jbang - a better java ?

by

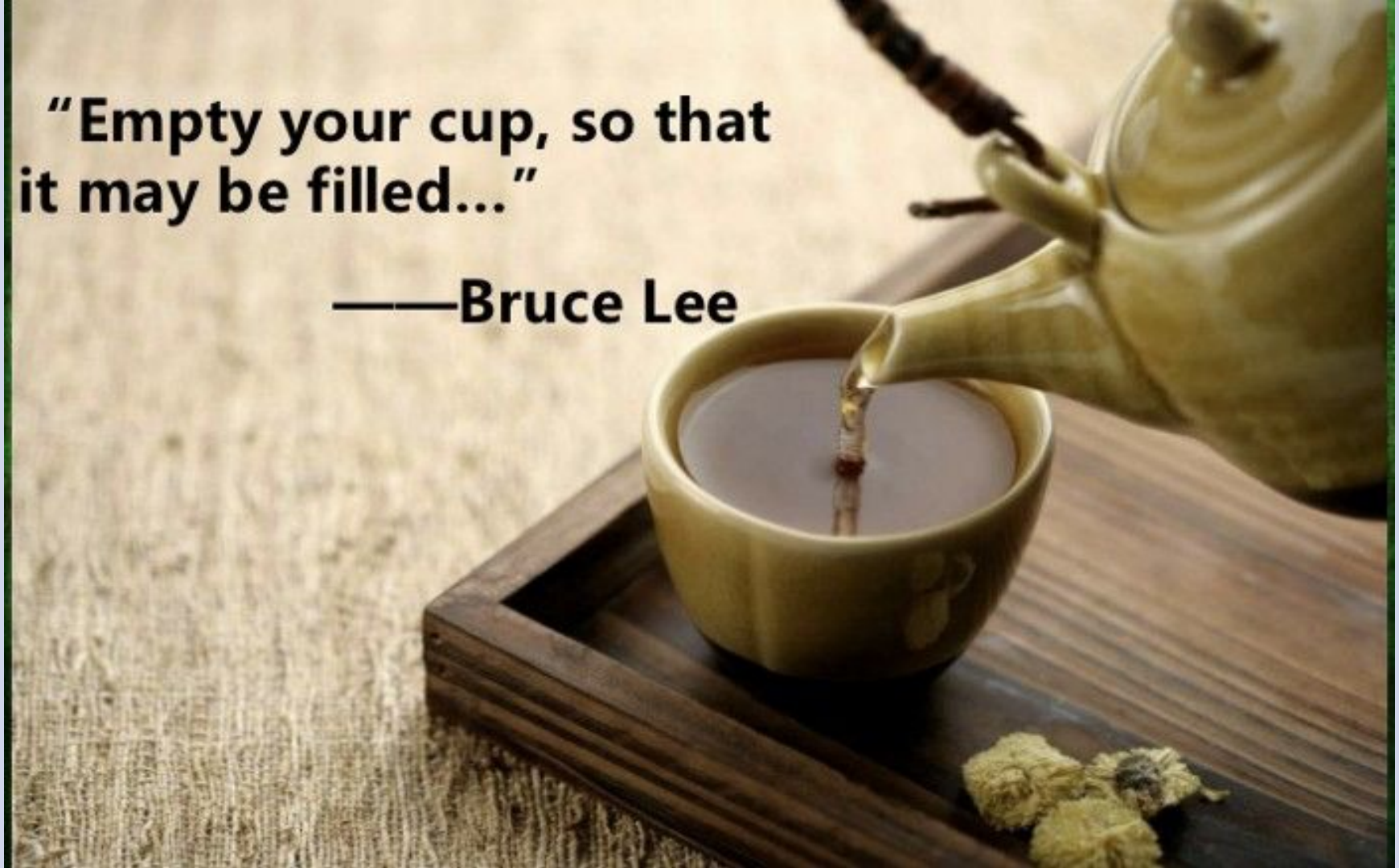
Max Rydahl Andersen
@maxandersen
https://jbang.dev
2020-11-17

Who is Max ?

Java Developer for 20+ years
Python for 22+ years
Involved in developing tooling for Java, Python, NodeJS, Go, and more…

Took a sabbatical 2018-2019...

Now back...


"Empty your cup, so that it may be filled..."
——Bruce Lee


J'BANG!

