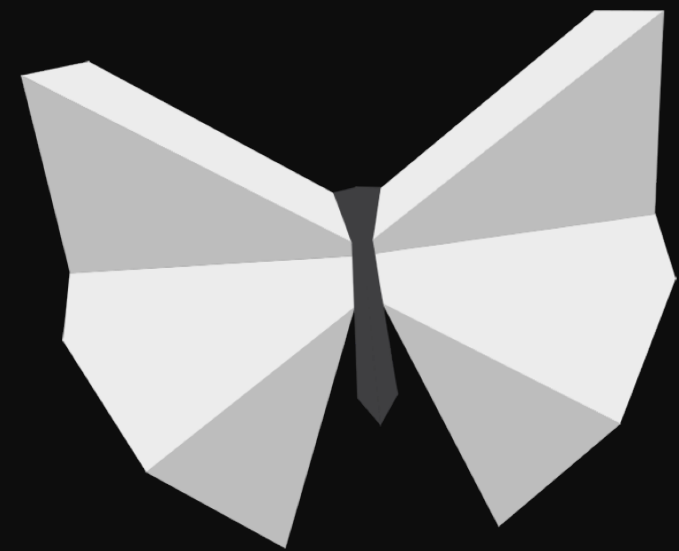


The first layer of Cake

An introduction to C# driven build orchestration

@devlead - Mattias Karlsson



SWETUGG

[illegible]

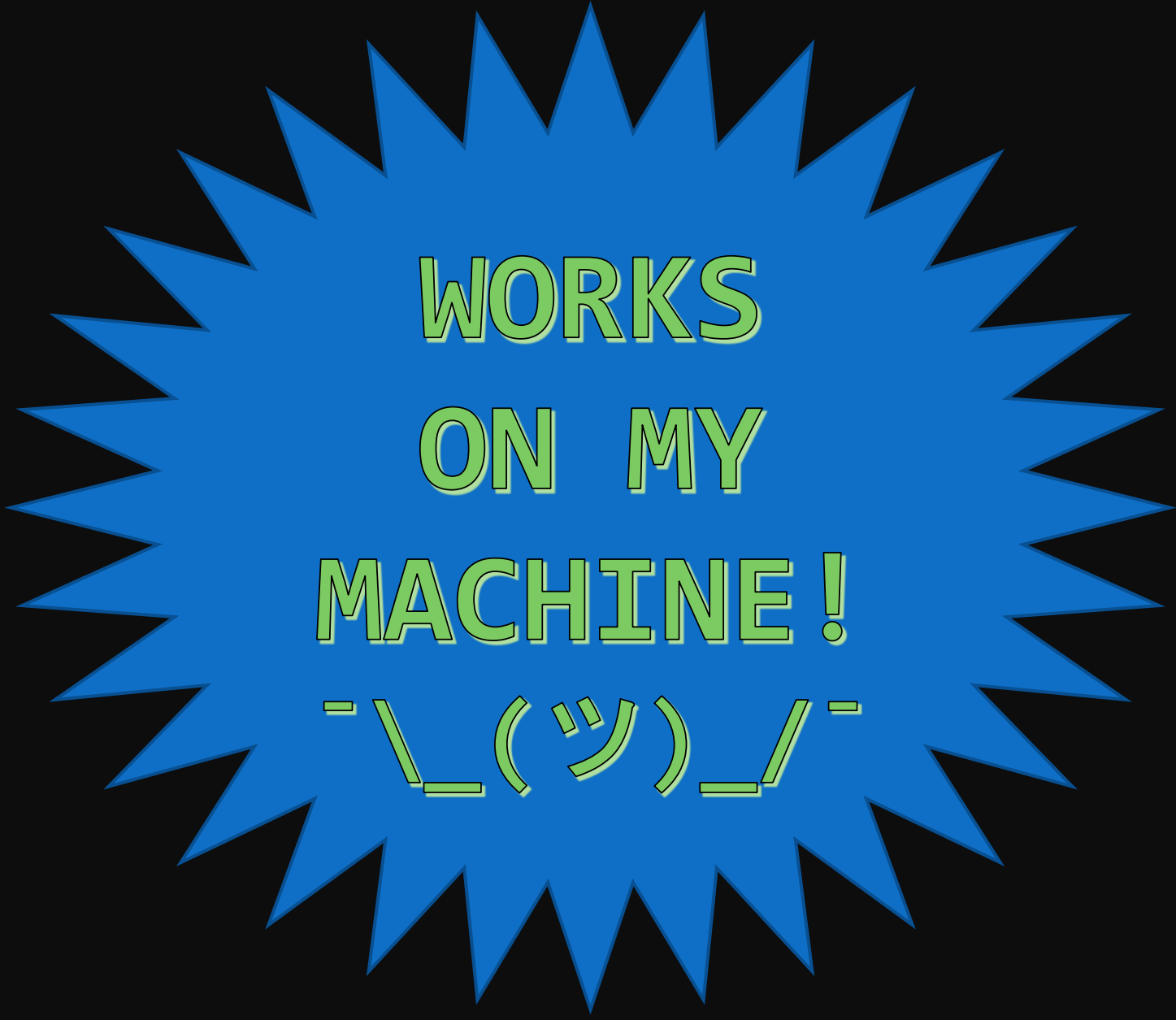
Welcome !

```

+##      #;;'
      #;;# .+;;;+;
      '+;;#;,+';;;';'#.
      ++''''';;;;';# ;#;
      ##';;;++'+#;;;';' . #;
      ;#      '+''';;;;';#`      #.
      `# ,      .'++;;;';:.....#
      '+      `.....';;;';:.....#
#.....+;;;';:.....#
#.....#';;;';'+'''''.#
#.....;';''''''##';;;';'+'''''#,
#''''''''''''''''###';;;';+'''''#
#''''''''''''''''####';;;';#'''#
#''''''''''''''''#####';;;';#'''#
#''''''''''''''''#####';;;';#'''#
#''''''''''''''''#####';;;';##
#''''''''''''''''#####';;;';#
#'''''''''''''''''+#####';#####';;;';#
#+####':,`      ,#####';;;;';'
                        +##'''''+

