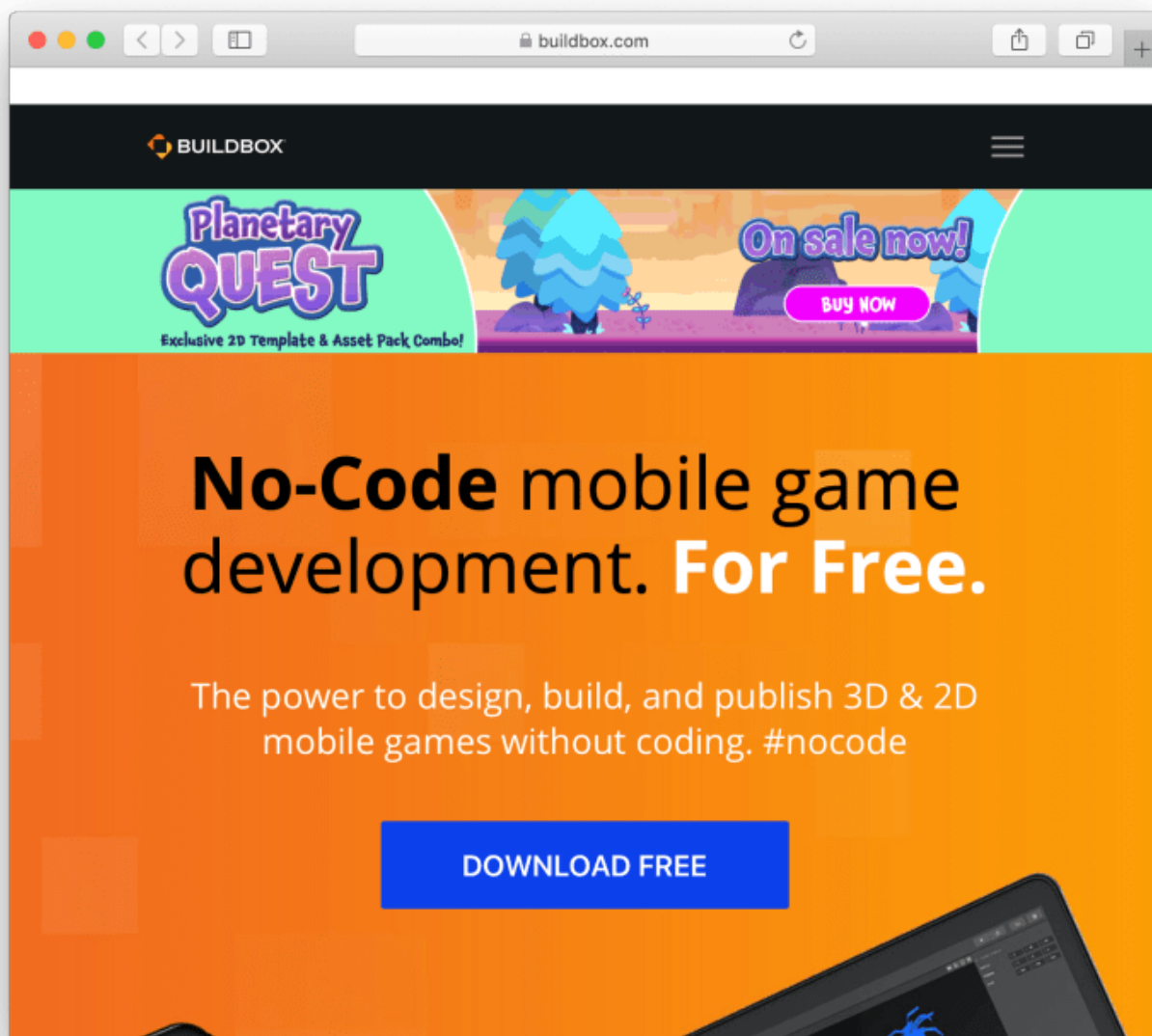
Framer



Een product van Nederlandse makelij! Framer zet vooral in op de mogelijkheden tot samenwerking en de animaties in de app. Het ziet er zo gelikt en mooi uit, terwijl het ook erg bruikbaar is. Framer is eenvoudig in gebruik en richt zich op het maken van prototypes. Hierbij krijg je dus geen hele app, maar kun je wel goed laten zien wat voor idee je hebt. Met de 'Framer Preview' app kun je de apps meteen testen op je eigen telefoon.