Learn

Run

Build

Package

Learn

LEARN

```
hello.java:
///usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS com.github.lalyos:jfiglet:0.0.8
import com.github.lalyos.jfiglet.FigletFont;
class hello {
    public static void main(String... args) throws Exception {
       System.out.println(FigletFont.convertOneLine(
                "Hello " + ((args.length>0)?args[0]:"jbang")));
    }
}
```

Valid Java AND shell script

External Dependency management

```
> ./hello.java jug.ch
[jbang] Resolving dependencies...
[jbang]     Resolving com.github.lalyos:jfiglet:0.0.8...
[jbang] Dependencies resolved
[jbang] Building jar...
  _   _          _   _                                  _
 | | | |   ___  | | | |   ___          (_)  _   _    __ _          ___   | |__
 | |_| |  / _ \ | | | |  / _ \         | | | | | |  / _` |        / __|  | '_ \
 |  _  | |  __/ | | | | | (_) |        | | | |_| | | (_| |   _   | (__   | | | |
 |_| |_|  \___| |_| |_|  \___/       _/ |  \__,_|  \__, |  (_)   \___|  |_| |_|
                                    |__/             |___/
```

Compile and run in one!

Learn

```
.
├── README.md
└── src
    └── com
        └── d2iq
            └── kubectl
                └── Main.java

8 directories, 13 files
```

```
.
├── README.md
└── kubectl-example

0 directories, 2 files
```

```
$ kubectl example pod list

| Pod Name
|==================================
| coredns-6955765f44-f966p
| coredns-6955765f44-xnzbg
| etcd-kind-control-plane
| kindnet-cznll
| kube-apiserver-kind-control-plane
| kube-controller-manager-kind-cont
| kube-proxy-tw9cb
| kube-scheduler-kind-control-plane
| local-path-provisioner-7745554f7f
```

LEARN

```
.
├── README.md
├── build.gradle
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
└── src
    └── main
        └── java
            └── com
                └── d2iq
                    └── kubectl
                        ├── ExampleCommand.java
                        ├── Main.java
                        ├── PodAddCommand.java
                        ├── PodCommand.java
                        ├── PodList2Command.java
                        ├── PodListCommand.java
                        └── ResourcesCommand.java

8 directories, 13 files
```

```
.
├── README.md
└── kubectl-example

0 directories, 2 files
```

```
$ kubectl example pod list

| Pod Name
|=================================
| coredns-6955765f44-f966p
| coredns-6955765f44-xnzbg
| etcd-kind-control-plane
| kindnet-cznll
| kube-apiserver-kind-control-plane
| kube-controller-manager-kind-cont
| kube-proxy-tw9cb
| kube-scheduler-kind-control-plane
| local-path-provisioner-7745554f7f
```

How about multiple files ?

```
///usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS io.quarkus:quarkus-resteasy:1.8.1.Final
//DEPS ...

//FILES META-INF/resources/index.html=index.html
//SOURCES **/*.java

import io.quarkus.runtime.Quarkus;
import javax.enterprise.context.ApplicationScoped;
import javax.ws.rs.*;

@Path("/hello")
@ApplicationScoped
public class myapp {

    @GET
    public String sayHello() {
        return "hello from Quarkus with " + App.NAME;
    }

}
```

Include files

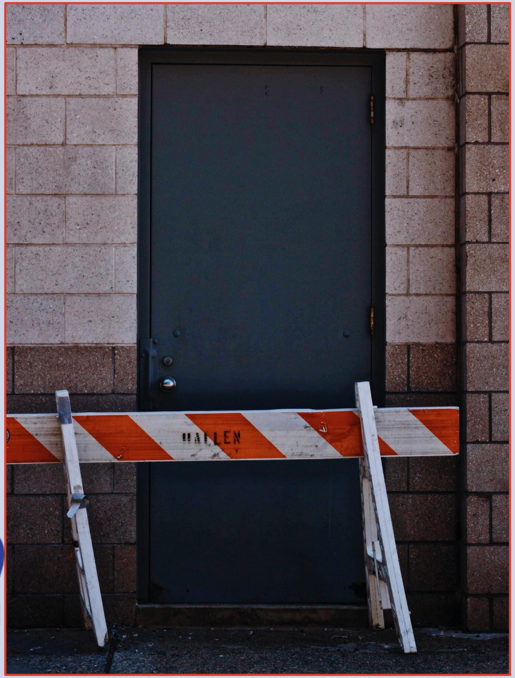
Optional explicit include source to be compiled

Just works!