```

Chivalry



WORKS
ON MY
MACHINE!

—_ (ツ) _ /—



WORKS ON

ANY

MACHINE!

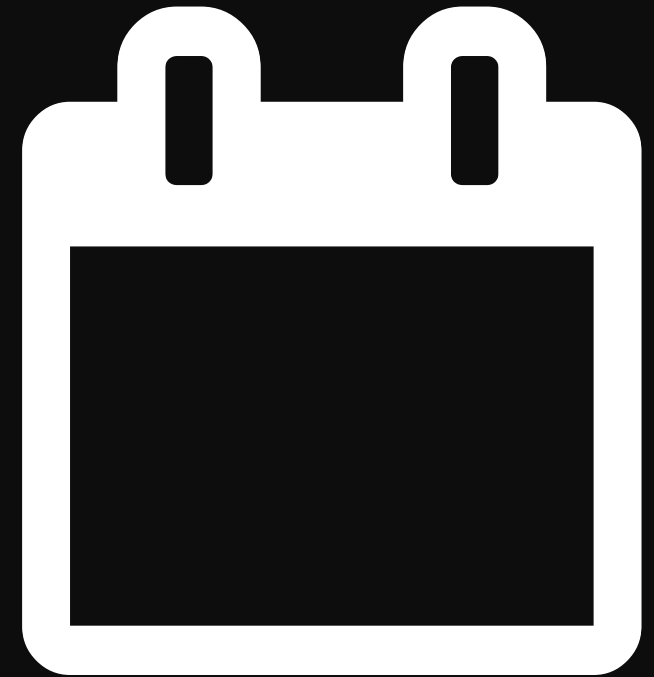


Agenda

CI / CD why?

Cake fundamentals

Cake demos



Mattias Karlsson

🏆 Microsoft Azure MVP

✂️ OzCode Magician

👤 Sr. architect, developer & partner at WCOM AB

🎮 Started coding late 80's on the C128

♥️ First love was the Amiga 500

👨👩👧👦 Husband of 1 & father of 2

👥 Maintainer & Contributor of Cake



Why CI / CD?

Reproducible

Quality

Confidence

Fidelity



"The definition of insanity is doing the same thing over and over and expecting different results."

-Unknown

"The definition of sanity is doing the same thing over and over and expecting same results."

- Mattias, 2018



The traits & goals of Cake

Capable

Extensible

Modular

Configurable

Open Source

Communital

Everywhere

Available



Capable

Supports many common build tools / tasks

MSBuild / Xbuild

.NET Core CLI

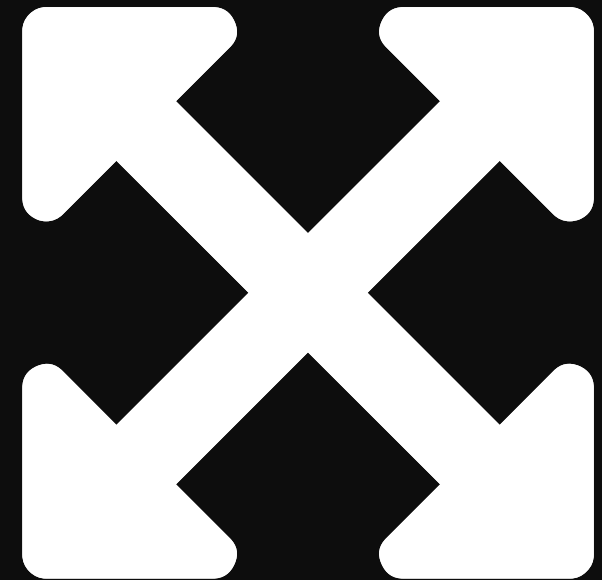
NUnit / Xunit / MSTest / VSTest

NuGet / Chocolatey / ZIP

Globber / Cleaning

etc.

