So you wanna be a WordPress Developer?

WordPress Sheffield Meetup - March 2017

Kimb Jones
Co-founder & Managing Director

make do

http://makedo.net
kimb@makedo.net
@makedoers

WHY?...

makedo

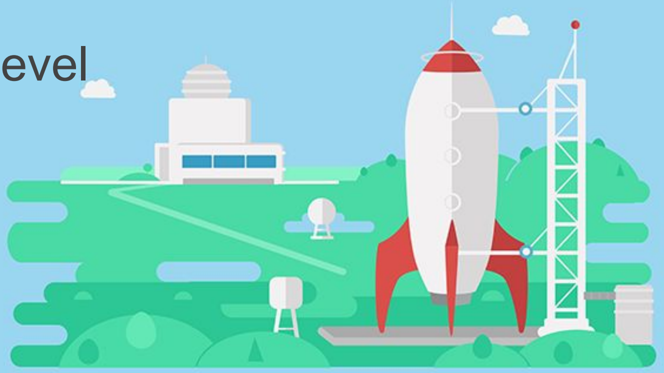
What exactly is A WordPress Developer?

make do

Plugins, themes and everything in between

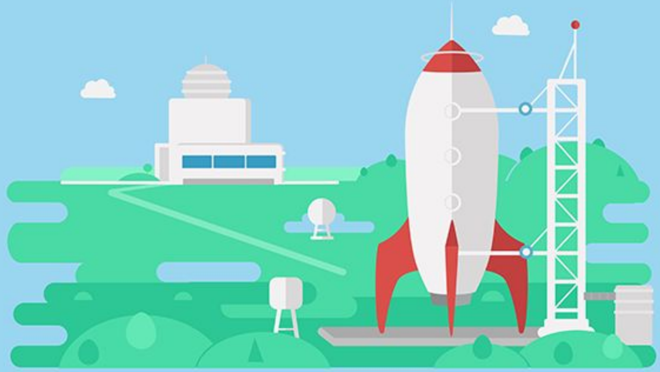
Plugin Developer: *Extending WordPress*

- Bend WordPress to your will
- Develop custom functions for WordPress
- Extend the core features
- Extend other plugins and create new features
- Integrate with other systems and apps
- Generally a very complex area
- Requires programming skill and a talent for code
- Known as a ‘back-end’ developer or ‘application’ level
- Good plugin developers are a rare breed



Theme Developer: *Theming WordPress*

- Turn any design/layout into a WordPress site
- Simpler than developing plugins (generally)
- Does not require any design skill but this helps
- Good for beginners
- Themeing can help extend basic HTML/CSS skills
- Lots of options with frameworks, parent/child themes, theme builders etc
- Tonnes of online tutorials and resources
- How I got started in WordPress *back in 2005*

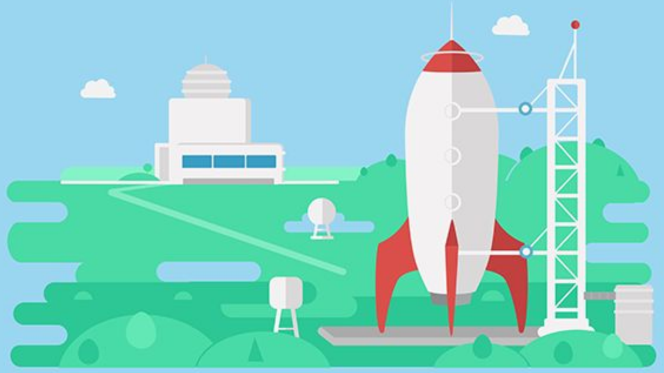


All-Rounder: *Everything WordPress*

- Themeing and Plugin coder, jack of all trades
- For when you can’t decide between the two
- Most professional WordPress developers will switch between these roles
- Themes are nothing without plugins
- Plugins are nothing without themes
- Common among solo-developers and freelancers
- Tools available which help like ACF, SiteOrigin, etc
- Works well for small sites, publishing and basic CMS sites
- Difficult to scale this skillset, especially working alone

Assembler/Hacker: *Not a Developer*

- Not to be confused with an actual developer (bespoke custom code)
- Good knowledge of WordPress
- Uses off-the-shelf themes and plugins
- Experience with servers/hosting
- Good with HTML/CSS
- Limited to what plugins/themes can offer
- Good for designers who want to ‘skin’ code
- Good for low-budget, quick projects
- Can often be a frustrating choice as the limitations often catch up on you



Which are you?

And which do you want to be?

Questions...

Becoming a ‘web’ developer

One and the same

The *fundamentals*

make do

HTML

CSS

JS

PHP

SERVERS

DATABASES

VERSION CONTROL

HTML/CSS

Powers 100% of the web

JS

JavaScript (classic)

JS (again)

JavaScript (applications)

PHP

What powers WordPress

SERVERS

Apache, Nginx, IIS, Azure

&

MAMP, Vagrant (vvv), DesktopServer

SERVERS...cont

SSH, FTP, SFTP, SSL, HTTPS, .htaccess.... *etc*

DATABASES

MySQL, phpMyAdmin,
Sequel Pro (Mac), MySQL Workbench (Windows)

VERSION CONTROL

Git & SVN

VERSION CONTROL

Git ~~& SVN~~

Questions...

Learning

Tutorials and resources...