Learn

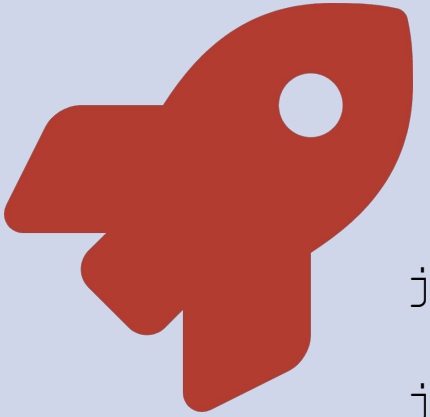
- Get started quickly
- Explore new API’s with no overhead
- Just focus on the code

- `jbang init` to create app
- `jbang xyz.java` or `./xyz.java` to run
- Dependencies as comments
- Edit in any modern IDE/editor using
  `jbang edit`



RUN

Java™

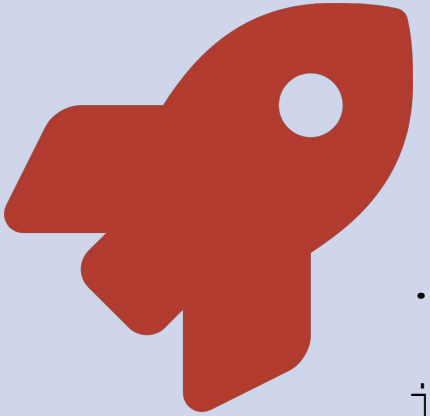


RUN

```
java xyz.java

jshell -q xyz.jsh*

java -jar xyz.jar

mvn exec:java

mvn javafx:run
```

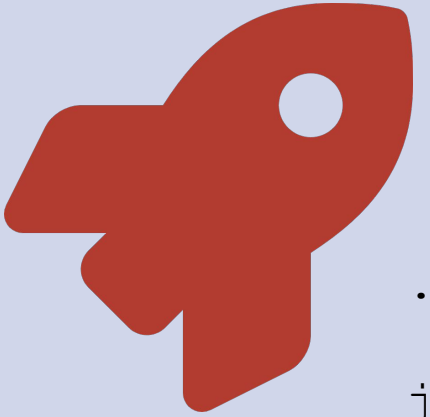
How do I pass arguments ?

How do I debug ?



RUN

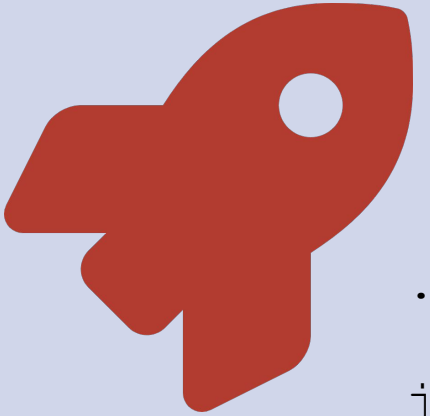
```
./xyz.java

jbang xyz.java

jbang xyz.jsh

jbang xyz.jar
```



RUN

```
./xyz.java param1 param2

jbang xyz.java param1 param2

jbang xyz.jsh param1 param2

jbang xyz.jar param1 param2
```

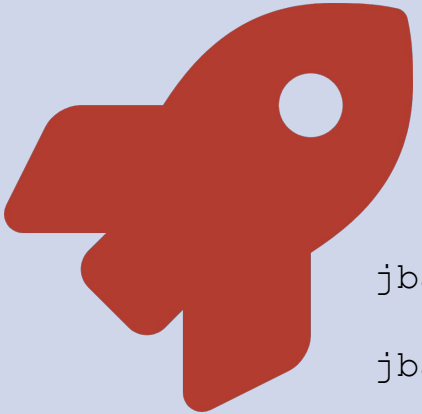
How do I pass arguments ?



RUN

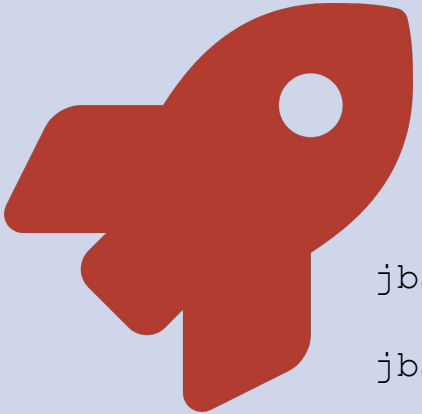
```
./xyz.java(*)

jbang --debug xyz.java

~~jbang xyz.jsh~~

jbang --debug xyz.jar
```

How do I debug ?