400+ built-in aliases



Extensible

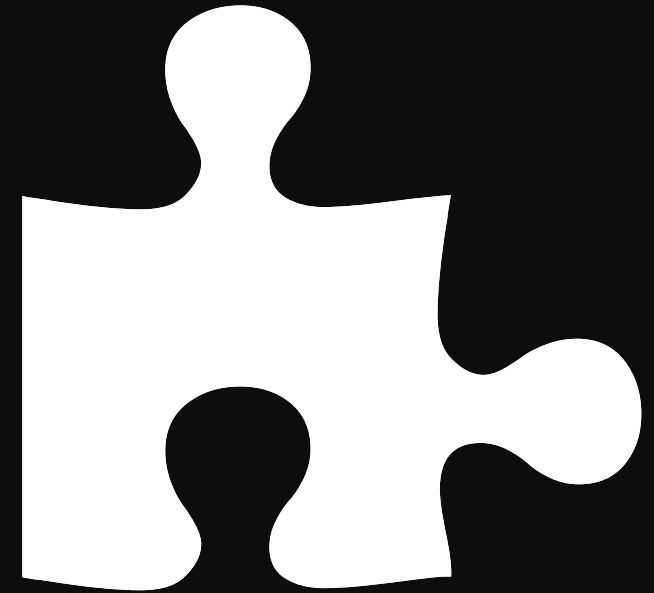
NuGet

.NET Assemblies

Addins

Scripts

Tools



Modular

Replace / Extend internals of Cake

Add core functionality

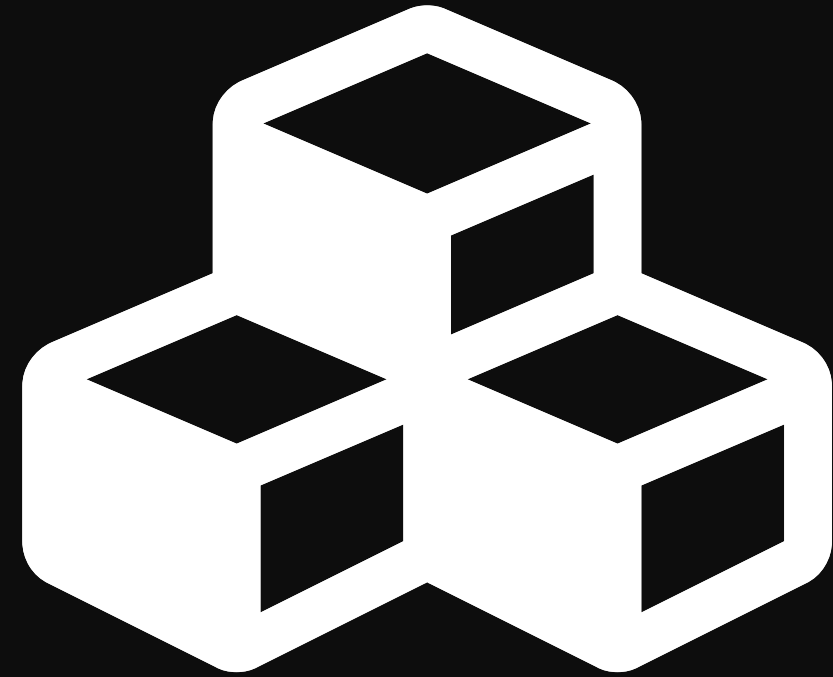
Tool resolution

Preprocessor directives

File system

Logging

etc.



Configurable

Cake lets you through

- Environment variables

- Configuration file

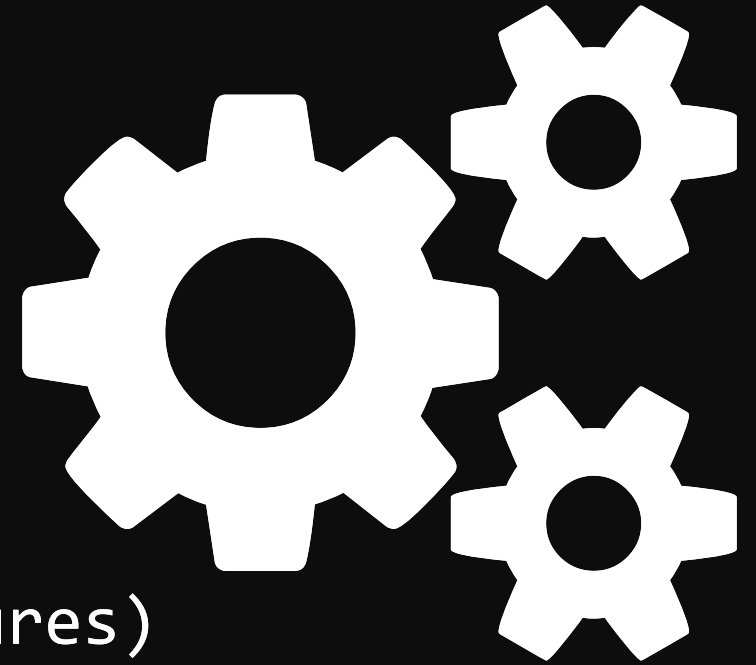
- Command line arguments

Control things like

- Default behaviors (opt in/out features)