[Framer: Samenwerken aan prototypes met animaties](#)

Buildbox



Leerlingen die games willen maken? Dan is Buildbox een mooie 'no-code' optie! Ook in de gratis versie kun je bijna alles doen, en dat zonder te programmeren. De instap is wat hoog, maar na het volgen van de tutorials kom je al heel ver. Buildbox wordt gebruikt om games te maken voor telefoons en tablets, en je kunt het dan ook meteen in de app-stores aanbieden (en verkopen!). Een mooi informatica project voor in de bovenbouw dus.

[\[Buildbox: games maken voor telefoons\]](#)

Tips bij het maken van apps

Ga je leerlingen voor het eerst een app of game laten maken? Voor de meeste leerlingen is het niet iets waar ze direct een idee voor hebben. Ze willen misschien heel graag, maar zonder idee is het moeilijk om tot een goed eindproduct te komen. Het klinkt suf, maar daarom nog een aantal praktische tips:

- **Eerst het probleem, dan de oplossing:** Een app is vaak vooral een oplossing voor een probleem. Ga dus eerst opzoek naar het probleem! Hoe duidelijker en kleiner het probleem, hoe eenvoudiger om er een app voor te maken. Denk aan een persoonlijke studeer-app, een app speciaal voor foto's voor je oma, of een familie-agenda-en-berichten-app.
- **Schets, schets, schets!** Het ontwikkelen van een app gebeurt vaak eerst op papier. Laat leerlingen schetsen maken van hoe de app er uit moet zien en moet werken. Hierdoor maak je visueel waar de verbanden zitten, en hoe het moet werken. Dit helpt enorm wanneer het project wat moeilijker wordt.
- **Niet alles kan, maar je kan het wel proberen:** Niets zo vervelend als leerling als een idee wat afgekeurd wordt. Begrijpelijk: soms is een idee gewoon te complex. Maar je kunt altijd een eerste stap zitten binnen een groot en ingewikkeld idee. Een nieuwe instagram ontwerpen? Begin met het kleinste stukje instagram wat je anders zou willen hebben.
- **Wees een rubber-eend:** 'Rubber duck debuggen' is een strategie die programmeurs gebruiken om moeilijke problemen op te lossen. Het idee: wanneer je vast zit leg je het probleem uit aan een bad-eendje. Door het probleem eenvoudig te verwoorden ontdek je vaak zelf al de oplossing. Wees als docent de bad-eend. Stel vragen, laat leerlingen zelf vertellen en uitleggen. Vaak vinden ze zo al het juiste antwoord.
- **Leer samen:** een nieuwe tool in de klas kan je als docent onzeker maken. Weet je wel meer dan de leerling? Durf dit aan, spreek dit uit, en leer samen. Zo stimuleer je leerlingen ook om onderling meer kennis uit te wisselen. Als docent kun je zo meer focussen op de denkvaardigheden, en minder op 'klik hier, en daarna daar'.

Met bovenstaande apps en tips kun je als docent snel aan de slag, ook op afstand. Het fijne is dat leerlingen er ook thuis aan kunnen werken! En ook als docent zijn het mooie tools. Denk bijvoorbeeld aan een glide-app om je lesmateriaal te organiseren! Hoe dan ook: de mogelijkheden zijn enorm! Programmeren is niet meer het probleem om dit soort producten te maken, en de vaardigheden binnen [computational thinking](#) te verbeteren.