RUN


```
jbang io.quarkus:quarkus-cli:1.9.1.Final:runner

jbang https://gist.github.com/tonivade/3db...

jbang https://github.com/jbangdev/jbang/../examples/inetTest.java

jbang https://twitter.com/maxandersen/status/1307832761164664834

jbang https://acme.org/download/myapp.jar
```




```
jbang alias add cli io.quarkus:quarkus-cli:1.9.0.Final:runner

jbang cli

jbang cli@quarkusio
```

RUN

RUN

Build

BUILD

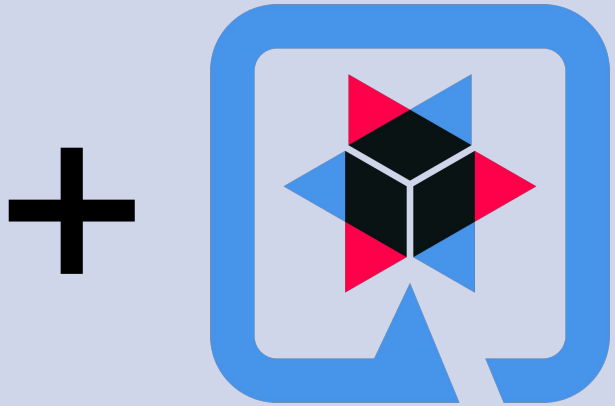
YAGNI

* You Ain't Gonna Need It (at least not initially)

Java™

Build

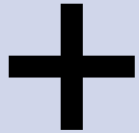
```
jbang build xyz.java

jbang run xyz.java

jbang --native xyz.java
```



Build

```
> jbang init -t qmetrics quarkusmetrics.java
[jbang] File initialized. You can now run it with 'jbang quarkusmetrics.java' or edit it using 'code `jbang edit quark
usmetrics.java`'
> jbang quarkusmetrics.java
[jbang] Resolving dependencies...
[jbang]     Resolving io.quarkus:quarkus-resteasy:1.8.0.Final...Done
[jbang]     Resolving io.quarkus:quarkus-smallrye-metrics:1.8.0.Final...Done
[jbang] Dependencies resolved
[jbang] Building jar...
[jbang] Post build with io.quarkus.launcher.JBangIntegration
Sep 15, 2020 3:56:54 PM org.jboss.threads.Version <clinit>
INFO: JBoss Threads version 3.1.1.Final
Sep 15, 2020 3:56:54 PM io.quarkus.deployment.QuarkusAugmentor run
INFO: Quarkus augmentation completed in 961ms
__  ____  __  _____   ___  __ ____  ______
 --/ __ \/ / / / _ | / _ \/ //_/ / / / __/
 -/ /_/ / /_/ / __ |/ , _/ ,< / /_/ /\ \
--\___\_\____/_/ |_/_/|_/_/|_|\____/___/
2020-09-15 15:56:56,155 INFO  [io.quarkus] (main) Quarkus 1.8.0.Final on JVM started in 0.820s. Listening on: http://0
.0.0.0:8080
2020-09-15 15:56:56,174 INFO  [io.quarkus] (main) Profile prod activated.
2020-09-15 15:56:56,174 INFO  [io.quarkus] (main) Installed features: [cdi, resteasy, smallrye-metrics]
```

Build



Requires jbang 0.45+ and Quarkus 1.8+

Quarkus Configuration with `//Q:CONFIG`

Native image generation:

```
jbang --native quarkusrest.java
```

Container deploy:

```
jbang -Dquarkus.container-image.build=true quarkus.java
```

EXPERIMENTAL! Please Open issues :)

YAGNI
KISS
DRY

BUILD

Package

Developers

1. Code it
2. Commit to Git
3. Build it
4. Publish it
5. Ship it

Users

1. Download Java
2. Install Java
3. Download App
4. Install App
5. Run it

PACKAGE

Package

https://jbang.dev (scoop, choco, brew, yum, snap, docker, mvn, gradle, …)

```
curl -Ls https://sh.jbang.dev | bash -s - xyz.java
```

```
iex "& { $(iwr https://ps.jbang.dev) } xyz.java"
```

How do a user get `jbang` ?