- Default paths for tools, addins & modules

- Default NuGet sources



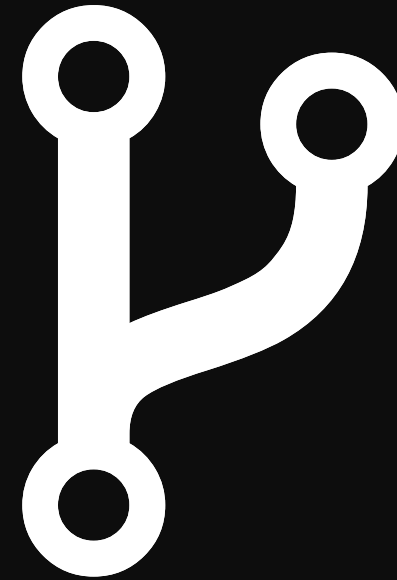
Open source

Permissive MIT License

Source available on GitHub

Developed in the open

Member of the .NET Foundation



.NET Foundation

Ensures longevity and stewardship of the project

Make .NET a credible open source development platform

Provide support to open source projects in the .NET ecosystem

Develop the grassroots .NET community

Takes on copyright and legal support

Does NOT take control of the project



The Team



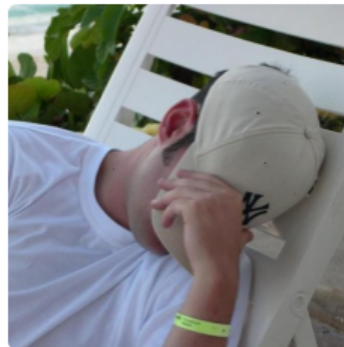
Patrik
patriksvensson

May
2014



Mattias Karlsson
devlead

September
2014



Gary Ewan Park
gep13

October
2015



Alistair Chapman
agc93

September
2016



Martin Björkström
mholo65

April
2017

Communital

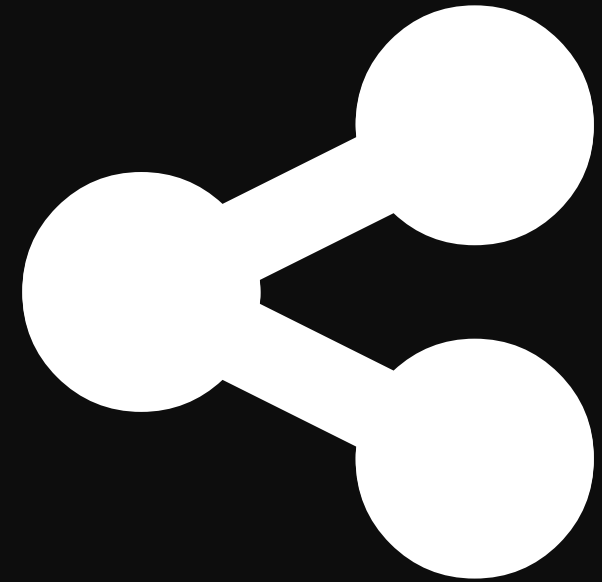
Welcoming

Approachable

Civil

Helpful

Awesome



Everywhere

Cross platform

Cross environment

Cross runtime

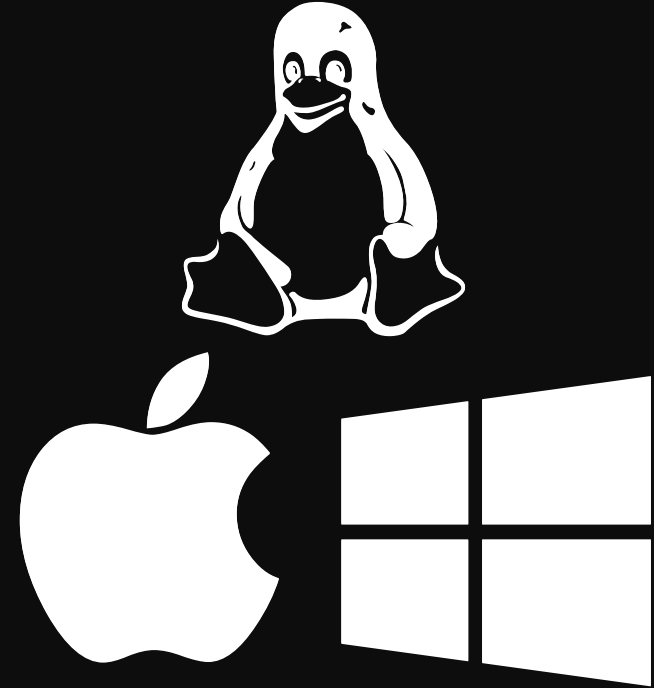