Make WordPress

https://make.wordpress.org/

Theme Shaper

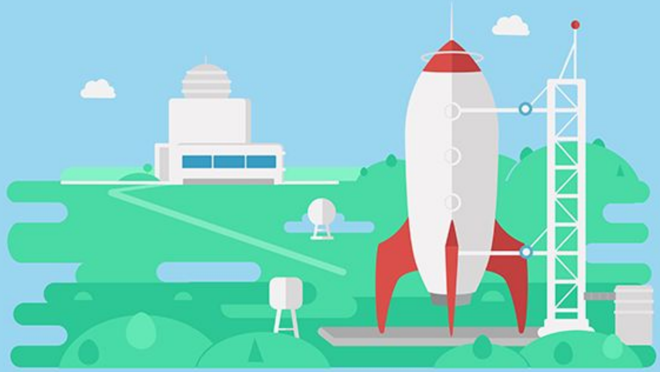
https://themeshaper.com/

Tuts+ WordPress

https://code.tutsplus.com/categories/wordpress

In-person Training

https://mikelittle.org/wordpress-training/



Um, Google!

“wordpress development tutorial”

Attend a meetup / WordCamp!

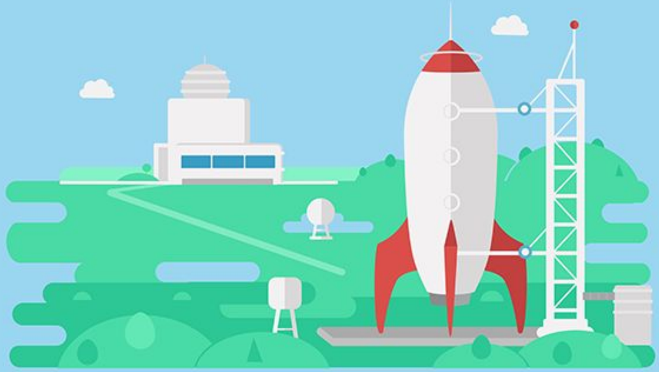
You’re doing the right thing!

Ask people you know

WordPress-ers love to share...

Jump in

Best way is to learn doing...



Questions & Break...

makedo

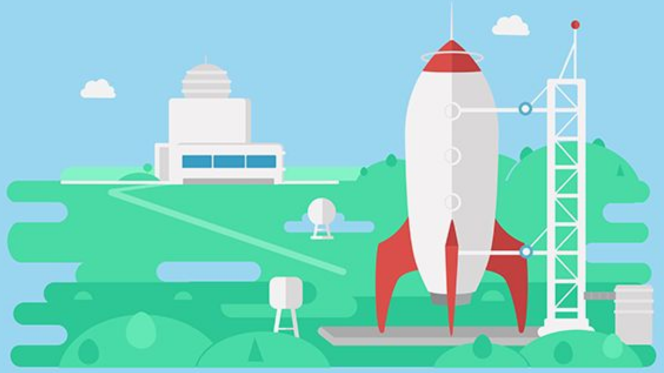
Starter Pack Checklist

Go away and do this…

The basics

Getting started hacking code

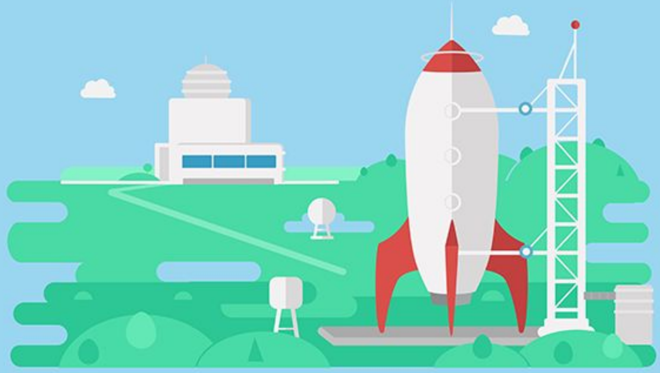
1. Setup a local server (MAMP, VVV, DesktopServer)
2. Install WordPress locally (from scratch)
3. Get a decent editor/IDE (SublimeText, Atom)
4. Download the _s theme
    a. Rename and install a custom version
    b. Explore the file structure
    c. Hack the code and see what happens
    d. Look for other ‘theme boilerplates’ online to help get you started
5. Open up the Hello Dolly plugin
    a. Examin how it interacts with WordPress
    b. Hack the code to make it display new text
    c. Look for ‘plugin boilerplates’ online to help get you started
6. Sign up for a cheap webhost
    a. Dreamhost, HostGator, SiteGround
    b. Setup WordPress externally
    c. Upload your custom theme and/or plugin



Version control

The important next step

1. Learn some basic CLI commands (Command Line Interface)
    a. Use iTerm or PowerShell to improve the experience
2. Install Git locally
3. Sign up for a GitHub account
    a. Create a new repo
    b. Follow the instructions to get your local code into the external repo
    c. Learn how to push, pull and branch in git and push to different versions of your code
4. Experiment Git into your work
    a. Share with others
    b. Create Pull Requests from other repos
    c. Learn the basic Git commands
        i. $ git add --all
        ii. $ git commit -m “your message”
        iii. $ git push [location] [branch]
        iv. $ git pull [options[
        v. $ git branch [branchname]
        vi. $ git checkout [branchname]
        vii. $ git merge [branch-a] [branch-b]



GO! Jump in!

Use WordPress for a real project

Over to you...

How did you learn WP?

Thank you...

Kimb Jones
Co-founder & Managing Director

make do

http://makedo.net
kimb@makedo.net
@makedoers