Package

How do a user get Java ?

```
$ jbang --java 15 xyz.java
[jbang] Downloading JDK 15. Be patient, this can take several minutes…
```

records.java:

```
//JAVA 14+
//JAVAC_OPTIONS --enable-preview --release 14
//JAVA_OPTIONS --enable-preview

import static java.lang.System.*;

public class records {

    record Point(int x, int y) {}

    public static void main(String[] args) {
        var p = new Point(2,4);
        out.println(p);
    }
}
```

Package

```
jbang init -t cli myapp.java

...code…

git add/commit/push

jbang https://github.com/acme/myapp/src/myapp.java
```

How do a dev code, publish, test and ship a `jbang` app ?

Developers

Users

1. Code it
2. Commit to Git

PACKAGE

5. Run it

```
curl -Ls https://sh.jbang.dev | bash -s - myapp@acme https://github.com/acme/myapp...
```

Alias Catalogs

jbang run@jruby
https://github.com/jruby/jbang-catalog

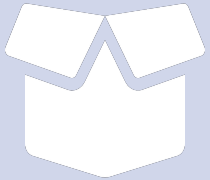
jbang cli@quarkusio
https://github.com/quarkusio/jbang-catalog

jbang gavsearch@jbangdev
https://github.com/jbangdev/jbang-catalog

Easy to get started, explore any API, Just focus on the code

Run as script, run java, jsh, jar, GAV via url, gist, github, bitbucket, gitlab, ...

Just use Java w/JBang, YAGNI anything else



Code, Commit, Run...

Things I haven’t covered

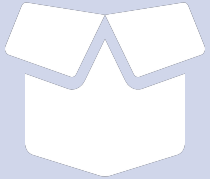
Java Flight Recorder (--jfr)
Class Data Sharing (--cds)
Wrapper (jbang wrapper install)
JDK Management (jbang jdk env)
Completion support (jbang completion)
Java Agents (--javaagent myagent.java)
...


The Road to 1.0 ?

More Examples, template via aliases catalogs
(`jbang init -t start@hibernate`)

Better caching, make second runs even faster!

Build output (`--output=mydir`), import scope/POM support



`jbang publish myapp.java` and `jbang install gavsearch@jbangdev`

The Road to 1.0 ?

Learn

Run

Build

Package

Want to help ?

Tweet

Stream

Facebook

LinkedIn

Blog

contribute

jbang-catalog

J'BANG!

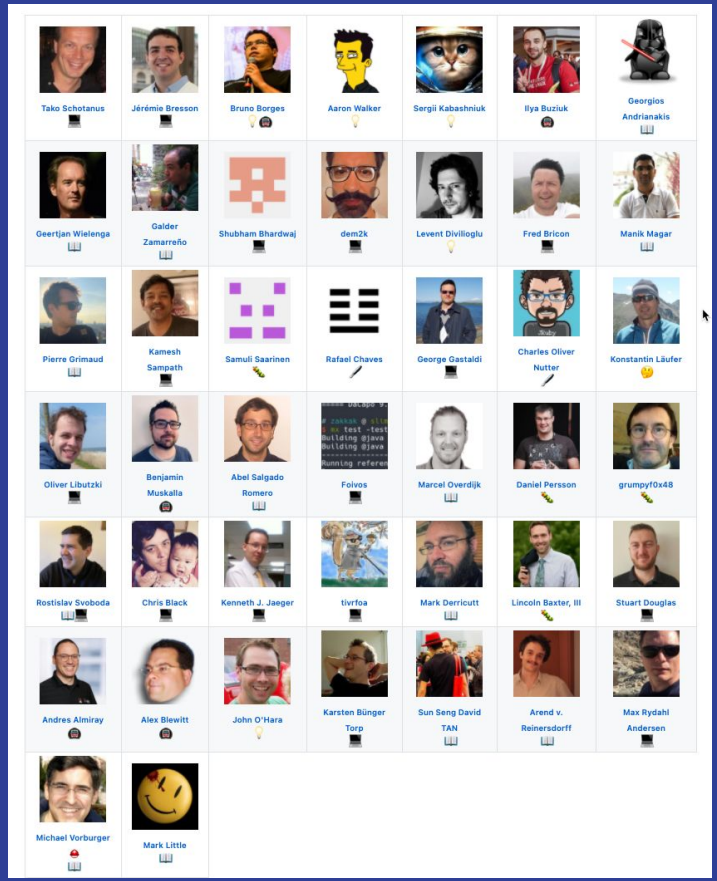
Tweet your gist/github jbang script with a **#jbang** **#jugch**

Questions ?

https://jbang.dev

@maxandersen

Thanks!