Cross platform

Mac

Linux

Windows



Cross environment

AppVeyor

MyGet

Bamboo

TeamCity

Bitbucket Pipelines

TFS

Bitrise

VSTS

ContinuaCI

TravisCI

GitLab

Developers machine

GoCD

Jenkins

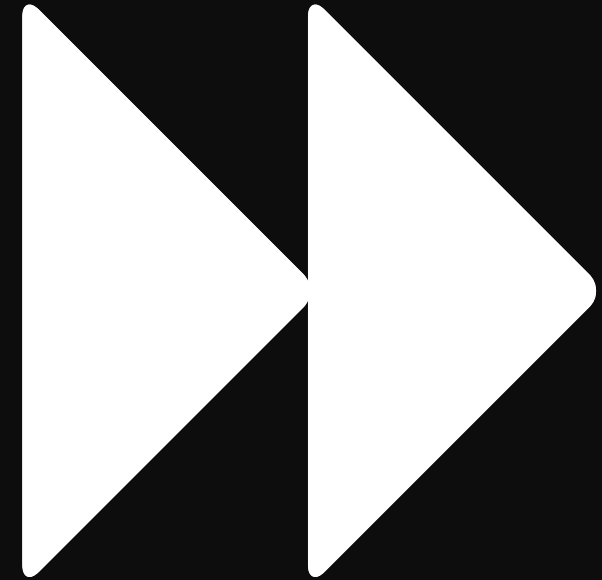


Cross runtime

.NET Full / Desktop

.NET Core

Mono



Available

NuGet

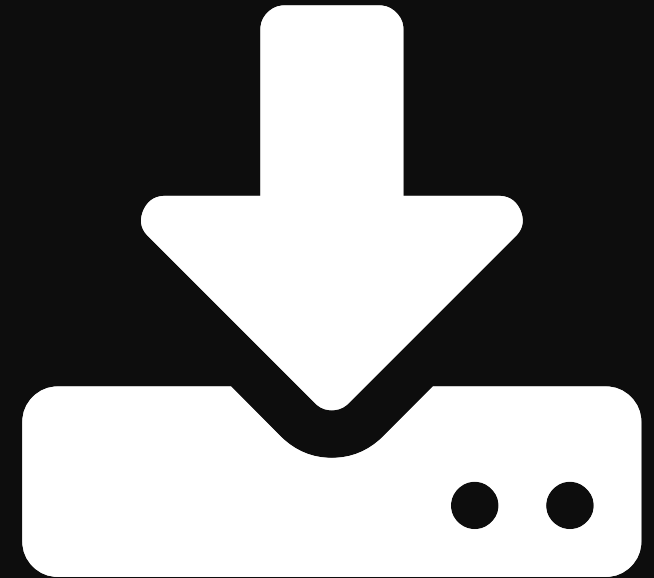
Chocolatey

Homebrew

GitHub

Docker HUB*

MyGet



So what is it?

Cake is a .NET based,
open source,
build orchestration tool,
with a C# scripting DSL,
powered by Roslyn.



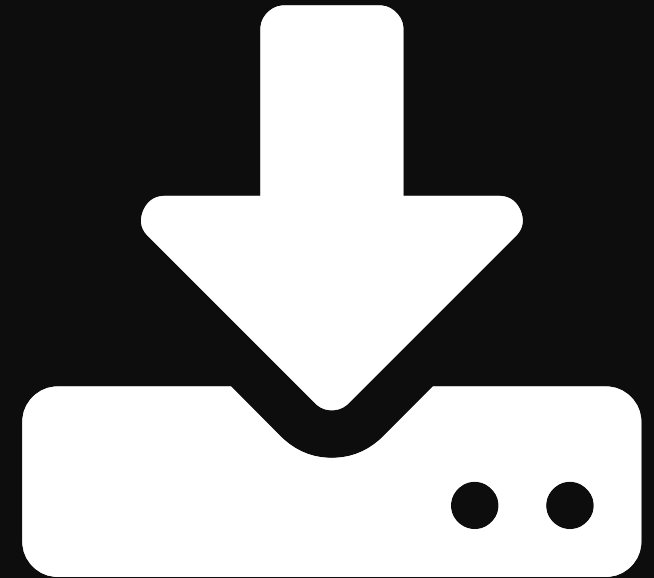
Obtaining Cake

Default recommended way is via NuGet
through official bootstrappers

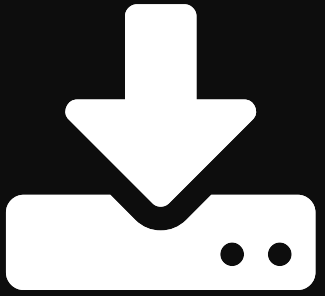
build.ps1



build.sh



Obtaining Cake



Invoke-WebRequest <https://cakebuild.net/download/bootstrapper/windows> -OutFile build.ps1

curl -Lsfo build.sh <https://cakebuild.net/download/bootstrapper/linux>

curl -Lsfo build.sh <https://cakebuild.net/download/bootstrapper/osx>

Why C# scripting?

No cruft

No project files

No solution files

No class / Main()

Low ceremony

Self contained

Compiled...

...the whole script

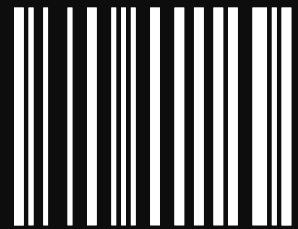
Strongly typed

Backed by all of .NET

Same experience regardless of...

SDK / Framework / Environment

X2



CODE

```
Information("Hello world!");
```

DSL?

Task orchestrator

Preprocessor directives

Aliases

Global Methods

Global Properties

x²

Task orchestration



```
Task("Clean")
```

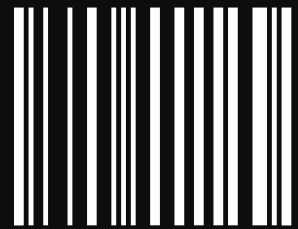
```
Task("Restore")
```

```
Task("Build")  
  .IsDependentOn("Clean")  
  .IsDependentOn("Restore")
```

```
Task("Test")  
  .IsDependentOn("Build")
```

```
Task("Package")  
  .IsDependentOn("Test")
```

```
Task("Publish")  
  .IsDependentOn("Package")
```



CODE

```
Task("Clean");

Task("Restore");

Task("Build")
    .IsDependentOn("Clean")
    .IsDependentOn("Restore");

Task("Test")
    .IsDependentOn("Build");

Task("Package")
    .IsDependentOn("Test");

Task("Publish")
    .IsDependentOn("Package");
```

Task lifecycle

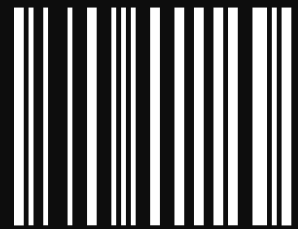
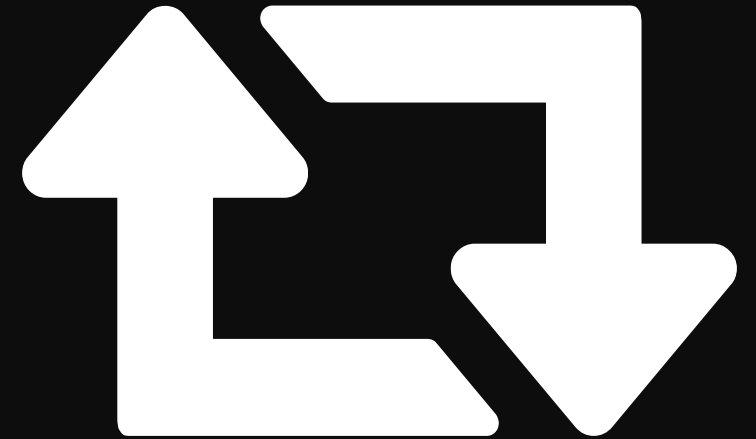
Setup

TaskSetup

Task

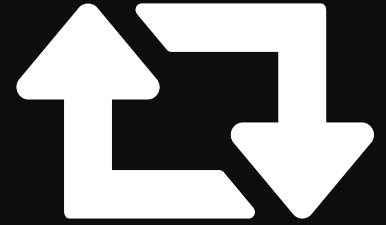
TaskTeardown

Teardown



CODE

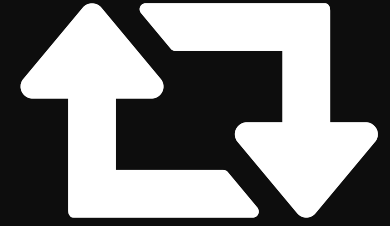
Setup / Teardown



```
Setup(context =>
{
    // Executed BEFORE the first task.
});
```

```
Teardown(context =>
{
    // Executed AFTER the last task.
});
```

Task Setup / Teardown



```
TaskSetup((context, task) =>
{
    var message = string.Format("Task: {0}", task.Task.Name);
    // custom logging
});
```

```
TaskTeardown((context, task) =>
{
    var message = string.Format("Task: {0}", task.Task.Name);
    // custom logging
});
```