RUN


```
jbang
  --javaagent=org.jboss.byteman:byteman:4.0.13
    =script:test.btm
  xyz.java param1 param2
```

Bonus: how do I run an agent ?

Java 11+

Cannot be called .java

Breaks every IDE :(

#! works
but must have --source 11

```
hello:
#!/usr/bin/env java --source 11
class hello {
    public static void main(String... args) throws Exception {
        System.out.println(
                "Hello " +
                ((args.length>0)?args[0]:"jbang"));
    }
}
```

```
❯ chmod +x hellobang; ./hellobang devnation
Hello devnation
```

		Java 1-8	Java 10	Java 11	Java 8+
1	Compile & build in one	✗	✓	✓	✓
2	External Dependencies	✗	✗	✗	✓
3	Executable scripts (./hello)	✗	✗	✓	✓
4	IDE support	✗	✗	✗	✓
5	(Easy) Debugging	✗	✗	✗	✓

```
hello.java:
//usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS com.github.lalyos:jfiglet:0.0.8

import com.github.lalyos.jfiglet.FigletFont;

class hello {

    public static void main(String... args) thr
        System.out.println(FigletFont.convertOneL
                "Hello " + ((args.length>0)?args[
    }
}
```

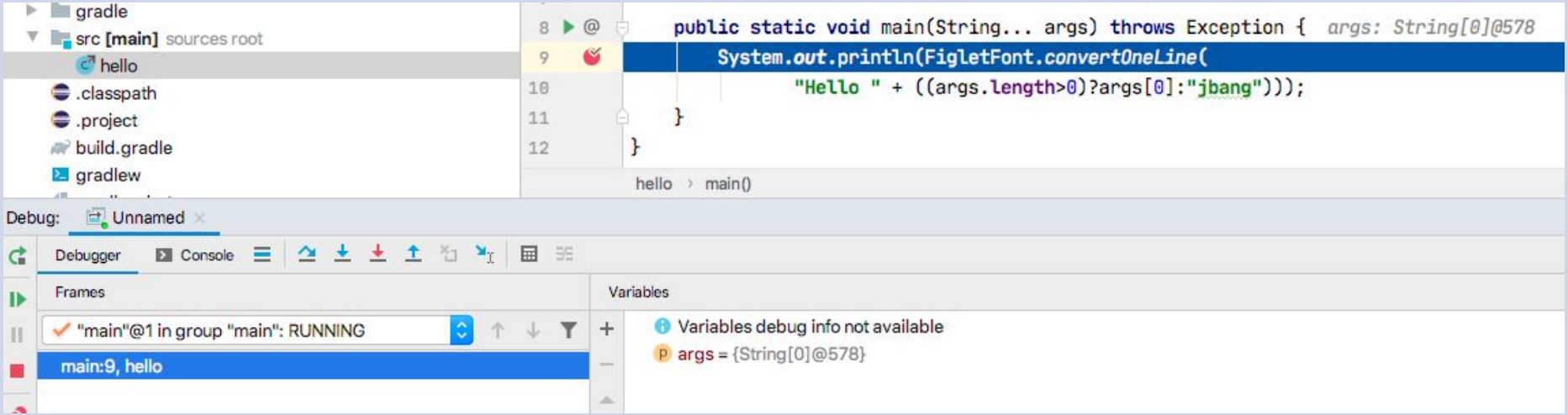
Valid Java AND shell script

External Dependency management

Compile and run in one!

```
❯ ./hello.java devnation!
[jbang] Resolving dependencies...
[jbang]     Resolving com.github.lalyos:jfiglet:0.0.8...Done
[jbang] Dependencies resolved
[jbang] Building jar...
```

```
> jbang edit --live=idea hello.java
[jbang] Running `idea /Users/max/.jbang/cache/projects/hello.java_jbang_ecdedafb2d74a2
ad5391f690630f6c6c48aeb5a96ef5394c93455a510b47b10a/hello`
[jbang] Watching for changes in /Users/max/code/personal/jbang-tutorial
```




Eclipse


Emacs w/ Spacemacs Java


Apache NetBeans

Yes - They All Work!