Preprocessor directives

Addin directive

Load directive

Module directive

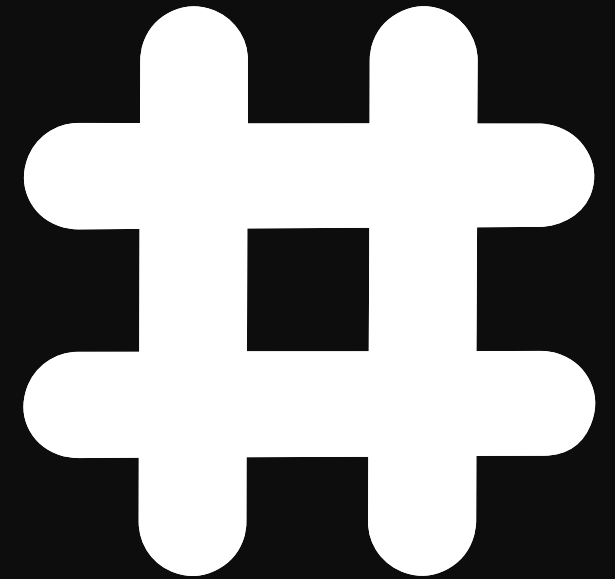
Reference directive

Tool directive

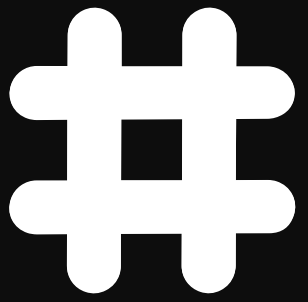
Shebang directive

Break directive

Using directive



Addin directive



```
#addin nuget:?package=Cake.Foo
```

```
#addin nuget:?package=Cake.Foo&version=1.2.3
```

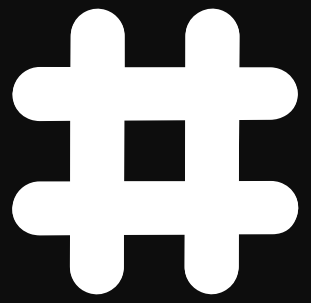
```
#addin nuget:?package=Cake.Foo&prerelease
```

```
#addin nuget:https://myget.org/f/Cake/?package=Cake.Foo&prerelease
```

```
#addin nuget:?package=Cake.Foo&exclude=/**/NoFoo.dll
```

```
#addin nuget:?package=Cake.Foo&include=/**/Foo.dll
```

Load directive



```
#1 scripts/utilities.cake
```

```
#load scripts/utilities.cake
```

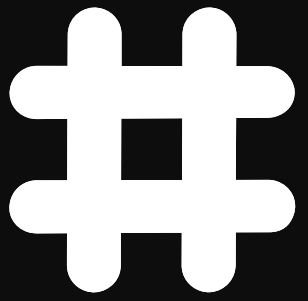
```
#1 nuget:?package=utilities.cake
```

```
#load nuget:?package=My.Cake.Scripts
```

```
#load nuget:?package=My.Cake.Scripts&exclude=/**/NoFoo.cake
```

```
#load nuget:?package=My.Cake.Scripts&include=/**/Foo.cake
```


Module directive

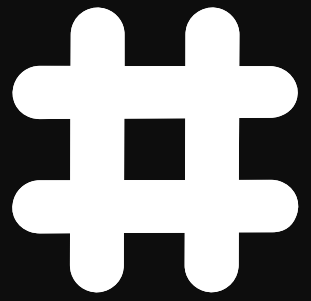


```
#module nuget:?package=Cake.Foo.Module
```

```
#module nuget:?package=Cake.Foo.Module&version=1.2.3
```

```
./cake.exe build.cake --bootstrap
```

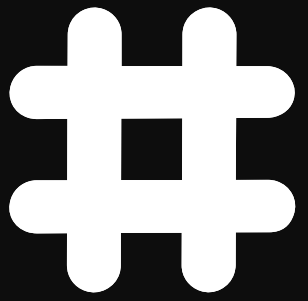
Reference directive



```
#r "bin/myassembly.dll"
```

```
#reference "bin/myassembly.dll"
```

Tool directive



```
#tool nuget:?package=Cake.Foo
```

```
#tool nuget:?package=Cake.Foo&version=1.2.3
```

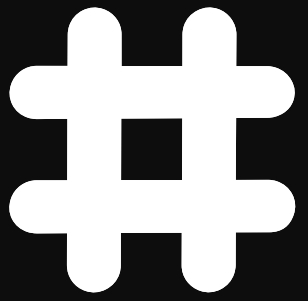
```
#tool nuget:?package=Cake.Foo&prerelease
```

```
#tool nuget:https://myget.org/f/Cake/?package=Cake.Foo&prerelease
```

```
#tool nuget:?package=Cake.Foo&exclude=/**/NoFoo.exe
```

```
#tool nuget:?package=Cake.Foo&include=/**/Foo.exe
```

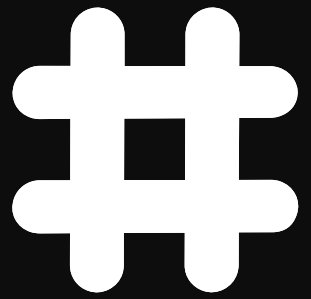
Shebang directive



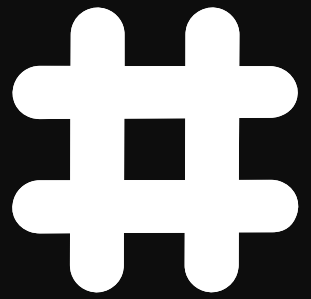
`#!/path/to/launch/cake`

Break directive

#break



Using directive

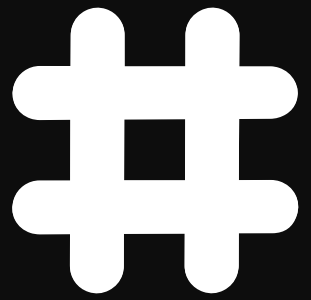


```
using Newtonsoft.Json;
```

System	System.Collections.Generic	System.Text
System.Linq	System.Threading.Tasks	System.IO
Cake.Core	Cake.Core.Scripting	Cake.Core.IO
Cake.Core.Diagnostics		

...and namespaces for alias methods & properties

Using static directive



```
using static System.Math;
```

```
Information(Round(1.1));
```

Aliases

Argument

Environment variables

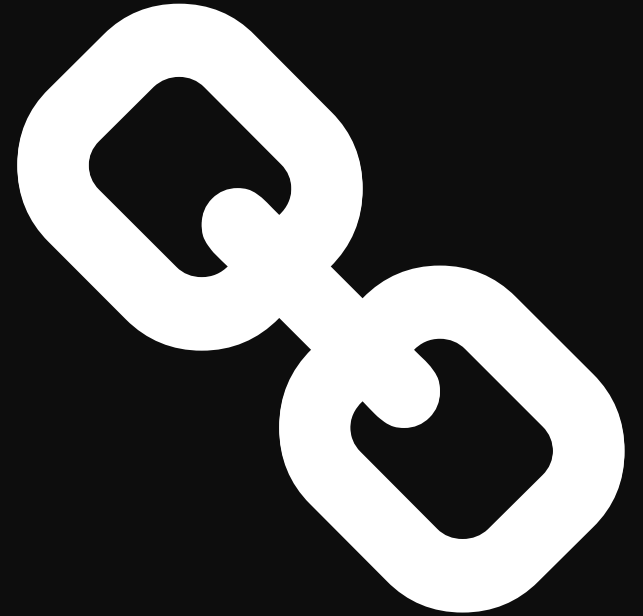
Cleaning

Globbering

MSBuild

.NET Core CLI

...etc.



Cake TL;DR

Build orchestration tool

Superset of C#

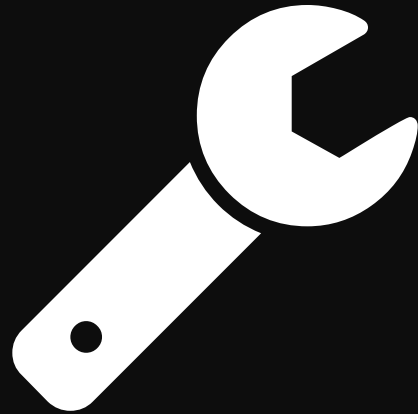
Cross platform, environment & runtime

Configurable, extensible & modular

Versioned build & release definition



THANKS !



```

+##  #;;'
###  .+;;;+,
'+;;#;+';;;;'#.
++''''';;;;;;# ;#;
##';;;++'+#;;;;'`#;
;#  '+''';;;;;;#`#.#
`#,'  .'++;;;;'':.....#
'+  `.....';;;;'':.....#
#.....+;;;;'':.....#
#.....#';;;;;'+''''''.#
#.....;''''''''##';;;;'+'''''',
#''''''''''''''''###';;;;+'''''#
#''''''''''''''''####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#''''''''''''''''#####';;;;#''''#
#+#####'+#####';;;;#''''#
#+#####':,` ,#####';;;;#''''#
                                     +##'''''+.

```

ackealike

Mattias Karlsson - @devlead

Contact info



Mattias



<https://medium.com/@devlead>



<https://github.com/devlead>



<https://twitter.com/devlead>



<https://www.linkedin.com/in/devlead/>



<https://officehours.io/people/devlead>



<https://devlead.se>

Cake



<http://cakebuild.net/>



<https://medium.com/@cakebuildnet>



<https://github.com/cake-build>



<https://twitter.com/cakebuildnet>



<https://gitter.im/cake-build/cake>