IntelliJ Idea


Neovim w/ spacevim Java

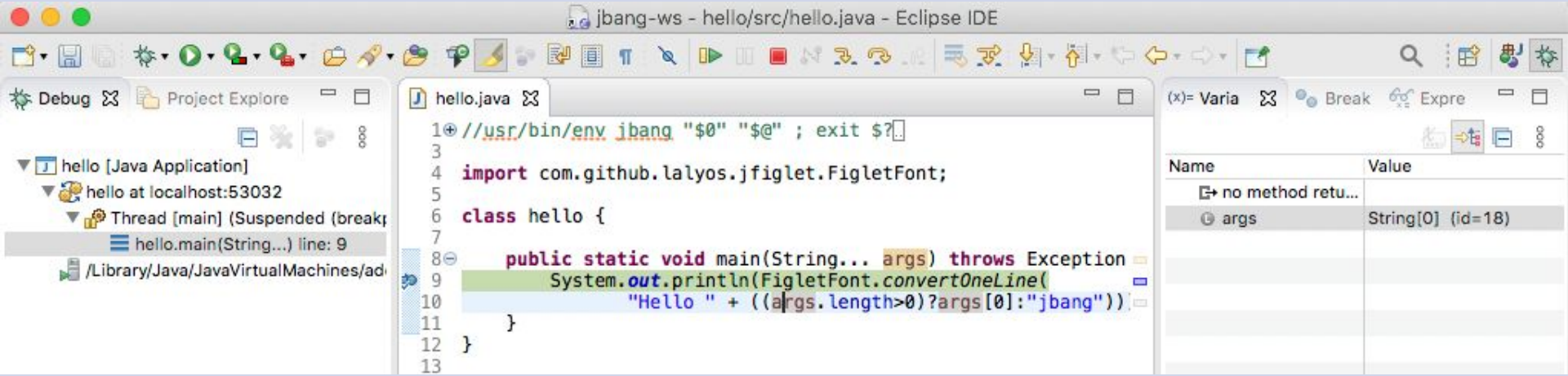

Visual Studio Code


Github Codespaces

Making Java™ easier...

```
java -agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=*:4004
hello.java
```

```
❯ jbang --debug hello.java
Listening for transport dt_socket at address: 4004
```



Run everything from anywhere

class, .jsh or jar

```
jbang hello.(java|jsh|jar)
```

url (with swizzling for github, gitlab, bitbucket, carbon.sh, twitter, …)

```
jbang https://(github|gitlab|bitbucket|carbon|twitter)/<path>
```

jar via GAV’s

```
jbang io.quarkus:quarkus-cli:999-SNAPSHOT:runner
```

Run everything from anywhere

```
❯ jbang https://github.com/jbangdev/jbang/blob/master/examples/jfxtiles.java
Initialization: 738ms
check 1
check 2
Rendering      : 2817ms
Nodes in Scene: 1349
```

Yup!
Auto-configuration of JavaFX modules

Automatic download of Java

```
//usr/bin/env jbang "$0" "$@" ; exit $?
//JAVA 14+
//JAVAC_OPTIONS --enable-preview -source 14
//JAVA_OPTIONS --enable-preview

import static java.lang.System.*;

public class records {

    record Point(int x, int y) {}

    publi
        v
        c
    }
}
```

```
> jbang records.java
[jbang] Downloading JDK 14. Be patient, this can take several minutes...
[jbang] Installing JDK 14...
[jbang] Building jar...
Note: records.java uses preview language features.
Note: Recompile with -Xlint:preview for details.
Point[x=2, y=4]
```

Aliases/Catalogs

Add/Run alias:

```
jbang alias add hello hello.java
jbang hello
```

Github/Gitlab/bitbucket powered catalogs:

```
jbang properties@jbang.dev
```

```
jbang groovy@maxandersen
```

```
jbang run@jruby -e "puts 'Hello Ruby World'"
```

Before & After Getting Started with Java

Download java

Download maven/gradle/IDE

Create folder

mkdir src/java/main

Write build.gradle / pom.xml

…

Distribute it somehow

Make installer

...

Download jbang

```
jbang init myapp.java
```

```
Push to git / catalog
```

```
jbang myapp@acme
```

How do I get it ?

https://jbang.dev/download

- Windows
- Linux
- Mac
- Docker/Container
- Github Action

Go Use JBang!

Spread the love!
Twitter, Facebook, linkedin, etc.

Init for getting started quickly

```
❯ jbang init -t cli hellocli.java
[jbang] File initialized. You can now run it with 'jbang hellocli.java' or edit it using 'code `jbang edit hellocli.ja
va`'
❯ ./hellocli.java
[jbang] Building jar...
Hello World!
❯ ./hellocli.java --help
Usage: hellocli [-hV] <greeting>
hellocli made with jbang
      <greeting>   The greeting to print
  -h, --help       Show this help message and exit.
  -V, --version    Print version information and exit.
```

```
//REPOS jboss,google
```

Easy add common maven repositories

```
//REPOS acme=https://maven.acme.local/maven
```

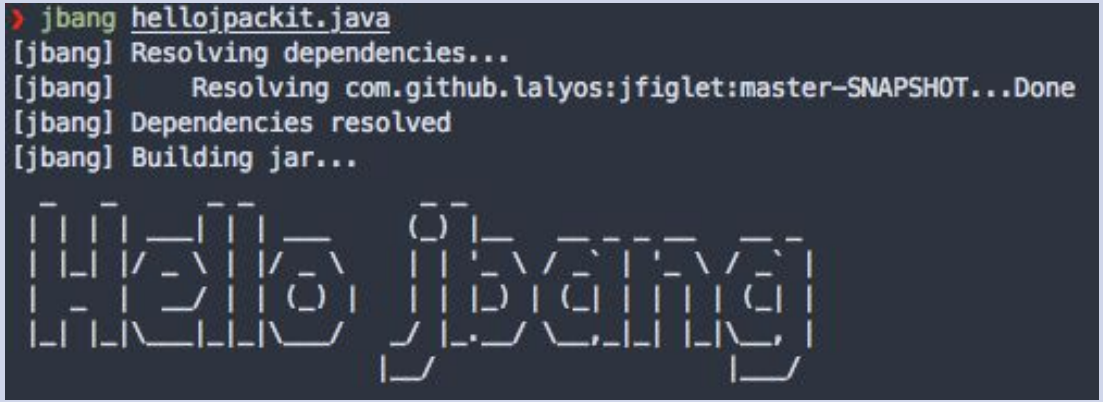
Use custom incl. Secured repoistories

```
@GrabResolver("mavenCentral") // (2)
@GrabResolver(name='acme', root='https://maven.acme.local/maven')
@Grapes({ // (3)
            @Grab(group="org.codehaus.groovy", module="groovy", version="2.5.8"),
            @Grab(module = "log4j", group = "log4j", version = "1.2.17")
})
```

Use Groovy style @Grab

```
hello.java:
//usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS https://github.com/lalyos/jfiglet

import com.github.lalyos.jfiglet.FigletFont;

class hello {

    public static void main(String... args) throws Exception {
        System.out.println(FigletFont.convertOneLine(
                "Hello " + ((args.length>0)?args[0]:"jbang")));
    }
}
```

Use github repos as dependencies (snapshot, branches & tags)



```
hello.jsh:
//usr/bin/env jbang "$0" "$@" ; exit $?
//DEPS com.github.lalyos:jfiglet:0.0.8

import com.github.lalyos.jfiglet.FigletFont;

System.out.println(FigletFont.convertOneLine(
                "Hello " + ((args.length>0)?args[0]:"jbang")));
/exit
```

jshell !

No main()!

arguments!

Executable jshell script WITH arguments!

```
> ./hello.jsh vjug
 _   _          _   _                                  _
| | | |   ___  | | | |   ___      __   __   (_)   _   _    __ _
| |_| |  / _ \ | | | |  / _ \     \ \ / /   | |  | | | |  / _` |
|  _  | |  __/ | | | | | (_) |     \ V /    | |  | |_| | | (_| |
|_| |_|  \___| |_| |_|  \___/       \_/    _/ |  \__,_|  \__, |
                                          |__/             |___/
```

Write kubectl extensions

```
kubectl example pod list
```

```
+-----------------------------------------------------------------+
| Pod Name                                      | namespace       |
|=================================================================|
| coredns-6955765f44-f966p                      | kube-system     |
| coredns-6955765f44-xnzbg                      | kube-system     |
| etcd-kind-control-plane                       | kube-system     |
| kindnet-cznll                                 | kube-system     |
| kube-apiserver-kind-control-plane             | kube-system     |
| kube-controller-manager-kind-control-plane    | kube-system     |
| kube-proxy-tw9cb                              | kube-system     |
| kube-scheduler-kind-control-plane             | kube-system     |
| local-path-provisioner-7745554f7f-gk5j9       | local-path-storage|
```

See how at https://github.com/jbangdev/k8s-cli-java